Table of Contents

© Copyright GATE Overflow. All rights reserved.

© Copyright GATE Overflow. All rights reserved.

Contributors

User	👍, Answers
Arjun Suresh	7484, 215
Praveen Saini	2352, 67
Gate Keeda	1449, 60
Akash Kanase	935, 39
Vikrant Singh	650, 17
Amar Vashishth	613, 21
Prashant Singh	601, 29
Bhagirathi Nayak	572, 21
Rajarshi Sarkar	556, 28
gatecse	555, 12
Pragy Agarwal	551, 17
Rajesh Pradhan	534, 21
Debashish Deka	515, 9
Digvijay	511, 27
Sankaranarayanan P.N	461, 20
Pooja Palod	435, 18
Kalpna Bhargav	434, 11
Abhilash Panicker	288, 6
Ankit Rokde	282, 10
Umang Raman	276, 15
Manu Thakur	269, 10
Ahwan Mishra	265, 10
Sachin Mittal	260, 8
Anurag Semwal	258, 8
Monanshi Jain	230, 8
Mithlesh Upadhyay	228, 7
jayendra	220, 9
minal	215, 7
Himanshu Agarwal	210, 7
Anoop Sonkar	204, 6
srestha	204, 16
Sandeep_Uniyal	186, 7
Pc	175, 6
suraj	172, 6
sriv_shubham	170, 5
Aditi Dan	168, 5
ryan sequeira	160, 3
Srinath Jayachandran	157, 2
Keith Kr	155, 6

User	?Added
Kathleen Bankson	447
Jotheeswari	187
makhdoom ghaya	146
Ishrat Jahan	122
Arjun Suresh	98
gatecse	35
Rucha Shelke	30
Sandeep Singh	25
Akash Kanase	24
Madhav kumar	11
khush tak	8

User	Done
kenzou	323
Milicevic3306	259
Naveen Kumar	134
Arjun Suresh	96
Lakshman Patel	52
Shikha Mallick	51
Pooja Khatri	33
Krithiga2101	29
Akash Dinkar	28
Ajay kumar soni	14
Subarna Das	10
Puja Mishra	7
Manu Thakur	6
srestha	6
Manoja Rajalakshmi Aravindakshan	4
Jotheeswari	4

© Copyright GATE Overflow. All rights reserved.

1 Algorithms (327)

Searching, Sorting, Hashing, Asymptotic worst case time and Space complexity, Algorithm design techniques: Greedy, Dynamic programming, and Divide-and-conquer, Graph search, Minimum spanning trees, Shortest paths.

Mark Distribution in Previous GATE

Year	2019	2018	2017-1	2017-2	2016-1	2016-2	Minimum	Average	Maximum
1 Mark Count	2	0	2	2	3	3	0	2	3
2 Marks Count	2	4	2	3	2	3	2	2.7	4
Total Marks	6	8	6	8	7	9	**6**	**7.3**	**9**

1.1 Algorithm Design (8)

1.1.1 Algorithm Design: GATE1992-8

https://gateoverflow.in/587

Let T be a Depth First Tree of a undirected graph G. An array P indexed by the vertices of G is given. $P[V]$ is the parent of vertex V, in T. Parent of the root is the root itself.

Give a method for finding and printing the cycle formed if the edge (u, v) of G not in T (i.e., $e \in G - T$) is now added to T.

Time taken by your method must be proportional to the length of the cycle.

Describe the algorithm in a PASCAL (C) – like language. Assume that the variables have been suitably declared.

gate1992 algorithms descriptive algorithm-design

1.1.2 Algorithm Design: GATE1994-7

https://gateoverflow.in/2503

An array A contains n integers in locations $A[0], A[1], \ldots A[n-1]$. It is required to shift the elements of the array cyclically to the left by K places, where $1 \leq K \leq n-1$. An incomplete algorithm for doing this in linear time, without using another array is given below. Complete the algorithm by filling in the blanks. Assume all variables are suitably declared.

```
min:=n;
i=0;
while _____ do
begin
    temp:=A[i];
    j:=i;
    while ____ do
    begin
        A[j]:=____;
        j:=(j+K) mod n;
       if j<min then
        min:=j;
    end;
    A[(n+i-K)mod n]:=____;
    i:=_____;
end;
```

gate1994 algorithms normal algorithm-design

1.1.3 Algorithm Design: GATE2006-17

https://gateoverflow.in/978

An element in an array X is called a leader if it is greater than all elements to the right of it in X. The best algorithm to find all leaders in an array

A. solves it in linear time using a left to right pass of the array
B. solves it in linear time using a right to left pass of the array
C. solves it using divide and conquer in time $\Theta(n \log n)$
D. solves it in time $\Theta(n^2)$

gate2006 algorithms normal algorithm-design

1.1.4 Algorithm Design: GATE2006-54

https://gateoverflow.in/1832

Given two arrays of numbers $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ where each number is 0 or 1, the fastest algorithm to find the

© Copyright GATE Overflow. All rights reserved.

largest span (i, j) such that $a_i + a_{i+1} + \cdots + a_j = b_i + b_{i+1} + \cdots + b_j$ or report that there is not such span,

A. Takes $O(3^n)$ and $\Omega(2^n)$ time if hashing is permitted
B. Takes $O(n^3)$ and $\Omega(n^{2.5})$ time in the key comparison mode
C. Takes $\Theta(n)$ time and space
D. Takes $O(\sqrt{n})$ time only if the sum of the $2n$ elements is an even number

gate2006 algorithms normal algorithm-design time-complexity

1.1.5 Algorithm Design: GATE2014-1-37

https://gateoverflow.in/1915

There are 5 bags labeled 1 to 5. All the coins in a given bag have the same weight. Some bags have coins of weight 10 gm, others have coins of weight 11 gm. I pick $1, 2, 4, 8, 16$ coins respectively from bags 1 to 5 Their total weight comes out to 323 gm. Then the product of the labels of the bags having 11 gm coins is ___.

gate2014-1 algorithms numerical-answers normal algorithm-design

1.1.6 Algorithm Design: GATE2019-25

https://gateoverflow.in/302823

Consider a sequence of 14 elements: $A = [-5, -10, 6, 3, -1, -2, 13, 4, -9, -1, 4, 12, -3, 0]$. The sequence sum $S(i, j) = \Sigma_{k=i}^{j} A[k]$. Determine the maximum of $S(i, j)$, where $0 \leq i \leq j < 14$. (Divide and conquer approach may be used.)

Subsequence : A subsequence is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements.

Answer: _____

gate2019 numerical-answers algorithms algorithm-design

1.1.7 Algorithm Design: TIFR2011-B-29

https://gateoverflow.in/20576

You are given ten rings numbered from 1 to 10, and three pegs labeled A, B, and C. Initially all the rings are on peg A, arranged from top to bottom in ascending order of their numbers. The goal is to move all the rings to peg B in the minimum number of moves obeying the following constraints:

i. In one move, only one ring can be moved.
ii. A ring can only be moved from the top of its peg to the top of a new peg.
iii. At no point can a ring be placed on top of another ring with a lower number.

How many moves are required?

A. 501 B. 1023 C. 2011 D. 10079 E. None of the above.

tifr2011 algorithms algorithm-design

1.1.8 Algorithm Design: TIFR2019-A-5

https://gateoverflow.in/280505

Asha and Lata play a game in which Lata first thinks of a natural number between 1 and 1000. Asha must find out that number by asking Lata questions, but Lata can only reply by saying “Yes” or “no”. Assume that Lata always tells the truth. What is the least number of questions that Asha needs to ask within which she can always find out the number Lata has thought of?

A. 10 B. 32 C. 100 D. 999 E. None of the above

tifr2019 algorithm-design binary-search

1.2 Algorithm Design Techniques (6)

1.2.1 Algorithm Design Techniques: GATE1990-12b

https://gateoverflow.in/86146

Consider the following problem. Given n positive integers $a_1, a_2 \ldots a_n$, it is required to partition them in to two parts A and B such that

$|\sum_{i \in A} a_i - \sum_{i \in B} a_i|$ is minimised

Consider a greedy algorithm for solving this problem. The numbers are ordered so that $a_1 \geq a_2 \geq \ldots a_n$, and at i^{th} step, a_i is placed in that part whose sum in smaller at that step. Give an example with $n = 5$ for which the solution produced by the greedy algorithm is not optimal.

© Copyright GATE Overflow. All rights reserved.

gate1990 descriptive algorithms algorithm-design-techniques

1.2.2 Algorithm Design Techniques: GATE1990-2-vii

https://gateoverflow.in/83991

Match the pairs in the following questions:

(a)	Strassen's matrix multiplication algorithm	(p)	Greedy method
(b)	Kruskal's minimum spanning tree algorithm	(q)	Dynamic programming
(c)	Biconnected components algorithm	(r)	Divide and Conquer
(d)	Floyd's shortest path algorithm	(s)	Depth-first search

gate1990 match-the-following algorithms algorithm-design-techniques

1.2.3 Algorithm Design Techniques: GATE1997-1.5

https://gateoverflow.in/2221

The correct matching for the following pairs is

A.	All pairs shortest path	1.	Greedy
B.	Quick Sort	2.	Depth-First Search
C.	Minimum weight spanning tree	3.	Dynamic Programming
D.	Connected Components	4.	Divide and Conquer

A. A-2 B-4 C-1 D-3 B. A-3 B-4 C-1 D-2 C. A-3 B-4 C-2 D-1 D. A-4 B-1 C-2 D-3

gate1997 algorithms normal algorithm-design-techniques

1.2.4 Algorithm Design Techniques: GATE2015-1-6

https://gateoverflow.in/8088

Match the following:

P.	Prim's algorithm for minimum spanning tree	i.	Backtracking
Q.	Floyd-Warshall algorithm for all pairs shortest path	ii.	Greedy method
R.	Merge sort	iii.	Dynamic programming
S.	Hamiltonian circuit	iv.	Divide and conquer

A. P-iii, Q-ii, R-iv, S-i
B. P-i, Q-ii, R-iv, S-iii
C. P-ii, Q-iii, R-iv, S-i
D. P-ii, Q-i, R-iii, S-iv

gate2015-1 algorithms normal algorithm-design-techniques

1.2.5 Algorithm Design Techniques: GATE2015-2-36

https://gateoverflow.in/8161

Given below are some algorithms, and some algorithm design paradigms.

1.	Dijkstra's Shortest Path	i.	Divide and Conquer
2.	Floyd-Warshall algorithm to compute all pairs shortest path	ii.	Dynamic Programming
3.	Binary search on a sorted array	iii.	Greedy design
4.	Backtracking search on a graph	iv.	Depth-first search
		v.	Breadth-first search

Match the above algorithms on the left to the corresponding design paradigm they follow.

A. 1-i, 2-iii, 3-i, 4-v
B. 1-iii, 2-iii, 3-i, 4-v
C. 1-iii, 2-ii, 3-i, 4-iv
D. 1-iii, 2-ii, 3-i, 4-v

gate2015-2 algorithms easy algorithm-design-techniques

© Copyright GATE Overflow. All rights reserved.

1.2.6 Algorithm Design Techniques: GATE2017-1-05

https://gateoverflow.in/118707

Consider the following table:

Algorithms	Design Paradigms
P. Kruskal	i. Divide and Conquer
Q. Quicksort	ii. Greedy
R. Floyd-Warshall	iii. Dynamic Programming

Match the algorithms to the design paradigms they are based on.

A. $(P) \leftrightarrow (ii), (Q) \leftrightarrow (iii), (R) \leftrightarrow (i)$
B. $(P) \leftrightarrow (iii), (Q) \leftrightarrow (i), (R) \leftrightarrow (ii)$
C. $(P) \leftrightarrow (ii), (Q) \leftrightarrow (i), (R) \leftrightarrow (iii)$
D. $(P) \leftrightarrow (i), (Q) \leftrightarrow (ii), (R) \leftrightarrow (iii)$

gate2017-1 algorithms algorithm-design-techniques

1.3 Asymptotic Notations (20)

1.3.1 Asymptotic Notations: GATE1994-1.23

https://gateoverflow.in/2466

Consider the following two functions:

$$g_1(n) = \begin{cases} n^3 \text{ for } 0 \leq n \leq 10,000 \\ n^2 \text{ for } n \geq 10,000 \end{cases}$$

$$g_2(n) = \begin{cases} n \text{ for } 0 \leq n \leq 100 \\ n^3 \text{ for } n > 100 \end{cases}$$

Which of the following is true?

A. $g_1(n)$ is $O(g_2(n))$
B. $g_1(n)$ is $O(n^3)$
C. $g_2(n)$ is $O(g_1(n))$
D. $g_2(n)$ is $O(n)$

gate1994 algorithms asymptotic-notations normal

1.3.2 Asymptotic Notations: GATE1996-1.11

https://gateoverflow.in/2715

Which of the following is false?

A. $100n \log n = O(\frac{n \log n}{100})$
B. $\sqrt{\log n} = O(\log \log n)$
C. If $0 < x < y$ then $n^x = O(n^y)$
D. $2^n \neq O(nk)$

gate1996 algorithms asymptotic-notations normal

1.3.3 Asymptotic Notations: GATE2000-2.17

https://gateoverflow.in/664

Consider the following functions

- $f(n) = 3n^{\sqrt{n}}$
- $g(n) = 2^{\sqrt{n} \log_2 n}$
- $h(n) = n!$

Which of the following is true?

A. $h(n)$ is $O(f(n))$
B. $h(n)$ is $O(g(n))$
C. $g(n)$ is not $O(f(n))$
D. $f(n)$ is $O(g(n))$

gate2000 algorithms asymptotic-notations normal

1.3.4 Asymptotic Notations: GATE2001-1.16

https://gateoverflow.in/709

Let $f(n) = n^2 \log n$ and $g(n) = n(\log n)^{10}$ be two positive functions of n. Which of the following statements is correct?

A. $f(n) = O(g(n))$ and $g(n) \neq O(f(n))$
B. $g(n) = O(f(n))$ and $f(n) \neq O(g(n))$
C. $f(n) \neq O(g(n))$ and $g(n) \neq O(f(n))$
D. $f(n) = O(g(n))$ and $g(n) = O(f(n))$

© Copyright GATE Overflow. All rights reserved.

gate2001 algorithms asymptotic-notations time-complexity normal

1.3.5 Asymptotic Notations: GATE2003-20

https://gateoverflow.in/910

Consider the following three claims:

I. $(n+k)^m = \Theta(n^m)$ where k and m are constants
II. $2^{n+1} = O(2^n)$
III. $2^{2n+1} = O(2^n)$

Which of the following claims are correct?

A. I and II B. I and III C. II and III D. I, II, and III

gate2003 algorithms asymptotic-notations normal

1.3.6 Asymptotic Notations: GATE2004-IT-55

https://gateoverflow.in/3698

Let $f(n)$, $g(n)$ and $h(n)$ be functions defined for positive integers such that $f(n) = O(g(n))$, $g(n) \neq O(f(n))$, $g(n) = O(h(n))$, and $h(n) = O(g(n))$.

Which one of the following statements is FALSE?

A. $f(n) + g(n) = O(h(n) + h(n))$
B. $f(n) = O(h(n))$
C. $h(n) \neq O(f(n))$
D. $f(n)h(n) \neq O(g(n)h(n))$

gate2004-it algorithms asymptotic-notations normal

1.3.7 Asymptotic Notations: GATE2008-39

https://gateoverflow.in/450

Consider the following functions:

- $f(n) = 2^n$
- $g(n) = n!$
- $h(n) = n^{\log n}$

Which of the following statements about the asymptotic behavior of $f(n)$, $g(n)$ and $h(n)$ is true?

A. $f(n) = O(g(n)); g(n) = O(h(n))$
B. $f(n) = \Omega(g(n)); g(n) = O(h(n))$
C. $g(n) = O(f(n)); h(n) = O(f(n))$
D. $h(n) = O(f(n)); g(n) = \Omega(f(n))$

gate2008 algorithms asymptotic-notations normal

1.3.8 Asymptotic Notations: GATE2008-IT-10

https://gateoverflow.in/3270

Arrange the following functions in increasing asymptotic order:

A. $n^{1/3}$
B. e^n
C. $n^{7/4}$
D. $n \log^9 n$
E. 1.0000001^n

A. a, d, c, e, b
B. d, a, c, e, b
C. a, c, d, e, b
D. a, c, d, b, e

gate2008-it algorithms asymptotic-notations normal

1.3.9 Asymptotic Notations: GATE2011-37

https://gateoverflow.in/2139

Which of the given options provides the increasing order of asymptotic complexity of functions f_1, f_2, f_3 and f_4?

- $f_1(n) = 2^n$
- $f_2(n) = n^{3/2}$
- $f_3(n) = n \log_2 n$
- $f_4(n) = n^{\log_2 n}$

A. f_3, f_2, f_4, f_1
B. f_3, f_2, f_1, f_4
C. f_2, f_3, f_1, f_4
D. f_2, f_3, f_4, f_1

gate2011 algorithms asymptotic-notations normal

© Copyright GATE Overflow. All rights reserved.

1.3.10 Asymptotic Notations: GATE2012-18

https://gateoverflow.in/50

Let $W(n)$ and $A(n)$ denote respectively, the worst case and average case running time of an algorithm executed on an input of size n. Which of the following is **ALWAYS TRUE**?

A. $A(n) = \Omega(W(n))$
B. $A(n) = \Theta(W(n))$
C. $A(n) = \mathrm{O}(W(n))$
D. $A(n) = \mathrm{o}(W(n))$

gate2012 algorithms easy asymptotic-notations

1.3.11 Asymptotic Notations: GATE2015-3-4

https://gateoverflow.in/8398

Consider the equality $\sum_{i=0}^{n} i^3 = X$ and the following choices for X:

I. $\Theta(n^4)$
II. $\Theta(n^5)$
III. $O(n^5)$
IV. $\Omega(n^3)$

The equality above remains correct if X is replaced by

A. Only I
B. Only II
C. I or III or IV but not II
D. II or III or IV but not I

gate2015-3 algorithms asymptotic-notations normal

1.3.12 Asymptotic Notations: GATE2015-3-42

https://gateoverflow.in/8501

Let $f(n) = n$ and $g(n) = n^{(1+\sin\ n)}$, where n is a positive integer. Which of the following statements is/are correct?

I. $f(n) = O(g(n))$
II. $f(n) = \Omega(g(n))$

A. Only I B. Only II C. Both I and II D. Neither I nor II

gate2015-3 algorithms asymptotic-notations normal

1.3.13 Asymptotic Notations: GATE2017-1-04

https://gateoverflow.in/118703

Consider the following functions from positive integers to real numbers:

$10,\ \sqrt{n},\ n,\ \log_2 n,\ \frac{100}{n}$.
The CORRECT arrangement of the above functions in increasing order of asymptotic complexity is:

A. $\log_2 n,\ \frac{100}{n},\ 10,\ \sqrt{n},\ n$
B. $\frac{100}{n},\ 10,\ \log_2 n,\ \sqrt{n},\ n$
C. $10,\ \frac{100}{n},\ \sqrt{n},\ \log_2 n,\ n$
D. $\frac{100}{n},\ \log_2 n,\ 10,\ \sqrt{n},\ n$

gate2017-1 algorithms asymptotic-notations normal

1.3.14 Asymptotic Notations: TIFR2011-B-27

https://gateoverflow.in/20573

Let n be a large integer. Which of the following statements is **TRUE**?

A. $n^{1/\sqrt{\log_2 n}} < \sqrt{\log_2 n} < n^{1/100}$
B. $n^{1/100} < n^{1/\sqrt{\log_2 n}} < \sqrt{\log_2 n}$
C. $n^{1/\sqrt{\log_2 n}} < n^{1/100} < \sqrt{\log_2 n}$
D. $\sqrt{\log_2 n} < n^{1/\sqrt{\log_2 n}} < n^{1/100}$
E. $\sqrt{\log_2 n} < n^{1/100} < n^{1/\sqrt{\log_2 n}}$

tifr2011 asymptotic-notations

1.3.15 Asymptotic Notations: TIFR2012-B-6

https://gateoverflow.in/25106

Let n be a large integer. Which of the following statements is **TRUE**?

A. $2^{\sqrt{2\log n}} < \frac{n}{\log n} < n^{1/3}$
B. $\frac{n}{\log n} < n^{1/3} < 2^{\sqrt{2\log n}}$
C. $2\sqrt{2\log n} < n^{1/3} < \frac{n}{\log n}$
D. $n^{1/3} < 2\sqrt{2\log n} < \frac{n}{\log n}$
E. $\frac{n}{\log n} < 2\sqrt{2\log n} < n^{1/3}$

© Copyright GATE Overflow. All rights reserved.

tifr2012 algorithms asymptotic-notations

1.3.16 Asymptotic Notations: TIFR2014-B-8

https://gateoverflow.in/27192

Which of these functions grows fastest with n?

A. e^n/n.
B. $e^{n-0.9\log n}$.
C. 2^n.
D. $(\log n)^{n-1}$.
E. None of the above.

tifr2014 algorithms asymptotic-notations

1.3.17 Asymptotic Notations: TIFR2016-B-7

https://gateoverflow.in/30720

Let $n = m!$. Which of the following is **TRUE**?

A. $m = \Theta(\log n/\log\log n)$
B. $m = \Omega(\log n/\log\log n)$ but not $m = O(\log n/\log\log n)$
C. $m = \Theta(\log^2 n)$
D. $m = \Omega(\log^2 n)$ but not $m = \mathrm{O}((\log^2 n)$
E. $m = \Theta(\log^{1.5} n)$

tifr2016 asymptotic-notations

1.3.18 Asymptotic Notations: TIFR2017-A-4

https://gateoverflow.in/94943

Which of the following functions asymptotically grows the fastest as n goes to infinity?

A. $(\log\log n)!$
B. $(\log\log n)^{\log n}$
C. $(\log\log n)^{\log\log\log n}$
D. $(\log n)^{\log\log n}$
E. $2^{\sqrt{\log\log n}}$

tifr2017 algorithms asymptotic-notations

1.3.19 Asymptotic Notations: TIFR2018-A-3

https://gateoverflow.in/179272

Which of the following statements is TRUE for all sufficiently large integers n ?

A. $2^{2^{\sqrt{\log\log n}}} < 2^{\sqrt{\log n}} < n$
B. $2^{\sqrt{\log n}} < n < 2^{2^{\sqrt{\log\log n}}}$
C. $n < 2^{\sqrt{\log n}} < 2^{2^{\sqrt{\log\log n}}}$
D. $n < 2^{2^{\sqrt{\log\log n}}} < 2^{\sqrt{\log n}}$
E. $2^{\sqrt{\log n}} < 2^{2^{\sqrt{\log\log n}}} < n$

tifr2018 asymptotic-notations

1.3.20 Asymptotic Notations: TIFR2019-B-5

https://gateoverflow.in/280490

Stirling's approximation for $n!$ states for some constants c_1, c_2

$$c_1 n^{n+\frac{1}{2}} e^{-n} \leq n! \leq c_2 n^{n+\frac{1}{2}} e^{-n}.$$

What are the tightest asymptotic bounds that can be placed on $n!$?

A. $n! = \Omega(n^n)$ and $n! = \mathcal{O}(n^{n+\frac{1}{2}})$
B. $n! = \Theta(n^{n+\frac{1}{2}})$
C. $n! = \Theta((\frac{n}{e})^n)$
D. $n! = \Theta((\frac{n}{e})^{n+\frac{1}{2}})$
E. $n! = \Theta(n^{n+\frac{1}{2}}2^{-n})$

tifr2019 algorithms asymptotic-notations

1.4 Dynamic Programming (12)

1.4.1 Dynamic Programming: GATE2008-80

https://gateoverflow.in/498

The subset-sum problem is defined as follows. Given a set of n positive integers, $S = \{a_1, a_2, a_3, \ldots, a_n\}$, and positive integer W, is there a subset of S whose elements sum to W? A dynamic program for solving this problem uses a 2-dimensional Boolean array, X, with n rows and $W+1$ columns. $X[i,j], 1 \leq i \leq n, 0 \leq j \leq W$, is TRUE, if and only if there is a subset of $\{a_1, a_2, \ldots, a_i\}$ whose elements sum to j.

Which of the following is valid for $2 \leq i \leq n$, and $a_i \leq j \leq W$?

A. $X[i,j] = X[i-1,j] \vee X[i, j-a_i]$

© Copyright GATE Overflow. All rights reserved.

B. $X[i,j] = X[i-1,j] \vee X[i-1,j-a_i]$
C. $X[i,j] = X[i-1,j] \wedge X[i,j-a_i]$
D. $X[i,j] = X[i-1,j] \wedge X[i-1,j-a_i]$

gate2008 algorithms normal dynamic-programming

1.4.2 Dynamic Programming: GATE2008-81

https://gateoverflow.in/43484

The subset-sum problem is defined as follows. Given a set of n positive integers, $S = \{a_1, a_2, a_3, \ldots, a_n\}$, and positive integer W, is there a subset of S whose elements sum to W? A dynamic program for solving this problem uses a 2-dimensional Boolean array, X, with n rows and $W+1$ columns. $X[i,j], 1 \leq i \leq n, 0 \leq j \leq W$, is TRUE, if and only if there is a subset of $\{a_1, a_2, \ldots, a_i\}$ whose elements sum to j.

Which entry of the array X, if TRUE, implies that there is a subset whose elements sum to W?

A. $X[1,W]$ B. $X[n,0]$ C. $X[n,W]$ D. $X[n-1,n]$

gate2008 algorithms normal dynamic-programming

1.4.3 Dynamic Programming: GATE2009-53

https://gateoverflow.in/1338

A sub-sequence of a given sequence is just the given sequence with some elements (possibly none or all) left out. We are given two sequences $X[m]$ and $Y[n]$ of lengths m and n, respectively with indexes of X and Y starting from 0.

We wish to find the length of the longest common sub-sequence (LCS) of $X[m]$ and $Y[n]$ as $l(m,n)$, where an incomplete recursive definition for the function $I(i,j)$ to compute the length of the LCS of $X[m]$ and $Y[n]$ is given below:

```
l(i,j)  = 0, if either i = 0 or j = 0
        = expr1, if i,j > 0 and X[i-1] = Y[j-1]
        = expr2, if i,j > 0 and X[i-1] ≠ Y[j-1]
```

Which one of the following options is correct?

A. $\text{expr1} = l(i-1,j)+1$
B. $\text{expr1} = l(i,j-1)$
C. $\text{expr2} = \max(l(i-1,j), l(i,j-1))$
D. $\text{expr2} = \max(l(i-1,j-1), l(i,j))$

gate2009 algorithms normal dynamic-programming recursion

1.4.4 Dynamic Programming: GATE2009-54

https://gateoverflow.in/43476

A sub-sequence of a given sequence is just the given sequence with some elements (possibly none or all) left out. We are given two sequences $X[m]$ and $Y[n]$ of lengths m and n, respectively with indexes of X and Y starting from 0.

We wish to find the length of the longest common sub-sequence (LCS) of $X[m]$ and $Y[n]$ as $l(m,n)$, where an incomplete recursive definition for the function $I(i,j)$ to compute the length of the LCS of $X[m]$ and $Y[n]$ is given below:

$l(i,j) = 0$, if either $i = 0$ or $j = 0$
$= expr1$, if $i,j > 0$ and $X[i-1] = Y[j-1]$
$= expr2$, if $i,j > 0$ and $X[i-1] \neq Y[j-1]$

The value of $l(i,j)$ could be obtained by dynamic programming based on the correct recursive definition of $l(i,j)$ of the form given above, using an array $L[M,N]$, where $M = m+1$ and $N = n+1$, such that $L[i,j] = l(i,j)$.

Which one of the following statements would be TRUE regarding the dynamic programming solution for the recursive definition of $l(i,j)$?

A. All elements of L should be initialized to 0 for the values of $l(i,j)$ to be properly computed.
B. The values of $l(i,j)$ may be computed in a row major order or column major order of $L[M,N]$.
C. The values of $l(i,j)$ cannot be computed in either row major order or column major order of $L[M,N]$.
D. $L[p,q]$ needs to be computed before $L[r,s]$ if either $p < r$ or $q < s$.

gate2009 normal algorithms dynamic-programming recursion

1.4.5 Dynamic Programming: GATE2010-34

https://gateoverflow.in/2208

The weight of a sequence $a_0, a_1, \ldots, a_{n-1}$ of real numbers is defined as $a_0 + a_1/2 + \cdots + a_{n-1}/2^{n-1}$. A subsequence of a sequence is obtained by deleting some elements from the sequence, keeping the order of the remaining elements the same. Let X denote the maximum possible weight of a subsequence of $a_o, a_1, \ldots, a_{n-1}$ and Y the maximum possible weight of a subsequence of $a_1, a_2, \ldots, a_{n-1}$. Then X is equal to

© Copyright GATE Overflow. All rights reserved.

A. $max(Y, a_0 + Y)$
B. $max(Y, a_0 + Y/2)$
C. $max(Y, a_0 + 2Y)$
D. $a_0 + Y/2$

gate2010 algorithms dynamic-programming normal

1.4.6 Dynamic Programming: GATE2011-25

https://gateoverflow.in/2127

An algorithm to find the length of the longest monotonically increasing sequence of numbers in an array $A[0 : n-1]$ is given below.

Let L_i, denote the length of the longest monotonically increasing sequence starting at index i in the array.

Initialize $L_{n-1} = 1$.

For all i such that $0 \leq i \leq n-2$

$$L_i = \begin{cases} 1 + L_{i+1} & \text{if A[i] < A[i+1]} \\ 1 & \text{Otherwise} \end{cases}$$

Finally, the length of the longest monotonically increasing sequence is $\max(L_0, L_1, \ldots, L_{n-1})$.

Which of the following statements is **TRUE**?

A. The algorithm uses dynamic programming paradigm
B. The algorithm has a linear complexity and uses branch and bound paradigm
C. The algorithm has a non-linear polynomial complexity and uses branch and bound paradigm
D. The algorithm uses divide and conquer paradigm

gate2011 algorithms easy dynamic-programming

1.4.7 Dynamic Programming: GATE2011-38

https://gateoverflow.in/2140

Four Matrices M_1, M_2, M_3 and M_4 of dimensions $p \times q$, $q \times r$, $r \times s$ and $s \times t$ respectively can be multiplied in several ways with different number of total scalar multiplications. For example when multiplied as $((M_1 \times M_2) \times (M_3 \times M_4))$, the total number of scalar multiplications is $pqr + rst + prt$. When multiplied as $(((M_1 \times M_2) \times M_3) \times M_4)$, the total number of scalar multiplications is $pqr + prs + pst$.

If $p = 10, q = 100, r = 20, s = 5$ and $t = 80$, then the minimum number of scalar multiplications needed is

A. 248000 B. 44000 C. 19000 D. 25000

gate2011 algorithms dynamic-programming normal

1.4.8 Dynamic Programming: GATE2014-2-37

https://gateoverflow.in/1996

Consider two strings A="qpqrr" and B="pqprqrp". Let x be the length of the longest common subsequence *(not necessarily contiguous)* between A and B and let y be the number of such longest common subsequences between A and B. Then $x + 10y =$ ___.

gate2014-2 algorithms normal numerical-answers dynamic-programming

1.4.9 Dynamic Programming: GATE2014-3-37

https://gateoverflow.in/2071

Suppose you want to move from 0 to 100 on the number line. In each step, you either move right by a unit distance or you take a *shortcut*. A shortcut is simply a pre-specified pair of integers i, j with $i < j$. Given a shortcut (i, j), if you are at position i on the number line, you may directly move to j. Suppose $T(k)$ denotes the smallest number of steps needed to move from k to 100. Suppose further that there is at most 1 shortcut involving any number, and in particular, from 9 there is a shortcut to 15. Let y and z be such that $T(9) = 1 + \min(T(y), T(z))$. Then the value of the product yz is _____.

gate2014-3 algorithms normal numerical-answers dynamic-programming

1.4.10 Dynamic Programming: GATE2016-2-14

https://gateoverflow.in/39570

The Floyd-Warshall algorithm for all-pair shortest paths computation is based on

A. Greedy paradigm.
B. Divide-and-conquer paradigm.
C. Dynamic Programming paradigm.
D. Neither Greedy nor Divide-and-Conquer nor Dynamic Programming paradigm.

© Copyright GATE Overflow. All rights reserved.

gate2016-2 algorithms dynamic-programming easy

1.4.11 Dynamic Programming: GATE2016-2-38

https://gateoverflow.in/39587

Let A_1, A_2, A_3 and A_4 be four matrices of dimensions $10 \times 5, 5 \times 20, 20 \times 10$ and 10×5, respectively. The minimum number of scalar multiplications required to find the product $A_1A_2A_3A_4$ using the basic matrix multiplication method is _____.

gate2016-2 dynamic-programming algorithms normal numerical-answers

1.4.12 Dynamic Programming: GATE2018-31

https://gateoverflow.in/204105

Assume that multiplying a matrix G_1 of dimension $p \times q$ with another matrix G_2 of dimension $q \times r$ requires pqr scalar multiplications. Computing the product of n matrices $G_1G_2G_3 \ldots G_n$ can be done by parenthesizing in different ways. Define G_iG_{i+1} as an **explicitly computed pair** for a given paranthesization if they are directly multiplied. Fr example, in the matrix multiplication chain $G_1G_2G_3G_4G_5G_6$ using parenthesization $(G_1(G_2G_3))(G_4(G_5G_6)), G_2G_3$ and G_5G_6 are only explicitly computed pairs.

Consider a matrix multiplication chain $F_1F_2F_3F_4F_5$, where matrices F_1, F_2, F_3, F_4 and F_5 are of dimensions $2 \times 25, 25 \times 3, 3 \times 16, 16 \times 1$ and 1×1000, respectively. In the parenthesization of $F_1F_2F_3F_4F_5$ that minimizes the total number of scalar multiplications, the explicitly computed pairs is/are

Explicitly computed pairs is (F3, F4)

A. F_1F_2 and F_3F_4 only
B. F_2F_3 only
C. F_3F_4 only
D. F_2F_2 and F_4F_5 only

gate2018 algorithms dynamic-programming

1.5 Graph Algorithms (48)

1.5.1 Graph Algorithms: GATE1994-1.22

https://gateoverflow.in/2465

Which of the following statements is false?

A. Optimal binary search tree construction can be performed efficiently using dynamic programming
B. Breadth-first search cannot be used to find connected components of a graph
C. Given the prefix and postfix walks over a binary tree, the binary tree cannot be uniquely constructed.
D. Depth-first search can be used to find connected components of a graph

gate1994 algorithms normal graph-algorithms

1.5.2 Graph Algorithms: GATE1994-24

https://gateoverflow.in/2520

An independent set in a graph is a subset of vertices such that no two vertices in the subset are connected by an edge. An incomplete scheme for a greedy algorithm to find a maximum independent set in a tree is given below:

```
V: Set of all vertices in the tree;
I := ϕ
while V ≠ ϕ do
begin
    select a vertex u ∈ V such that
    _______;
    V := V - {u};
    if u is such that
    _______then I := I U {u}
end;
Output(I);
```

a. Complete the algorithm by specifying the property of vertex u in each case.
b. What is the time complexity of the algorithm?

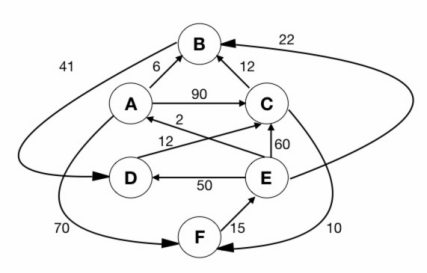
gate1994 algorithms graph-algorithms normal

1.5.3 Graph Algorithms: GATE1996-17

https://gateoverflow.in/2769

Let G be the directed, weighted graph shown in below figure

© Copyright GATE Overflow. All rights reserved.



We are interested in the shortest paths from A.

a. Output the sequence of vertices identified by the Dijkstra's algorithm for single source shortest path when the algorithm is started at node A
b. Write down sequence of vertices in the shortest path from A to E
c. What is the cost of the shortest path from A to E?

gate1996 algorithms graph-algorithms normal

1.5.4 Graph Algorithms: GATE1998-1.21, ISRO2008-16

https://gateoverflow.in/1658

Which one of the following algorithm design techniques is used in finding all pairs of shortest distances in a graph?

A. Dynamic programming
B. Backtracking
C. Greedy
D. Divide and Conquer

gate1998 algorithms graph-algorithms easy isro2008

1.5.5 Graph Algorithms: GATE2000-1.13

https://gateoverflow.in/636

The most appropriate matching for the following pairs

X: depth first search	1: heap
Y: breadth first search	2: queue
Z: sorting	3: stack

is:

A. X - 1, Y - 2, Z - 3
B. X - 3, Y - 1, Z - 2
C. X - 3, Y - 2, Z - 1
D. X - 2, Y - 3, Z - 1

gate2000 algorithms easy graph-algorithms

1.5.6 Graph Algorithms: GATE2001-2.14

https://gateoverflow.in/732

Consider an undirected, unweighted graph G. Let a breadth-first traversal of G be done starting from a node r. Let $d(r,u)$ and $d(r,v)$ be the lengths of the shortest paths from r to u and v respectively in G. If u is visited before v during the breadth-first traversal, which of the following statements is correct?

A. $d(r,u) < d(r,v)$
B. $d(r,u) > d(r,v)$
C. $d(r,u) \leq d(r,v)$
D. None of the above

gate2001 algorithms graph-algorithms normal

1.5.7 Graph Algorithms: GATE2002-12

https://gateoverflow.in/865

Fill in the blanks in the following template of an algorithm to compute all pairs shortest path lengths in a directed graph G with $n * n$ adjacency matrix A. $A[i,j]$ equals 1 if there is an edge in G from i to j, and 0 otherwise. Your aim in filling in the blanks is to ensure that the algorithm is correct.

```
INITIALIZATION: For i = 1 ... n
    {For j = 1 ... n
        { if a[i,j] = 0 then P[i,j] =_______ else P[i,j] =_______;}
    }

ALGORITHM: For i = 1 ... n
    {For j = 1 ... n
        {For k = 1 ... n
            {P[__,__] = min{_______,______}; }
```

© Copyright GATE Overflow. All rights reserved.

```
        }
    }
```

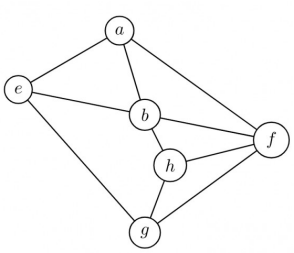
a. Copy the complete line containing the blanks in the Initialization step and fill in the blanks.
b. Copy the complete line containing the blanks in the Algorithm step and fill in the blanks.
c. Fill in the blank: The running time of the Algorithm is $O(\_\_\_)$.

gate2002 algorithms graph-algorithms time-complexity normal descriptive

1.5.8 Graph Algorithms: GATE2003-21

https://gateoverflow.in/911

Consider the following graph:



Among the following sequences:

I. abeghf
II. abfehg
III. abfhge
IV. afghbe

Which are the depth-first traversals of the above graph?

A. I, II and IV only B. I and IV only C. II, III and IV only D. I, III and IV only

gate2003 algorithms graph-algorithms normal

1.5.9 Graph Algorithms: GATE2003-67

https://gateoverflow.in/954

Let $G=(V,E)$ be an undirected graph with a subgraph $G_1=(V_1,E_1)$. Weights are assigned to edges of G as follows.

$$w(e)=\begin{cases}0, & \text{if } e\in E_1\\ 1, & \text{otherwise}\end{cases}$$

A single-source shortest path algorithm is executed on the weighted graph (V,E,w) with an arbitrary vertex v_1 of V_1 as the source. Which of the following can always be inferred from the path costs computed?

A. The number of edges in the shortest paths from v_1 to all vertices of G
B. G_1 is connected
C. V_1 forms a clique in G
D. G_1 is a tree

gate2003 algorithms graph-algorithms normal

1.5.10 Graph Algorithms: GATE2003-70

https://gateoverflow.in/957

Let $G=(V,E)$ be a directed graph with n vertices. A path from v_i to v_j in G is a sequence of vertices ($v_i,v_{i+1},\ldots,v_j$) such that $(v_k,v_k+1)\in E$ for all k in i through $j-1$. A simple path is a path in which no vertex appears more than once.

Let A be an $n\times n$ array initialized as follows:

$$A[j,k]=\begin{cases}1, & \text{if } (j,k)\in E\\ 0, & \text{otherwise}\end{cases}$$

Consider the following algorithm:

© Copyright GATE Overflow. All rights reserved.

```
for i=1 to n
    for j=1 to n
        for k=1 to n
            A[j,k] = max(A[j,k], A[j,i] + A[i,k]);
```

Which of the following statements is necessarily true for all j and k after termination of the above algorithm?

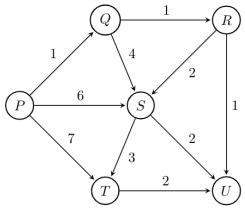
A. $A[j,k] \leq n$
B. If $A[j,j] \geq n-1$ then G has a Hamiltonian cycle
C. If there exists a path from j to k, $A[j,k]$ contains the longest path length from j to k
D. If there exists a path from j to k, every simple path from j to k contains at most $A[j,k]$ edges

gate2003 algorithms graph-algorithms normal

1.5.11 Graph Algorithms: GATE2004-44

https://gateoverflow.in/1041

Suppose we run Dijkstra's single source shortest path algorithm on the following edge-weighted directed graph with vertex P as the source.



In what order do the nodes get included into the set of vertices for which the shortest path distances are finalized?

A. P,Q,R,S,T,U B. P,Q,R,U,S,T C. P,Q,R,U,T,S D. P,Q,T,R,U,S

gate2004 algorithms graph-algorithms normal

1.5.12 Graph Algorithms: GATE2004-81

https://gateoverflow.in/1075

Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be connected graphs on the same vertex set V with more than two vertices. If $G_1 \cap G_2 = (V, E_1 \cap E_2)$ is not a connected graph, then the graph $G_1 \cup G_2 = (V, E_1 \cup E_2)$

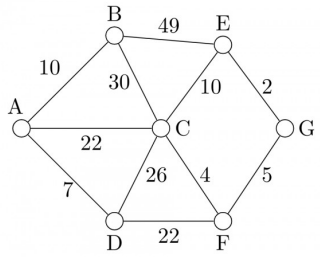
A. cannot have a cut vertex
B. must have a cycle
C. must have a cut-edge (bridge)
D. has chromatic number strictly greater than those of G_1 and G_2

gate2004 algorithms graph-algorithms normal

1.5.13 Graph Algorithms: GATE2004-IT-56

https://gateoverflow.in/3699

Consider the undirected graph below:



Using Prim's algorithm to construct a minimum spanning tree starting with node A, which one of the following sequences of edges represents a possible order in which the edges would be added to construct the minimum spanning tree?

A. (E, G), (C, F), (F, G), (A, D), (A, B), (A, C)
B. (A, D), (A, B), (A, C), (C, F), (G, E), (F, G)
C. (A, B), (A, D), (D, F), (F, G), (G, E), (F, C)
D. (A, D), (A, B), (D, F), (F, C), (F, G), (G, E)

gate2004-it algorithms graph-algorithms normal

© Copyright GATE Overflow. All rights reserved.

1.5.14 Graph Algorithms: GATE2005-38

https://gateoverflow.in/1374

Let $G(V,E)$ be an undirected graph with positive edge weights. Dijkstra's single source shortest path algorithm can be implemented using the binary heap data structure with time complexity:

A. $O\left(|V|^2\right)$
B. $O(|E|+|V|\log|V|)$
C. $O(|V|\log|V|)$
D. $O((|E|+|V|)\log|V|)$

gate2005 algorithms graph-algorithms normal

1.5.15 Graph Algorithms: GATE2005-82a

https://gateoverflow.in/1404

Let s and t be two vertices in a undirected graph $G=(V,E)$ having distinct positive edge weights. Let $[X,Y]$ be a partition of V such that $s\in X$ and $t\in Y$. Consider the edge e having the minimum weight amongst all those edges that have one vertex in X and one vertex in Y.

The edge e must definitely belong to:

A. the minimum weighted spanning tree of G
B. the weighted shortest path from s to t
C. each path from s to t
D. the weighted longest path from s to t

gate2005 algorithms graph-algorithms normal

1.5.16 Graph Algorithms: GATE2005-82b

https://gateoverflow.in/82129

Let s and t be two vertices in a undirected graph $G=(V,E)$ having distinct positive edge weights. Let $[X,Y]$ be a partition of V such that $s\in X$ and $t\in Y$. Consider the edge e having the minimum weight amongst all those edges that have one vertex in X and one vertex in Y.

Let the weight of an edge e denote the congestion on that edge. The congestion on a path is defined to be the maximum of the congestions on the edges of the path. We wish to find the path from s to t having minimum congestion. Which of the following paths is always such a path of minimum congestion?

A. a path from s to t in the minimum weighted spanning tree
B. a weighted shortest path from s to t
C. an Euler walk from s to t
D. a Hamiltonian path from s to t

gate2005 algorithms graph-algorithms normal

1.5.17 Graph Algorithms: GATE2005-IT-14

https://gateoverflow.in/3759

In a depth-first traversal of a graph G with n vertices, k edges are marked as tree edges. The number of connected components in G is

A. k
B. $k+1$
C. $n-k-1$
D. $n-k$

gate2005-it algorithms graph-algorithms normal

1.5.18 Graph Algorithms: GATE2005-IT-15

https://gateoverflow.in/3760

In the following table, the left column contains the names of standard graph algorithms and the right column contains the time complexities of the algorithms. Match each algorithm with its time complexity.

1. Bellman-Ford algorithm	A: $O(m\log n)$
2. Kruskal's algorithm	B: $O(n^3)$
3. Floyd-Warshall algorithm	C: $O(nm)$
4. Topological sorting	D: $O(n+m)$

A. $1\rightarrow$ C, $2\rightarrow$ A, $3\rightarrow$ B, $4\rightarrow$ D
B. $1\rightarrow$ B, $2\rightarrow$ D, $3\rightarrow$ C, $4\rightarrow$ A
C. $1\rightarrow$ C, $2\rightarrow$ D, $3\rightarrow$ A, $4\rightarrow$ B
D. $1\rightarrow$ B, $2\rightarrow$ A, $3\rightarrow$ C, $4\rightarrow$ D

gate2005-it algorithms graph-algorithms normal

1.5.19 Graph Algorithms: GATE2005-IT-84a

https://gateoverflow.in/3856

A sink in a directed graph is a vertex i such that there is an edge from every vertex $j\neq i$ to i and there is no edge from i to any other vertex. A directed graph G with n vertices is represented by its adjacency matrix A, where $A[i][j]=1$ if there is an edge directed from vertex i to j and 0 otherwise. The following algorithm determines whether there is a sink in the graph G.

© Copyright GATE Overflow. All rights reserved.

```
i = 0;
do {
    j = i + 1;
    while ((j < n) && E1) j++;
    if (j < n) E2;
} while (j < n);
flag = 1;
for (j = 0; j < n; j++)
    if ((j! = i) && E3) flag = 0;
if (flag) printf("Sink exists");
else printf ("Sink does not exist");
```

Choose the correct expressions for E_1 and E_2

A. $E_1 : A[i][j]$ and $E_2 : i = j$;
B. $E_1 : !A[i][j]$ and $E_2 : i = j+1$;
C. $E_1 : !A[i][j]$ and $E_2 : i = j$;
D. $E_1 : A[i][j]$ and $E_2 : i = j+1$;

gate2005-it algorithms graph-algorithms normal

1.5.20 Graph Algorithms: GATE2005-IT-84b

https://gateoverflow.in/3857

A sink in a directed graph is a vertex i such that there is an edge from every vertex $j \neq i$ to i and there is no edge from i to any other vertex. A directed graph G with n vertices is represented by its adjacency matrix A, where $A[i][j] = 1$ if there is an edge directed from vertex i to j and 0 otherwise. The following algorithm determines whether there is a sink in the graph G.

```
i = 0;
do {
    j = i + 1;
    while ((j < n) && E1) j++;
    if (j < n) E2;
} while (j < n);
flag = 1;
for (j = 0; j < n; j++)
    if ((j! = i) && E3) flag = 0;
if (flag) printf("Sink exists") ;
else printf ("Sink does not exist");
```

Choose the correct expression for E_3

A. $(A[i][j]$ && $!A[j][i])$
B. $(!A[i][j]$ && $A[j][i])$
C. $(!A[i][j]$ || $A[j][i])$
D. $(A[i][j]$ || $!A[j][i])$

gate2005-it algorithms graph-algorithms normal

1.5.21 Graph Algorithms: GATE2006-12

https://gateoverflow.in/891

To implement Dijkstra's shortest path algorithm on unweighted graphs so that it runs in linear time, the data structure to be used is:

A. Queue B. Stack C. Heap D. B-Tree

gate2006 algorithms graph-algorithms easy

1.5.22 Graph Algorithms: GATE2006-48

https://gateoverflow.in/1824

Let T be a depth first search tree in an undirected graph G. Vertices u and v are leaves of this tree T. The degrees of both u and v in G are at least 2. which one of the following statements is true?

A. There must exist a vertex w adjacent to both u and v in G
B. There must exist a vertex w whose removal disconnects u and v in G
C. There must exist a cycle in G containing u and v
D. There must exist a cycle in G containing u and all its neighbours in G

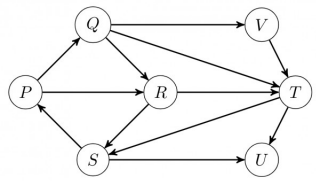
gate2006 algorithms graph-algorithms normal

1.5.23 Graph Algorithms: GATE2006-IT-46

https://gateoverflow.in/3589

Which of the following is the correct decomposition of the directed graph given below into its strongly connected components?

© Copyright GATE Overflow. All rights reserved.



A. $\{P,Q,R,S\},\{T\},\{U\},\{V\}$
B. $\{P,Q,R,S,T,V\},\{U\}$
C. $\{P,Q,S,T,V\},\{R\},\{U\}$
D. $\{P,Q,R,S,T,U,V\}$

gate2006-it algorithms graph-algorithms normal

1.5.24 Graph Algorithms: GATE2006-IT-47

https://gateoverflow.in/3590

Consider the depth-first-search of an undirected graph with 3 vertices P, Q, and R. Let discovery time $d(u)$ represent the time instant when the vertex u is first visited, and finish time $f(u)$ represent the time instant when the vertex u is last visited. Given that

$d(P) = 5$ units	$f(P) = 12$ units
$d(Q) = 6$ units	$f(Q) = 10$ units
$d(R) = 14$ unit	$f(R) = 18$ units

Which one of the following statements is TRUE about the graph?

A. There is only one connected component
B. There are two connected components, and P and R are connected
C. There are two connected components, and Q and R are connected
D. There are two connected components, and P and Q are connected

gate2006-it algorithms graph-algorithms normal

1.5.25 Graph Algorithms: GATE2007-41

https://gateoverflow.in/1239

In an unweighted, undirected connected graph, the shortest path from a node S to every other node is computed most efficiently, in terms of *time complexity*, by

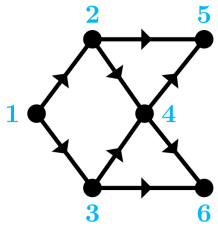
A. Dijkstra's algorithm starting from S.
B. Warshall's algorithm.
C. Performing a DFS starting from S.
D. Performing a BFS starting from S.

gate2007 algorithms graph-algorithms easy

1.5.26 Graph Algorithms: GATE2007-5

https://gateoverflow.in/31821

Consider the DAG with $V = \{1,2,3,4,5,6\}$ shown below.



Which of the following is not a topological ordering?

A. $1\,2\,3\,4\,5\,6$ B. $1\,3\,2\,4\,5\,6$ C. $1\,3\,2\,4\,6\,5$ D. $3\,2\,4\,1\,6\,5$

gate2007 algorithms graph-algorithms

© Copyright GATE Overflow. All rights reserved.

1.5.27 Graph Algorithms: GATE2007-IT-24

https://gateoverflow.in/3457

A depth-first search is performed on a directed acyclic graph. Let $d[u]$ denote the time at which vertex u is visited for the first time and $f[u]$ the time at which the DFS call to the vertex u terminates. Which of the following statements is always TRUE for all edges (u,v) in the graph ?

A. $d[u] < d[v]$
B. $d[u] < f[v]$
C. $f[u] < f[v]$
D. $f[u] > f[v]$

gate2007-it algorithms graph-algorithms normal

1.5.28 Graph Algorithms: GATE2007-IT-3, UGCNET-June2012-III-34

https://gateoverflow.in/3434

Consider a weighted, undirected graph with positive edge weights and let uv be an edge in the graph. It is known that the shortest path from the source vertex s to u has weight 53 and the shortest path from s to v has weight 65. Which one of the following statements is always TRUE?

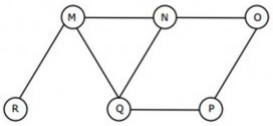
A. Weight $(u,v) \leq 12$
B. Weight $(u,v) = 12$
C. Weight $(u,v) \geq 12$
D. Weight $(u,v) > 12$

gate2007-it algorithms graph-algorithms normal ugcnetjune2012iii

1.5.29 Graph Algorithms: GATE2008-19

https://gateoverflow.in/417

The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is:

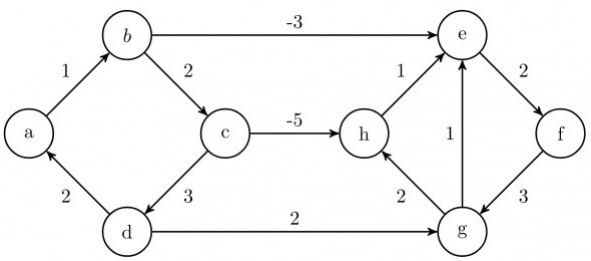


A. $MNOPQR$ B. $NQMPOR$ C. $QMNPRO$ D. $QMNPOR$

gate2008 normal algorithms graph-algorithms

1.5.30 Graph Algorithms: GATE2008-45

https://gateoverflow.in/457



Dijkstra's single source shortest path algorithm when run from vertex a in the above graph, computes the correct shortest path distance to

A. only vertex a
B. only vertices a, e, f, g, h
C. only vertices a, b, c, d
D. all the vertices

gate2008 algorithms graph-algorithms normal

1.5.31 Graph Algorithms: GATE2008-7

https://gateoverflow.in/405

The most efficient algorithm for finding the number of connected components in an undirected graph on n vertices and m edges has time complexity

A. $\Theta(n)$ B. $\Theta(m)$ C. $\Theta(m+n)$ D. $\Theta(mn)$

gate2008 algorithms graph-algorithms time-complexity normal

1.5.32 Graph Algorithms: GATE2008-IT-47

https://gateoverflow.in/3357

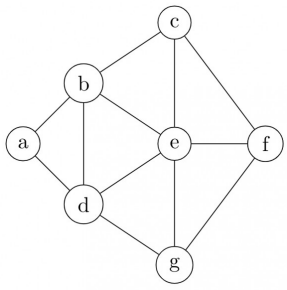
Consider the following sequence of nodes for the undirected graph given below:

1. $a\ b\ e\ f\ d\ g\ c$

© Copyright GATE Overflow. All rights reserved.

2. $a\,b\,e\,f\,c\,g\,d$
3. $a\,d\,g\,e\,b\,c\,f$
4. $a\,d\,b\,c\,g\,e\,f$

A Depth First Search (DFS) is started at node a. The nodes are listed in the order they are first visited. Which of the above is/are possible output(s)?



A. 1 and 3 only B. 2 and 3 only C. $2,3$ and 4 only D. $1,2$ and 3 only

gate2008-it algorithms graph-algorithms normal

1.5.33 Graph Algorithms: GATE2009-13

https://gateoverflow.in/1305

Which of the following statement(s) is/are correct regarding Bellman-Ford shortest path algorithm?

P: Always finds a negative weighted cycle, if one exists.

Q: Finds whether any negative weighted cycle is reachable from the source.

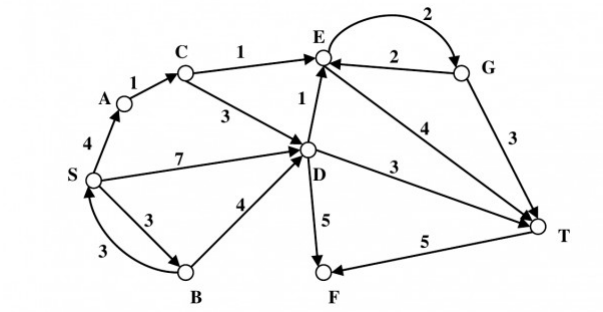
A. P only B. Q only C. Both P and Q D. Neither P nor Q

gate2009 algorithms graph-algorithms normal

1.5.34 Graph Algorithms: GATE2012-40

https://gateoverflow.in/1765

Consider the directed graph shown in the figure below. There are multiple shortest paths between vertices S and T. Which one will be reported by Dijkstra's shortest path algorithm? Assume that, in any iteration, the shortest path to a vertex v is updated only when a strictly shorter path to v is discovered.



A. SDT B. SBDT C. SACDT D. SACET

gate2012 algorithms graph-algorithms normal

1.5.35 Graph Algorithms: GATE2013-19

https://gateoverflow.in/1441

What is the time complexity of Bellman-Ford single-source shortest path algorithm on a complete graph of n vertices?

A. $\theta(n^2)$ B. $\theta(n^2 \log n)$
C. $\theta(n^3)$ D. $\theta(n^3 \log n)$

gate2013 algorithms graph-algorithms normal

1.5.36 Graph Algorithms: GATE2014-1-11

https://gateoverflow.in/1771

Let G be a graph with n vertices and m edges.What is the tightest upper bound on the running time of Depth First Search on G, when G is represented as an adjacency matrix?

A. $\Theta(n)$ B. $\Theta(n+m)$ C. $\Theta(n^2)$ D. $\Theta(m^2)$

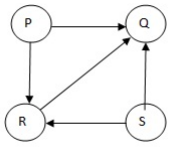
© Copyright GATE Overflow. All rights reserved.

gate2014-1 algorithms graph-algorithms normal

1.5.37 Graph Algorithms: GATE2014-1-13

https://gateoverflow.in/1779

Consider the directed graph below given.



Which one of the following is **TRUE**?

A. The graph does not have any topological ordering.
B. Both PQRS and SRQP are topological orderings.
C. Both PSRQ and SPRQ are topological orderings.
D. PSRQ is the only topological ordering.

gate2014-1 graph-algorithms easy

1.5.38 Graph Algorithms: GATE2014-2-14

https://gateoverflow.in/1969

Consider the tree arcs of a BFS traversal from a source node W in an unweighted, connected, undirected graph. The tree T formed by the tree arcs is a data structure for computing

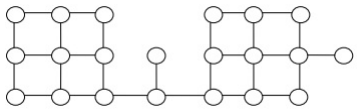
A. the shortest path between every pair of vertices.
B. the shortest path from W to every vertex in the graph.
C. the shortest paths from W to only those nodes that are leaves of T.
D. the longest path in the graph.

gate2014-2 algorithms graph-algorithms normal

1.5.39 Graph Algorithms: GATE2014-3-13

https://gateoverflow.in/2047

Suppose depth first search is executed on the graph below starting at some unknown vertex. Assume that a recursive call to visit a vertex is made only after first checking that the vertex has not been visited earlier. Then the maximum possible recursion depth (including the initial call) is _____.



gate2014-3 algorithms graph-algorithms numerical-answers normal

1.5.40 Graph Algorithms: GATE2015-1-45

https://gateoverflow.in/8321

Let $G = (V, E)$ be a simple undirected graph, and s be a particular vertex in it called the source. For $x \in V$, let $d(x)$ denote the shortest distance in G from s to x. A breadth first search (BFS) is performed starting at s. Let T be the resultant BFS tree. If (u, v) is an edge of G that is not in T, then which one of the following CANNOT be the value of $d(u) - d(v)$?

A. -1 B. 0 C. 1 D. 2

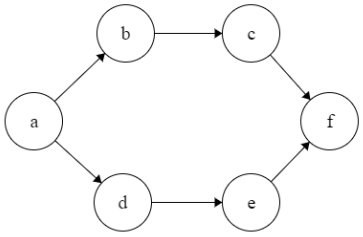
gate2015-1 algorithms graph-algorithms normal

1.5.41 Graph Algorithms: GATE2016-1-11

https://gateoverflow.in/39669

Consider the following directed graph:

© Copyright GATE Overflow. All rights reserved.



The number of different topological orderings of the vertices of the graph is _____.

gate2016-1 algorithms graph-algorithms normal numerical-answers

1.5.42 Graph Algorithms: GATE2016-2-11

https://gateoverflow.in/39563

Breadth First Search (BFS) is started on a binary tree beginning from the root vertex. There is a vertex t at a distance four from the root. If t is the n^{th} vertex in this BFS traversal, then the maximum possible value of n is _____

gate2016-2 algorithms graph-algorithms normal numerical-answers

1.5.43 Graph Algorithms: GATE2016-2-41

https://gateoverflow.in/39620

In an adjacency list representation of an undirected simple graph $G = (V, E)$, each edge (u, v) has two adjacency list entries: $[v]$ in the adjacency list of u, and $[u]$ in the adjacency list of v. These are called twins of each other. A twin pointer is a pointer from an adjacency list entry to its twin. If $|E| = m$ and $|V| = n$, and the memory size is not a constraint, what is the time complexity of the most efficient algorithm to set the twin pointer in each entry in each adjacency list?

A. $\Theta(n^2)$
B. $\Theta(n+m)$
C. $\Theta(m^2)$
D. $\Theta(n^4)$

gate2016-2 algorithms graph-algorithms normal

1.5.44 Graph Algorithms: GATE2017-1-26

https://gateoverflow.in/118306

Let $G = (V, E)$ be *any* connected, undirected, edge-weighted graph. The weights of the edges in E are positive and distinct. Consider the following statements:

I. Minimum Spanning Tree of G is always unique.
II. Shortest path between any two vertices of G is always unique.

Which of the above statements is/are necessarily true?

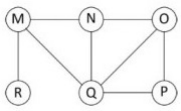
A. I only B. II only C. both I and II D. neither I nor II

gate2017-1 algorithms graph-algorithms normal

1.5.45 Graph Algorithms: GATE2017-2-15

https://gateoverflow.in/118196

The Breadth First Search (BFS) algorithm has been implemented using the queue data structure. Which one of the following is a possible order of visiting the nodes in the graph below?



A. MNOPQR B. NQMPOR C. QMNROP D. POQNMR

gate2017-2 algorithms graph-algorithms

1.5.46 Graph Algorithms: Gate2000-2.19

https://gateoverflow.in/4208

Let G be an undirected graph. Consider a depth-first traversal of G, and let T be the resulting depth-first search tree. Let u be a vertex in G and let v be the first new (unvisited) vertex visited after visiting u in the traversal. Which of the following statement is always true?

A. $\{u, v\}$ must be an edge in G, and u is a descendant of v in T
B. $\{u, v\}$ must be an edge in G, and v is a descendant of u in T

© Copyright GATE Overflow. All rights reserved.

C. If $\{u,v\}$ is not an edge in G then u is a leaf in T
D. If $\{u,v\}$ is not an edge in G then u and v must have the same parent in T

gate2000 algorithms graph-algorithms normal

1.5.47 Graph Algorithms: TIFR2013-B-5

https://gateoverflow.in/25666

Given a weighted directed graph with n vertices where edge weights are integers (positive, zero, or negative), determining whether there are paths of arbitrarily large weight can be performed in time

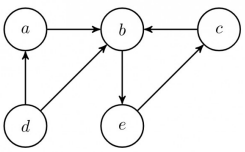
A. $O(n)$
B. $O(n.\log(n))$ but not $O(n)$
C. $O(n^{1.5})$ but not $O(n \log n)$
D. $O(n^3)$ but not $O(n^{1.5})$
E. $O(2^n)$ but not $O(n^3)$

tifr2013 algorithms graph-algorithms

1.5.48 Graph Algorithms: TIFR2014-B-3

https://gateoverflow.in/27137

Consider the following directed graph.



Suppose a depth-first traversal of this graph is performed, assuming that whenever there is a choice, the vertex earlier in the alphabetical order is to be chosen. Suppose the number of tree edges is T, the number of back edges is B and the number of cross edges is C. Then

a. $B = 1$, $C = 1$, and $T = 4$.
b. $B = 0$, $C = 2$, and $T = 4$.
c. $B = 2$, $C = 1$, and $T = 3$.
d. $B = 1$, $C = 2$, and $T = 3$.
e. $B = 2$, $C = 2$, and $T = 1$.

tifr2014 algorithms graph-algorithms

1.6 Graph Connectivity (1)

1.6.1 Graph Connectivity: GATE2018-43

https://gateoverflow.in/204117

Let G be a graph with 100! vertices, with each vertex labelled by a distinct permutation of the numbers $1,2,\ldots,100$. There is an edge between vertices u and v if and only if the label of u can be obtained by swapping two adjacent numbers in the label of v. Let y denote the degree of a vertex in G, and z denote the number of connected components in G. Then, $y + 10z =$ ____

gate2018 algorithms graph-algorithms graph-connectivity numerical-answers

1.7 Greedy Algorithm (7)

1.7.1 Greedy Algorithm: GATE1999-2.20

https://gateoverflow.in/466

The minimum number of record movements required to merge five files A (with 10 records), B (with 20 records), C (with 15 records), D (with 5 records) and E (with 25 records) is:

A. 165 B. 90 C. 75 D. 65

gate1999 algorithms normal greedy-algorithm

1.7.2 Greedy Algorithm: GATE2003-69

https://gateoverflow.in/956

The following are the starting and ending times of activities A,B,C,D,E,F,G and H respectively in chronological order: "$a_s\ b_s\ c_s\ a_e\ d_s\ c_e\ e_s\ f_s\ b_e\ d_e\ g_s\ e_e\ f_e\ h_s\ g_e\ h_e$" . Here, x_s denotes the starting time and x_e denotes the ending time of activity X. We need to schedule the activities in a set of rooms available to us. An activity can be scheduled in a room only if the room is reserved for the activity for its entire duration. What is the minimum number of rooms required?

A. 3 B. 4 C. 5 D. 6

gate2003 algorithms normal greedy-algorithm

© Copyright GATE Overflow. All rights reserved.

1.7.3 Greedy Algorithm: GATE2005-84a

https://gateoverflow.in/1406

We are given 9 tasks $T_1, T_2, \ldots, T_9$. The execution of each task requires one unit of time. We can execute one task at a time. Each task T_i has a profit P_i and a deadline d_i. Profit P_i is earned if the task is completed before the end of the d_i^{th} unit of time.

Task	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
Profit	15	20	30	18	18	10	23	16	25
Deadline	7	2	5	3	4	5	2	7	3

Are all tasks completed in the schedule that gives maximum profit?

A. All tasks are completed
B. T_1 and T_6 are left out
C. T_1 and T_8 are left out
D. T_4 and T_6 are left out

gate2005 algorithms greedy-algorithm process-schedule normal

1.7.4 Greedy Algorithm: GATE2005-84b

https://gateoverflow.in/82514

We are given 9 tasks $T_1, T_2, \ldots, T_9$. The execution of each task requires one unit of time. We can execute one task at a time. Each task T_i has a profit P_i and a deadline d_i. Profit P_i is earned if the task is completed before the end of the d_i^{th} unit of time.

Task	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
Profit	15	20	30	18	18	10	23	16	25
Deadline	7	2	5	3	4	5	2	7	3

What is the maximum profit earned?

A. 147 B. 165 C. 167 D. 175

gate2005 algorithms greedy-algorithm process-schedule normal

1.7.5 Greedy Algorithm: GATE2006-IT-48

https://gateoverflow.in/3591

The characters a to h have the set of frequencies based on the first 8 Fibonacci numbers as follows

$a : 1, b : 1, c : 2, d : 3, e : 5, f : 8, g : 13, h : 21$

A Huffman code is used to represent the characters. What is the sequence of characters corresponding to the following code?

110111100111010

A. $fdheg$ B. $ecgdf$ C. $dchfg$ D. $fehdg$

gate2006-it algorithms greedy-algorithm normal

1.7.6 Greedy Algorithm: GATE2007-76

https://gateoverflow.in/1271

Suppose the letters a, b, c, d, e, f have probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{32}$, respectively.

Which of the following is the Huffman code for the letter a, b, c, d, e, f ?

A. 0, 10, 110, 1110, 11110, 11111
B. 11, 10, 011, 010, 001, 000
C. 11, 10, 01, 001, 0001, 0000
D. 110, 100, 010, 000, 001, 111

gate2007 algorithms greedy-algorithm normal

1.7.7 Greedy Algorithm: GATE2018-48

https://gateoverflow.in/204123

Consider the weights and values of items listed below. Note that there is only one unit of each item.

Item number	Weight (in Kgs)	Value (in rupees)
1	10	60
2	7	28
3	4	20
4	2	24

© Copyright GATE Overflow. All rights reserved.

The task is to pick a subset of these items such that their total weight is no more than 11 Kgs and their total value is maximized. Moreover, no item may be split. The total value of items picked by an optimal algorithm is denoted by V_{opt}. A greedy algorithm sorts the items by their value-to-weight ratios in descending order and packs them greedily, starting from the first item in the ordered list. The total value of items picked by the greedy algorithm is denoted by V_{greedy}.

The value of $V_{opt} - V_{greedy}$ is ____

gate2018 algorithms greedy-algorithm numerical-answers

1.8 Hashing (1)

1.8.1 Hashing: GATE1990-13b

https://gateoverflow.in/86225

Consider a hash table with chaining scheme for overflow handling:

i. What is the worst-case timing complexity of inserting n elements into such a table?
ii. For what type of instance does this hashing scheme take the worst-case time for insertion?

gate1990 hashing algorithms

1.9 Huffman Code (3)

1.9.1 Huffman Code: GATE1989-13a

https://gateoverflow.in/93172

A language uses an alphabet of six letters, $\{a,b,c,d,e,f\}$. The relative frequency of use of each letter of the alphabet in the language is as given below:

LETTER	RELATIVE FREQUENCY OF USE
a	0.19
b	0.05
c	0.17
d	0.08
e	0.40
f	0.11

Design a prefix binary code for the language which would minimize the average length of the encoded words of the language.

descriptive gate1989 algorithms huffman-code

1.9.2 Huffman Code: GATE2007-77

https://gateoverflow.in/43513

Suppose the letters a, b, c, d, e, f have probabilities $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{32}$, respectively.

What is the average length of the Huffman code for the letters a, b, c, d, e, f ?

A. 3 B. 2.1875 C. 2.25 D. 1.9375

gate2007 algorithms greedy-algorithm normal huffman-code

1.9.3 Huffman Code: GATE2017-2-50

https://gateoverflow.in/118395

A message is made up entirely of characters from the set $X = \{P,Q,R,S,T\}$. The table of probabilities for each of the characters is shown below:

Character	Probability
P	0.22
Q	0.34
R	0.17
S	0.19
T	0.08
Total	1.00

© Copyright GATE Overflow. All rights reserved.

If a message of 100 characters over X is encoded using Huffman coding, then the expected length of the encoded message in bits is _____.

gate2017-2 huffman-code numerical-answers algorithms

1.10 Identify Function (38)

1.10.1 Identify Function: GATE1989-8a

https://gateoverflow.in/89080

What is the output produced by the following program, when the input is "HTGATE"

```
Function what (s:string): string;
var n:integer;
begin
     n = s.length
     if n <= 1
     then what := s
     else what :=contact (what (substring (s, 2, n)), s.C [1])
end;
```

Note

i. type string=record
   length:integer;
   C:array[1..100] of char
   end
ii. Substring (s, i, j): this yields the string made up of the i^{th} through j^{th} characters in s; for appropriately defined in i and j.
iii. Contact (s_1, s_2): this function yields a string of length s_1 length + s_2 - length obtained by concatenating s_1 with s_2 such that s_1 precedes s_2.

gate1989 descriptive algorithms identify-function

1.10.2 Identify Function: GATE1990-11b

https://gateoverflow.in/85987

The following program computes values of a mathematical function $f(x)$. Determine the form of $f(x)$.

```
main ()
{
    int m, n; float x, y, t;
    scanf ("%f%d", &x, &n);
    t = 1; y = 0; m = 1;
    do
    {
        t *= (-x/m);
        y += t;
    } while (m++ < n);
    printf ("The value of y is %f", y);
}
```

gate1990 descriptive algorithms identify-function

1.10.3 Identify Function: GATE1991-03-viii

https://gateoverflow.in/523

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

Consider the following Pascal function:

```
Function X(M:integer):integer;
Var i:integer;
Begin
    i := 0;
    while i*i < M
    do i:= i+1
    X := i
 end
```

The function call $X(N)$, if N is positive, will return

A. $\lfloor \sqrt{N} \rfloor$
B. $\lfloor \sqrt{N} \rfloor + 1$
C. $\lceil \sqrt{N} \rceil$
D. $\lceil \sqrt{N} \rceil + 1$
E. None of the above

gate1991 algorithms easy identify-function

© Copyright GATE Overflow. All rights reserved.

1.10.4 Identify Function: GATE1993-7.4

https://gateoverflow.in/2292

What does the following code do?

```
var a, b: integer;
begin
    a:=a+b;
    b:=a-b;
    a:a-b;
end;
```

A. exchanges a and b
B. doubles a and stores in b
C. doubles b and stores in a
D. leaves a and b unchanged
E. none of the above

gate1993 algorithms identify-function easy

1.10.5 Identify Function: GATE1994-6

https://gateoverflow.in/2502

What function of x, n is computed by this program?

```
Function what(x, n:integer): integer:
Var
    value : integer
begin
    value := 1
    if n > 0 then
    begin
        if n mod 2 =1 then
            value := value * x;
        value := value * what(x*x, n div 2);
    end;
    what := value;
end;
```

gate1994 algorithms identify-function normal

1.10.6 Identify Function: GATE1995-1.4

https://gateoverflow.in/2591

In the following Pascal program segment, what is the value of X after the execution of the program segment?

```
X := -10; Y := 20;
If X > Y then if X < 0 then X := abs(X) else X := 2*X;
```

A. 10 B. -20 C. -10 D. None

gate1995 algorithms identify-function easy

1.10.7 Identify Function: GATE1995-2.3

https://gateoverflow.in/2615

Assume that X and Y are non-zero positive integers. What does the following Pascal program segment do?

```
while X <> Y do
if  X > Y then
    X := X - Y
else
    Y := Y - X;
write(X);
```

A. Computes the LCM of two numbers
B. Divides the larger number by the smaller number
C. Computes the GCD of two numbers
D. None of the above

gate1995 algorithms identify-function normal

1.10.8 Identify Function: GATE1995-4

https://gateoverflow.in/2640

a. Consider the following Pascal function where A and B are non-zero positive integers. What is the value of $GET(3,2)$?

```
function GET(A,B:integer): integer;
begin
    if B=0 then
        GET:= 1
    else if A < B then
```

© Copyright GATE Overflow. All rights reserved.

```
        GET:= 0
    else
        GET:= GET(A-1, B) + GET(A-1, B-1)
end;
```

b. The Pascal procedure given for computing the transpose of an $N \times N, (N > 1)$ matrix A of integers has an error. Find the error and correct it. Assume that the following declaration are made in the main program

```
const
    MAXSIZE=20;
type
    INTARR=array [1..MAXSIZE,1..MAXSIZE] of integer;
Procedure TRANSPOSE (var A: INTARR; N : integer);
var
    I, J, TMP: integer;
begin
    for I:=1 to N - 1 do
    for J:=1 to N do
    begin
        TMP:= A[I, J];
        A[I, J]:= A[J, I];
        A[J, I]:= TMP
    end
end;
```

gate1995 algorithms identify-function normal

1.10.9 Identify Function: GATE1998-2.12

https://gateoverflow.in/1684

What value would the following function return for the input $x = 95$?

```
Function fun (x:integer):integer;
Begin
    If x > 100 then fun = x - 10
    Else fun = fun(fun (x+11))
End;
```

A. 89 B. 90 C. 91 D. 92

gate1998 algorithms recursion identify-function normal

1.10.10 Identify Function: GATE1999-2.24

https://gateoverflow.in/1501

Consider the following C function definition

```
int Trial (int a, int b, int c)
{
    if ((a>=b) && (c<b)) return b;
    else if (a>=b) return Trial(a, c, b);
    else return Trial(b, a, c);
}
```

The functional Trial:

A. Finds the maximum of a, b, and c
B. Finds the minimum of a, b, and c
C. Finds the middle number of a, b, c
D. None of the above

gate1999 algorithms identify-function normal

1.10.11 Identify Function: GATE2000-2.15

https://gateoverflow.in/662

Suppose you are given an array $s[1...n]$ and a procedure reverse (s, i, j) which reverses the order of elements in s between positions i and j (both inclusive). What does the following sequence do, where $1 \leqslant k \leqslant n$:

```
        reverse (s, 1, k);
        reverse (s, k+1, n);
        reverse (s, 1, n);
```

A. Rotates s left by k positions
B. Leaves s unchanged
C. Reverses all elements of s
D. None of the above

gate2000 algorithms normal identify-function

© Copyright GATE Overflow. All rights reserved.

1.10.12 Identify Function: GATE2003-1

https://gateoverflow.in/892

Consider the following C function.

For large values of y, the return value of the function f best approximates

```
float f,(float x, int y) {
    float p, s; int i;
    for (s=1,p=1,i=1; i<y; i++) {
        p *= x/i;
        s += p;
    }
    return s;
}
```

A. x^y B. e^x C. $\ln(1+x)$ D. x^x

gate2003 algorithms identify-function normal

1.10.13 Identify Function: GATE2003-88

https://gateoverflow.in/971

In the following C program fragment, j, k, n and TwoLog_n are integer variables, and A is an array of integers. The variable n is initialized to an integer ≥ 3, and TwoLog_n is initialized to the value of $2^{*}\lceil \log_2(n) \rceil$

```
for (k = 3;  k <= n; k++)
        A[k] = 0;
for (k = 2; k <= TwoLog_n; k++)
    for (j = k+1; j <= n; j++)
        A[j] = A[j] || (j%k);
for (j = 3; j <= n; j++)
    if (!A[j]) printf("%d", j);
```

The set of numbers printed by this program fragment is

A. $\{m \mid m \leq n, (\exists i)\, [m = i!]\}$

B. $\{m \mid m \leq n, (\exists i)\, [m = i^2]\}$

C. $\{m \mid m \leq n, \text{m is prime}\}$

D. $\{\,\}$

gate2003 algorithms identify-function normal

1.10.14 Identify Function: GATE2004-41

https://gateoverflow.in/1038

Consider the following C program

```
main()
{
    int x, y, m, n;
    scanf("%d %d", &x, &y);
    /* Assume x>0 and y>0*/
    m = x; n = y;
    while(m != n)
        {
            if (m > n)
                m = m-n;
            else
                n = n-m;
        }
    printf("%d", n);
}
```

The program computes

A. $x+y$ using repeated subtraction

B. $x \mod y$ using repeated subtraction

C. the greatest common divisor of x and y

D. the least common multiple of x and y

gate2004 algorithms normal identify-function

1.10.15 Identify Function: GATE2004-42

https://gateoverflow.in/1039

What does the following algorithm approximate? (Assume $m > 1, \epsilon > 0$).

```
x = m;
y = 1;
While (x-y > ϵ)
{
     x = (x+y)/2;
     y = m/x;
```

© Copyright GATE Overflow. All rights reserved.

```
}
print(x);
```

A. $\log m$ B. m^2 C. $m^{\frac{1}{2}}$ D. $m^{\frac{1}{3}}$

gate2004 algorithms identify-function normal

1.10.16 Identify Function: GATE2005-31

https://gateoverflow.in/1367

Consider the following C-program:

```
void foo (int n, int sum) {
    int k = 0, j = 0;
    if (n == 0) return;
    k = n % 10; j = n/10;
    sum = sum + k;
    foo (j, sum);
    printf ("%d,",k);
}

int main() {
    int a = 2048, sum = 0;
    foo(a, sum);
    printf("%d\n", sum);
}
```

What does the above program print?

A. $8, 4, 0, 2, 14$ B. $8, 4, 0, 2, 0$
C. $2, 0, 4, 8, 14$ D. $2, 0, 4, 8, 0$

gate2005 algorithms identify-function recursion normal

1.10.17 Identify Function: GATE2005-IT-57

https://gateoverflow.in/3818

What is the output printed by the following program?

```
#include <stdio.h>

int f(int n, int k) {
    if (n == 0) return 0;
    else if (n % 2) return f(n/2, 2*k) + k;
    else return f(n/2, 2*k) - k;
}

int main () {
    printf("%d", f(20, 1));
    return 0;
}
```

A. 5 B. 8 C. 9 D. 20

gate2005-it algorithms identify-function normal

1.10.18 Identify Function: GATE2006-50

https://gateoverflow.in/1828

A set X can be represented by an array $x[n]$ as follows:

$$x[i] = \begin{cases} 1 & \text{if } i \in X \\ 0 & \text{otherwise} \end{cases}$$

Consider the following algorithm in which x, y, and z are Boolean arrays of size n:

```
algorithm zzz(x[], y[], z[]) {
    int i;

    for(i=0; i<n; ++i)
        z[i] = (x[i] ∧ ~y[i]) ∨ (~x[i] ∧ y[i]);
}
```

The set Z computed by the algorithm is:

A. $(X \cup Y)$ B. $(X \cap Y)$ C. $(X - Y) \cap (Y - X)$ D. $(X - Y) \cup (Y - X)$

gate2006 algorithms identify-function normal

© Copyright GATE Overflow. All rights reserved.

1.10.19 Identify Function: GATE2006-53

https://gateoverflow.in/1831

Consider the following C-function in which $a[n]$ and $b[m]$ are two sorted integer arrays and $c[n+m]$ be another integer array,

```
void xyz(int a[], int b [], int c []){
    int i,j,k;
    i=j=k=0;
    while ((i<n) && (j<m))
        if (a[i] < b[j]) c[k++] = a[i++];
        else c[k++] = b[j++];
}
```

Which of the following condition(s) hold(s) after the termination of the while loop?

i. $j<m, k=n+j-1$ and $a[n-1]<b[j]$ if $i=n$
ii. $i<n, k=m+i-1$ and $b[m-1]\leq a[i]$ if $j=m$

A. only (i)
B. only (ii)
C. either (i) or (ii) but not both
D. neither (i) nor (ii)

gate2006 algorithms identify-function normal

1.10.20 Identify Function: GATE2006-IT-52

https://gateoverflow.in/3595

The following function computes the value of $\binom{m}{n}$ correctly for all legal values m and n ($m \geq 1, n \geq 0$ and $m > n$)

```
int func(int m, int n)
{
    if (E) return 1;
    else return(func(m -1, n) + func(m - 1, n - 1));
}
```

In the above function, which of the following is the correct expression for E?

A. $(n == 0) || (m == 1)$
B. $(n == 0)$ && $(m == 1)$
C. $(n == 0) || (m == n)$
D. $(n == 0)$ && $(m == n)$

gate2006-it algorithms identify-function normal

1.10.21 Identify Function: GATE2008-IT-82

https://gateoverflow.in/3406

Consider the code fragment written in C below :

```
void f (int n)
{
  if (n <=1)  {
   printf ("%d", n);
  }
  else {
   f (n/2);
   printf ("%d", n%2);
  }
}
```

What does f(173) print?

A. 010110101
B. 010101101
C. 10110101
D. 10101101

gate2008-it algorithms recursion identify-function normal

1.10.22 Identify Function: GATE2008-IT-83

https://gateoverflow.in/3407

Consider the code fragment written in C below :

```
void f (int n)
{
    if (n <= 1)  {
        printf ("%d", n);
    }
    else {
        f (n/2);
        printf ("%d", n%2);
    }
}
```

© Copyright GATE Overflow. All rights reserved.

Which of the following implementations will produce the same output for $f(173)$ as the above code?

P1

```
void f (int n)
{
    if (n/2)  {
        f(n/2);
    }
    printf ("%d", n%2);
}
```

P2

```
void f (int n)
{
    if (n <=1)  {
        printf ("%d", n);
    }
    else {
        printf ("%d", n%2);
        f (n/2);
    }
}
```

A. Both $P1$ and $P2$ B. $P2$ only C. $P1$ only D. Neither $P1$ nor $P2$

gate2008-it algorithms recursion identify-function normal

1.10.23 Identify Function: GATE2009-18

https://gateoverflow.in/1310

Consider the program below:

```
#include <stdio.h>
int fun(int n, int *f_p) {
    int t, f;
    if (n <= 1) {
        *f_p = 1;
        return 1;
    }
    t = fun(n-1, f_p);
    f = t + *f_p;
    *f_p = t;
    return f;
}

int main() {
    int x = 15;
    printf("%d/n", fun(5, &x));
    return 0;
}
```

The value printed is:

A. 6 B. 8 C. 14 D. 15

gate2009 algorithms recursion identify-function normal

1.10.24 Identify Function: GATE2010-35

https://gateoverflow.in/2336

What is the value printed by the following C program?

```
#include<stdio.h>

int f(int *a, int n)
{
    if (n <= 0) return 0;
    else if (*a % 2 == 0) return *a+f(a+1, n-1);
    else return *a - f(a+1, n-1);
}

int main()
{
    int a[] = (12, 7, 13, 4, 11, 6);
    printf("%d", f(a, 6));
    return 0;
}
```

A. -9 B. 5 C. 15 D. 19

gate2010 algorithms recursion identify-function normal

1.10.25 Identify Function: GATE2011-48

https://gateoverflow.in/2154

Consider the following recursive C function that takes two arguments.

```
unsigned int foo(unsigned int n, unsigned int r) {
```

© Copyright GATE Overflow. All rights reserved.

```
    if (n>0) return ((n%r) + foo(n/r, r));
    else return 0;
}
```

What is the return value of the function foo when it is called as $\text{foo}(345, 10)$?

A. 345 B. 12 C. 5 D. 3

gate2011 algorithms recursion identify-function normal

1.10.26 Identify Function: GATE2011-49

https://gateoverflow.in/43324

Consider the following recursive C function that takes two arguments.

```
unsigned int foo(unsigned int n, unsigned int r) {
    if (n>0) return ((n%r) + foo(n/r, r));
    else return 0;
}
```

What is the return value of the function foo when it is called as $\text{foo}(513, 2)$?

A. 9 B. 8 C. 5 D. 2

gate2011 algorithms recursion identify-function normal

1.10.27 Identify Function: GATE2013-31

https://gateoverflow.in/1542

Consider the following function:

```
int unknown(int n){

int i, j, k=0;
for (i=n/2; i<=n; i++)
    for (j=2; j<=n; j=j*2)
        k = k + n/2;
 return (k);

}
```

The return value of the function is

A. $\Theta(n^2)$
B. $\Theta(n^2 \log n)$
C. $\Theta(n^3)$
D. $\Theta(n^3 \log n)$

gate2013 algorithms identify-function normal

1.10.28 Identify Function: GATE2014-1-41

https://gateoverflow.in/1919

Consider the following C function in which **size** is the number of elements in the array **E**:

```
int MyX(int *E, unsigned int size)
{
   int Y = 0;
   int Z;
   int i, j, k;

   for(i = 0; i< size; i++)
          Y = Y + E[i];

    for(i=0; i < size; i++)
        for(j = i; j < size; j++)
        {
            Z = 0;
            for(k = i; k <= j; k++)
               Z = Z + E[k];
            if(Z > Y)
               Y = Z;
        }
   return Y;
}
```

The value returned by the function **MyX** is the

A. maximum possible sum of elements in any sub-array of array **E**.
B. maximum element in any sub-array of array **E.**
C. sum of the maximum elements in all possible sub-arrays of array **E**.
D. the sum of all the elements in the array **E.**

© Copyright GATE Overflow. All rights reserved.

gate2014-1 algorithms identify-function normal

1.10.29 Identify Function: GATE2014-2-10

https://gateoverflow.in/1964

Consider the function func shown below:

```
int func(int num) {
   int count = 0;
   while (num) {
     count++;
     num>>= 1;
   }
   return (count);
}
```

The value returned by func(435) is _____

gate2014-2 algorithms identify-function numerical-answers easy

1.10.30 Identify Function: GATE2014-3-10

https://gateoverflow.in/2044

Let A be the square matrix of size $n \times n$. Consider the following pseudocode. What is the expected output?

```
C=100;
for i=1 to n do
    for j=1 to n do
    {
        Temp = A[i][j]+C;
        A[i][j] = A[j][i];
        A[j][i] = Temp -C;
    }
for i=1 to n do
    for j=1 to n do
        output (A[i][j]);
```

A. The matrix A itself
B. Transpose of the matrix A
C. Adding 100 to the upper diagonal elements and subtracting 100 from lower diagonal elements of A
D. None of the above

gate2014-3 algorithms identify-function easy

1.10.31 Identify Function: GATE2015-1-31

https://gateoverflow.in/8263

Consider the following C function.

```
int fun1 (int n) {
     int i, j, k, p, q = 0;
     for (i = 1; i < n; ++i)
     {
        p = 0;
       for (j = n; j > 1; j = j/2)
           ++p;
       for (k = 1; k < p; k = k * 2)
           ++q;
     }
     return q;
}
```

Which one of the following most closely approximates the return value of the function fun1?

A. n^3 B. $n(\log n)^2$ C. $n \log n$ D. $n \log(\log n)$

gate2015-1 algorithms normal identify-function

1.10.32 Identify Function: GATE2015-2-11

https://gateoverflow.in/8060

Consider the following C function.

```
int fun(int n) {
    int x=1, k;
    if (n==1) return x;
    for (k=1; k<n; ++k)
        x = x + fun(k) * fun (n-k);
    return x;
```

© Copyright GATE Overflow. All rights reserved.

```
}
```

The return value of $fun(5)$ is _____.

gate2015-2 algorithms identify-function recurrence normal numerical-answers

1.10.33 Identify Function: GATE2015-3-49

https://gateoverflow.in/8558

Suppose $c = \langle c[0], \ldots, c[k-1] \rangle$ is an array of length k, where all the entries are from the set $\{0,1\}$. For any positive integers a and n, consider the following pseudocode.

DOSOMETHING (*c, a, n)*

$z \leftarrow 1$
for $i \leftarrow 0$ to $k-1$
  do $z \leftarrow z^2 \bmod n$
  if *c*[i]=1
    then $z \leftarrow (z \times a) \bmod n$
return z

If $k = 4, c = \langle 1,0,1,1 \rangle, a = 2,$ and $n = 8$, then the output of DOSOMETHING(*c, a, n)* is _____.

gate2015-3 algorithms identify-function normal numerical-answers

1.10.34 Identify Function: GATE2019-26

https://gateoverflow.in/302822

Consider the following C function.

```
void convert (int n ) {
        if (n<0)
            printf{"%d", n);
        else {
            convert(n/2);
            printf("%d", n%2);
        }
}
```

Which one of the following will happen when the function *convert* is called with any positive integer n as argument?

A. It will print the binary representation of n and terminate
B. It will print the binary representation of n in the reverse order and terminate
C. It will print the binary representation of n but will not terminate
D. It will not print anything and will not terminate

gate2019 algorithms identify-function

1.10.35 Identify Function: TIFR2010-B-24

https://gateoverflow.in/18742

Consider the following program operating on four variables u, v, x, y, and two constants X and Y.

```
x, y, u, v:= X, Y, Y, X;
While (x ≠ y)
do
    if (x > y) then x, v := x - y, v + u;
    else if (y > x) then y, u:= y - x, u + v;
od;
print ((x + y) / 2);  print ((u + v) / 2);
```

Given $X > 0 \wedge Y > 0$, pick the true statement out of the following:

A. The program prints $\gcd(X,Y)$ and the first prime larger than both X and Y.
B. The program prints $\gcd(X,Y)$ followed by $\text{lcm}(X,Y)$.
C. The program prints $\gcd(X,Y)$ followed by $\frac{1}{2} \times \text{lcm}(X,Y)$.
D. The program prints $\frac{1}{2} \times \gcd(X,Y)$ followed by $\frac{1}{2} \times \text{lcm}(X,Y)$.
E. The program does none of the above.

tifr2010 algorithms identify-function

© Copyright GATE Overflow. All rights reserved.

1.10.36 Identify Function: TIFR2014-B-2

https://gateoverflow.in/27136

Consider the following code.

```
def brian(n):
    count = 0

    while ( n ! = 0 )
          n = n & ( n-1 )
         count = count + 1

  return count
```

Here n is meant to be an unsigned integer. The operator & considers its arguments in binary and computes their bit wise AND. For example, 22 & 15 gives 6, because the binary (say 8-bit) representation of 22 is 00010110 and the binary representation of 15 is 00001111, and the bit-wise AND of these binary strings is 00000110, which is the binary representation of 6. What does the function brian return?

a. The highest power of 2 dividing n, but zero if n is zero.
b. The number obtained by complementing the binary representation of n.
c. The number of ones in the binary representation of n.
d. The code might go into an infinite loop for some n.
e. The result depends on the number of bits used to store unsigned integers.

tifr2014 algorithms identify-function

1.10.37 Identify Function: TIFR2014-B-20

https://gateoverflow.in/27354

Consider the following game. There is a list of distinct numbers. At any round, a player arbitrarily chooses two numbers a, b from the list and generates a new number c by subtracting the smaller number from the larger one. The numbers a and b are put back in the list. If the number c is non-zero and is not yet in the list, c is added to the list. The player is allowed to play as many rounds as the player wants. The score of a player at the end is the size of the final list.

Suppose at the beginning of the game the list contains the following numbers: $48, 99, 120, 165$ and 273. What is the score of the best player for this game?

A. 40 B. 16 C. 33 D. 91 E. 123

tifr2014 algorithms identify-function

1.10.38 Identify Function: TIFR2017-A-12

https://gateoverflow.in/95299

Consider the following program modifying an $n \times n$ square matrix A:

```
for i=1 to n:
    for j=1 to n:
        temp=A[i][j]+10
        A[i][j]=A[j][i]
        A[j][i]=temp-10
    end for
end for
```

Which of the following statements about the contents of matrix A at the end of this program must be TRUE?

A. the new A is the transpose of the old A
B. all elements above the diagonal have their values increased by 10 and all the values below have their values decreased by 10
C. all elements above the diagonal have their values decreased by 10 and all the values below have their values increased by 10
D. the new matrix A is symmetric, that is, $A[i][j] = A[j][i]$ for all $1 \leq i, j \leq n$
E. A remains unchanged

tifr2017 algorithms identify-function

1.11 Minimum Maximum (4)

1.11.1 Minimum Maximum: GATE2014-1-39

https://gateoverflow.in/1917

The minimum number of comparisons required to find the minimum and the maximum of 100 numbers is _____

© Copyright GATE Overflow. All rights reserved.

gate2014-1 algorithms numerical-answers normal minimum-maximum

1.11.2 Minimum Maximum: TIFR2014-B-10

https://gateoverflow.in/27198

Given a set of n distinct numbers, we would like to determine both the smallest and the largest number. Which of the following statements is TRUE?

A. These two elements can be determined using $O\left(\log^{100} n\right)$ comparisons.
B. $O\left(\log^{100} n\right)$ comparisons do not suffice, however these two elements can be determined using $n + O(\log n)$ comparisons.
C. $n + O(\log n)$ comparisons do not suffice, however these two elements can be determined using $3\lceil n/2 \rceil$ comparisons.
D. $3\lceil n/2 \rceil$ comparisons do not suffice, however these two elements can be determined using $2(n-1)$ comparisons.
E. None of the above.

tifr2014 algorithms minimum-maximum

1.11.3 Minimum Maximum: TIFR2014-B-6

https://gateoverflow.in/27183

Consider the problem of computing the minimum of a set of n distinct numbers. We choose a permutation uniformly at random (i.e., each of the n! permutations of $\langle 1, \ldots, n \rangle$ is chosen with probability $(1/n!)$ and we inspect the numbers in the order given by this permutation. We maintain a variable MIN that holds the minimum value seen so far. MIN is initialized to ∞ and if we see a value smaller than MIN during our inspection, then MIN is updated. For example, in the inspection given by the following sequence, MIN is updated four times.

5 9 4 2 6 8 0 3 1 7

What is the expected number of times MIN is updated?

A. $O(1)$ B. $H_n = \sum_{i=1}^{n} 1/i$ C. $\sqrt{n}$ D. $n/2$ E. n

tifr2014 algorithms minimum-maximum

1.11.4 Minimum Maximum: TIFR2014-B-9

https://gateoverflow.in/27194

Given a set of n distinct numbers, we would like to determine the smallest three numbers in this set using comparisons. Which of the following statements is TRUE?

A. These three elements can be determined using $O\left(\log^2 n\right)$ comparisons.
B. $O\left(\log^2 n\right)$ comparisons do not suffice, however these three elements can be determined using $n + O(1)$ comparisons.
C. $n + O(1)$ comparisons do not suffice, however these three elements can be determined using $n + O(\log n)$ comparisons.
D. $n + O(\log n)$ comparisons do not suffice, however these three elements can be determined using $O(n)$ comparisons.
E. None of the above.

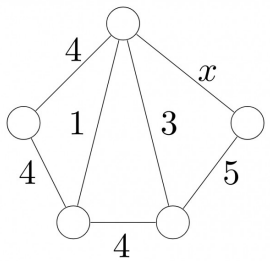
tifr2014 algorithms minimum-maximum

1.12 Minimum Spanning Trees (3)

1.12.1 Minimum Spanning Trees: GATE2018-47

https://gateoverflow.in/204122

Consider the following undirected graph G:



Choose a value for x that will maximize the number of minimum weight spanning trees (MWSTs) of G. The number of MWSTs of G for this value of x is ____.

gate2018 algorithms graph-algorithms minimum-spanning-trees numerical-answers

© Copyright GATE Overflow. All rights reserved.

1.12.2 Minimum Spanning Trees: TIFR2018-B-13

https://gateoverflow.in/179297

Let $n \geq 3$, and let G be a simple, connected, undirected graph with the same number n of vertices and edges. Each edge of G has a distinct real weight associated with it. Let T be the minimum weight spanning tree of G. Which of the following statements is NOT ALWAYS TRUE ?

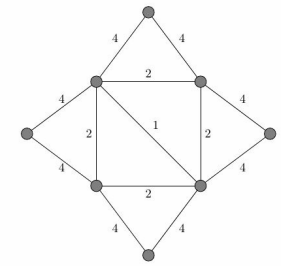
A. The minimum weight edge of G is in T.
B. The maximum weight edge of G is not in T.
C. G has a unique cycle C and the minimum weight edge of C is also in T.
D. G has a unique cycle C and the maximum weight edge of C is not in T.
E. T can be found in $O(n)$ time from the adjacency list representation of G.

tifr2018 graph-algorithms minimum-spanning-trees

1.12.3 Minimum Spanning Trees: TIFR2019-B-2

https://gateoverflow.in/280493

How many distinct minimum weight spanning trees does the following undirected, weighted graph have ?



A. 8 B. 16 C. 32 D. 64 E. None of the above

tifr2019 algorithms minimum-spanning-trees

1.13 P Np Npc Nph (12)

1.13.1 P Np Npc Nph: GATE1992-02,vi

https://gateoverflow.in/561

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

Which of the following problems is not NP-hard?

a. Hamiltonian circuit problem
b. The $0/1$ Knapsack problem
c. Finding bi-connected components of a graph
d. The graph coloring problem

gate1992 p-np-npc-nph algorithms

1.13.2 P Np Npc Nph: GATE2003-12

https://gateoverflow.in/903

Ram and Shyam have been asked to show that a certain problem Π is NP-complete. Ram shows a polynomial time reduction from the 3-SAT problem to Π, and Shyam shows a polynomial time reduction from Π to 3-SAT. Which of the following can be inferred from these reductions?

A. Π is NP-hard but not NP-complete
B. Π is in NP, but is not NP-complete
C. Π is NP-complete
D. Π is neither NP-hard, nor in NP

gate2003 algorithms p-np-npc-nph normal

1.13.3 P Np Npc Nph: GATE2004-30, ISRO2017-10

https://gateoverflow.in/1027

The problem 3-SAT and 2-SAT are

A. both in P
B. both NP complete
C. NP-complete and in P respectively
D. undecidable and NP complete respectively

gate2004 algorithms p-np-npc-nph easy isro2017

1.13.4 P Np Npc Nph: GATE2006-16, ISRO-DEC2017-27

https://gateoverflow.in/977

Let S be an NP-complete problem and Q and R be two other problems not known to be in NP. Q is polynomial time reducible to S and S is polynomial-time reducible to R. Which one of the following statements is true?

© Copyright GATE Overflow. All rights reserved.

A. R is NP-complete
B. R is NP-hard
C. Q is NP-complete
D. Q is NP-hard

gate2006 algorithms p-np-npc-nph normal isrodec2017

1.13.5 P Np Npc Nph: GATE2008-44

https://gateoverflow.in/456

The subset-sum problem is defined as follows: Given a set S of n positive integers and a positive integer W, determine whether there is a subset of S whose elements sum to W. An algorithm Q solves this problem in $O(nW)$ time. Which of the following statements is false?

A. Q solves the subset-sum problem in polynomial time when the input is encoded in unary
B. Q solves the subset-sum problem in polynomial time when the input is encoded in binary
C. The subset sum problem belongs to the class NP
D. The subset sum problem is NP-hard

gate2008 algorithms p-np-npc-nph normal

1.13.6 P Np Npc Nph: TIFR2010-B-39

https://gateoverflow.in/18754

Suppose a language L is $\mathbf{NP}$ complete. Then which of the following is FALSE?

A. $L \in \mathbf{NP}$
B. Every problem in $\mathbf{P}$ is polynomial time reducible to L.
C. Every problem in $\mathbf{NP}$ is polynomial time reducible to L.
D. The Hamilton cycle problem is polynomial time reducible to L.
E. $\mathbf{P} \neq \mathbf{NP}$ and $L \in \mathbf{P}$.

tifr2010 algorithms p-np-npc-nph

1.13.7 P Np Npc Nph: TIFR2011-B-37

https://gateoverflow.in/20922

Given an integer $n \geq 3$, consider the problem of determining if there exist integers $a, b \geq 2$ such that $n = a^b$. Call this the forward problem. The reverse problem is: given a and b, compute a^b (mod b). Note that the input length for the forward problem is $\lfloor \log n \rfloor + 1$, while the input length for the reverse problem is $\lfloor \log a \rfloor + \lfloor \log b \rfloor + 2$. Which of the following statements is TRUE?

a. Both the forward and reverse problems can be solved in time polynomial in the lengths of their respective inputs.
b. The forward problem can be solved in polynomial time, however the reverse problem is NP-hard.
c. The reverse problem can be solved in polynomial time, however the forward problem is NP-hard.
d. Both the forward and reverse problem are NP-hard.
e. None of the above.

tifr2011 algorithms p-np-npc-nph

1.13.8 P Np Npc Nph: TIFR2012-B-20

https://gateoverflow.in/26480

This question concerns the classes P and NP. If you are familiar with them, you may skip the definitions and go directly to the question.

Let L be a set. We say that L is in P if there is some algorithm which given input x decides if x is in L or not in time bounded by a polynomial in the length of x. For example, the set of all connected graphs is in P, because there is an algorithm which, given a graph graph, can decide if it is connected or not in time roughly proportional to the number of edges of the graph.

The class NP is a superset of class P. It contains those sets that have membership witnesses that can be verified in polynomial time. For example, the set of composite numbers is in NP. To see this take the witness for a composite number to be one of its divisors. Then the verification process consists of performing just one division using two reasonable size numbers. Similarly, the set of those graphs that have a Hamilton cycle, i.e. a cycle containing all the vertices of the graph, is in in NP. To verify that the graph has a Hamilton cycle we just check if the witnessing sequence of vertices indeed a cycle of the graph that passes through all the vertices of the graph. This can be done in time that is polynomial in the size of the graph.

More precisely, if L is a set in P consisting of elements of the form (x, w), then the set

$M = \{x :$ $\exists$

is in N P .

Let $G = (V, E)$ be a graph. G is said to have perfect matching if there is a subset M of the edges of G so that

© Copyright GATE Overflow. All rights reserved.

i. No two edges in M intersect (have a vertex in common); and
ii. Every vertex of G has an edge in M.

Let MATCH be the set of all graphs that have a perfect matching. Let $\overline{\text{MATCH}}$ be the set of graphs that do not have a perfect matching. Let $o(G)$ be the number of components of G that have an odd number of vertices.

Tutte's Theorem: $G \in \text{MATCH}$ if and only if for all subsets S of V, the number of components in $G - S$ (the graph formed by deleting the vertices in S) with an odd number of vertices is at most $|S|$. That is,

$G \in \text{MATCH} \leftrightarrow \qquad \forall$

Which of the following is true?

A. $\text{MATCH} \in NP$ and $\overline{\text{MATCH}} \notin NP$
B. $\overline{\text{MATCH}} \in NP$ and $\text{MATCH} \notin NP$
C. $\text{MATCH} \in NP$ and $\overline{\text{MATCH}} \in NP$
D. $\text{MATCH} \notin P$ and $\overline{\text{MATCH}} \notin P$
E. none of the above

tifr2012 algorithms p-np-npc-nph

1.13.9 P Np Npc Nph: TIFR2013-B-7

https://gateoverflow.in/25668

Which of the following is not implied by $P = NP$?

a. 3SAT can be solved in polynomial time.
b. Halting problem can be solved in polynomial time.
c. Factoring can be solved in polynomial time.
d. Graph isomorphism can be solved in polynomial time.
e. Travelling salesman problem can be solved in polynomial time.

tifr2013 algorithms p-np-npc-nph

1.13.10 P Np Npc Nph: TIFR2017-B-15

https://gateoverflow.in/95828

A multivariate polynomial in n variables with integer coefficients has a binary root if it is possible to assign each variable either 0 or 1, so that the polynomial evaluates to 0. For example, the multivariate polynomial $-2x_1^3 - x_1x_2 + 2$ has the binary root $(x_1 = 1, x_2 = 0)$. Then determining whether a multivariate polynomial, given as the sum of monimials, has a binary root:

A. is trivial: every polynomial has a binary root
B. can be done in polynomial time
C. is NP-hard, but not in NP
D. is in NP, but not in P and not NP-hard
E. is both in NP and NP-hard

tifr2017 algorithms p-np-npc-nph

1.13.11 P Np Npc Nph: TIFR2017-B-2

https://gateoverflow.in/95673

Consider the following statements:

i. Checking if a given *undirected* graph has a cycle is in P
ii. Checking if a given *undirected* graph has a cycle is in NP
iii. Checking if a given *directed* graph has a cycle is in P
iv. Checking if a given *directed* graph has a cycle is in NP

Which of the above statements is/are TRUE? Choose from the following options.

A. Only i and ii B. Only ii and iv C. Only ii, iii, and iv D. Only i, ii and iv E. All of them

tifr2017 algorithms p-np-npc-nph

1.13.12 P Np Npc Nph: TIFR2019-B-7

https://gateoverflow.in/280488

A formula is said to be a 3-CF-formula if it is a conjunction (i.e., an AND) of clauses, and each clause has at most 3 literals. Analogously, a formula is said to be a 3-DF-formula if it is a disjunction (i.e., an OR) of clauses of at most 3 literals each.

Define the languages 3-CF-SAT and 3-DF-SAT as follows:

$$\text{3-CF-SAT} = \{\Phi \mid \Phi \text{ is a } \textit{satisfiable} \text{ 3-CF formula}\}$$

© Copyright GATE Overflow. All rights reserved.

$$\text{3-DF-SAT} = \{\Phi \mid \Phi \text{ is a } \textit{satisfiable} \text{3-DF formula}\}$$

Which of the following best represents our current knowledge of these languages ?

A. Both 3-CF-SAT and 3-DF-SAT are in NP but only 3-CF-SAT is NP-complete
B. Both 3-CF-SAT and 3-DF-SAT are in NP-complete
C. Both 3-CF-SAT and 3-DF-SAT are in P
D. Both 3-CF-SAT and 3-DF-SAT are in NP but only 3-DF-SAT is NP-complete
E. Neither 3-CF-SAT nor 3-DF-SAT are in P

tifr2019 algorithms p-np-npc-nph

1.14 Quicksort (2)

1.14.1 Quicksort: GATE2019-20 https://gateoverflow.in/302828

An array of 25 distinct elements is to be sorted using quicksort. Assume that the pivot element is chosen uniformly at random. The probability that the pivot element gets placed in the worst possible location in the first round of partitioning (rounded off to 2 decimal places) is _____

gate2019 numerical-answers algorithms quicksort probability

1.14.2 Quicksort: TIFR2018-B-7 https://gateoverflow.in/179291

Consider the recursive quicksort algorithm with "random pivoting". That is, in each recursive call, a pivot is chosen uniformly at random from the sub-array being sorted.When this randomized algorithm is applied to an array of size n all whose elements are distinct, what is the probability that the smallest and the largest elements in the array are compared during a run of the algorithm ?

A. $\left(\frac{1}{n}\right)$ B. $\left(\frac{2}{n}\right)$ C. $\Theta\left(\frac{1}{n \log n}\right)$ D. $O\left(\frac{1}{n^2}\right)$ E. $\Theta\left(\frac{1}{n \log^2 n}\right)$

tifr2018 algorithms sorting quicksort

1.15 Recurrence (37)

1.15.1 Recurrence: GATE1987-10a https://gateoverflow.in/82450

Solve the recurrence equations:

- $T(n) = T(n-1) + n$
- $T(1) = 1$

gate1987 algorithms recurrence

1.15.2 Recurrence: GATE1988-13iv https://gateoverflow.in/94637

Solve the recurrence equations:

- $T(n) = T(\frac{n}{2}) + 1$
- $T(1) = 1$

gate1988 descriptive algorithms recurrence

1.15.3 Recurrence: GATE1989-13b https://gateoverflow.in/93175

Find a solution to the following recurrence equation:

$T(n) = \sqrt{n} + T(\frac{n}{2})$

$T(1) = 1$

gate1989 descriptive algorithms recurrence

© Copyright GATE Overflow. All rights reserved.

1.15.4 Recurrence: GATE1990-17a

https://gateoverflow.in/86878

Express $T(n)$ in terms of the harmonic number $H_n = \sum_{t=1}^{n} 1/i, n \geq 1$, where $T(n)$ satisfies the recurrence relation,

$T(n) = \frac{n+1}{n} T(n-1) + 1$, for $n \geq \sum$ and $T(1) = 1$

What is the asymptotic behaviour of $T(n)$ as a function of n ?

gate1990 descriptive algorithms recurrence

1.15.5 Recurrence: GATE1992-07a

https://gateoverflow.in/586

Consider the function $F(n)$ for which the pseudocode is given below :

```
Function F(n)
begin
F1 ← 1
if(n=1) then F ← 3
 else
   For i = 1 to n do
        begin
            C ← 0
           For j = 1 to n - 1 do
           begin C ← C + 1 end
           F1 = F1 * C
       end
 F = F1
end
```

[n is a positive integer greater than zero]

(a) Derive a recurrence relation for $F(n)$

gate1992 algorithms recurrence descriptive

1.15.6 Recurrence: GATE1992-07b

https://gateoverflow.in/43600

Consider the function $F(n)$ for which the pseudocode is given below :

```
Function F(n)
begin
F1 ← 1
if(n=1) then F ← 3
 else
   For i = 1 to n do
        begin
            C ← 0
           For j = 1 to n - 1 do
           begin C ← C + 1 end
           F1 = F1 * C
       end
 F = F1
end
```

[n is a positive integer greater than zero]

Solve the recurrence relation for a closed form solution of $F(n)$.

gate1992 algorithms recurrence descriptive

1.15.7 Recurrence: GATE1993-15

https://gateoverflow.in/2312

Consider the recursive algorithm given below:

```
procedure bubblesort (n);
var i,j: index; temp : item;
begin
    for i:=1 to n-1 do
        if A[i] > A[i+1] then
        begin
            temp := A[i];
            A[i] := A[i+1];
            A[i+1] := temp;
        end;
    bubblesort (n-1)
end
```

© Copyright GATE Overflow. All rights reserved.

Let a_n be the number of times the 'if…then…' statement gets executed when the algorithm is run with value n. Set up the recurrence relation by defining a_n in terms of a_{n-1}. Solve for a_n.

gate1993 algorithms recurrence normal

1.15.8 Recurrence: GATE1994-1.7, ISRO2017-14

https://gateoverflow.in/2444

The recurrence relation that arises in relation with the complexity of binary search is:

A. $T(n) = 2T\left(\frac{n}{2}\right) + k$, k is a constant
B. $T(n) = T\left(\frac{n}{2}\right) + k$, k is a constant
C. $T(n) = T\left(\frac{n}{2}\right) + \log n$
D. $T(n) = T\left(\frac{n}{2}\right) + n$

gate1994 algorithms recurrence easy isro2017

1.15.9 Recurrence: GATE1996-2.12

https://gateoverflow.in/2741

The recurrence relation

- $T(1) = 2$
- $T(n) = 3T(\frac{n}{4}) + n$

has the solution $T(n)$ equal to

A. $O(n)$
B. $O(\log n)$
C. $O\left(n^{\frac{3}{4}}\right)$
D. None of the above

gate1996 algorithms recurrence normal

1.15.10 Recurrence: GATE1997-15

https://gateoverflow.in/2275

Consider the following function.

```
Function F(n, m:integer):integer;
begin
    If (n<=0 or (m<=0) then F:=1
    else
      F:F(n-1, m) + F(n, m-1);
    end;
```

Use the recurrence relation $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$ to answer the following questions. Assume that n, m are positive integers. Write only the answers without any explanation.

a. What is the value of $F(n,2)$?
b. What is the value of $F(n,m)$?
c. How many recursive calls are made to the function F, including the original call, when evaluating $F(n,m)$.

gate1997 algorithms recurrence normal

1.15.11 Recurrence: GATE1997-4.6

https://gateoverflow.in/2247

Let $T(n)$ be the function defined by $T(1) = 1$, $T(n) = 2T(\lfloor \frac{n}{2} \rfloor) + \sqrt{n}$ for $n \geq 2$.

Which of the following statements is true?

A. $T(n) = O\sqrt{n}$
B. $T(n) = O(n)$
C. $T(n) = O(\log n)$
D. None of the above

gate1997 algorithms recurrence normal

1.15.12 Recurrence: GATE1998-6a

https://gateoverflow.in/44584

Solve the following recurrence relation

$x_n = 2x_{n-1} - 1, n > 1$

$x_1 = 2$

© Copyright GATE Overflow. All rights reserved.

gate1998 algorithms recurrence descriptive

1.15.13 Recurrence: GATE1999-2.21

https://gateoverflow.in/1498

If $T_1 = O(1)$, give the correct matching for the following pairs:

(M) $T_n = T_{n-1} + n$	(U) $T_n = O(n)$
(N) $T_n = T_{n/2} + n$	(V) $T_n = O(n \log n)$
(O) $T_n = T_{n/2} + n \log n$	(W) $T_n = O(n^2)$
(P) $T_n = T_{n-1} + \log n$	(X) $T_n = O(\log^2 n)$

A. M-W, N-V, O-U, P-X
B. M-W, N-U, O-X, P-V
C. M-V, N-W, O-X, P-U
D. M-W, N-U, O-V, P-X

gate1999 algorithms recurrence asymptotic-notations normal

1.15.14 Recurrence: GATE2002-1.3

https://gateoverflow.in/807

The solution to the recurrence equation $T(2^k) = 3T(2^{k-1}) + 1, T(1) = 1$ is

A. 2^k
B. $\frac{(3^{k+1}-1)}{2}$
C. $3^{\log_2 k}$
D. $2^{\log_3 k}$

gate2002 algorithms recurrence normal

1.15.15 Recurrence: GATE2002-2.11

https://gateoverflow.in/841

The running time of the following algorithm

Procedure $A(n)$

If $n \leqslant 2$ return (1) else return $(A(\lceil \sqrt{n} \rceil))$;

is best described by

A. $O(n)$
B. $O(\log n)$
C. $O(\log \log n)$
D. $O(1)$

gate2002 algorithms recurrence normal

1.15.16 Recurrence: GATE2003-35

https://gateoverflow.in/925

Consider the following recurrence relation

$T(1) = 1$

$T(n+1) = T(n) + \lfloor \sqrt{n+1} \rfloor$ for all $n \geq 1$

The value of $T(m^2)$ for $m \geq 1$ is

A. $\frac{m}{6}(21m - 39) + 4$
B. $\frac{m}{6}(4m^2 - 3m + 5)$
C. $\frac{m}{2}(3m^{2.5} - 11m + 20) - 5$
D. $\frac{m}{6}(5m^3 - 34m^2 + 137m - 104) + \frac{5}{6}$

gate2003 algorithms time-complexity recurrence difficult

1.15.17 Recurrence: GATE2004-83, ISRO2015-40

https://gateoverflow.in/1077

The time complexity of the following C function is (assume $n > 0$)

```
int recursive (int n) {
    if(n == 1)
        return (1);
    else
        return (recursive (n-1) + recursive (n-1));
}
```

A. $O(n)$
B. $O(n \log n)$
C. $O(n^2)$
D. $O(2^n)$

gate2004 algorithms recurrence time-complexity normal isro2015

1.15.18 Recurrence: GATE2004-84

https://gateoverflow.in/1078

The recurrence equation

$T(1) = 1$

$T(n) = 2T(n-1) + n, n \geq 2$

© Copyright GATE Overflow. All rights reserved.

evaluates to

A. $2^{n+1}-n-2$ B. 2^n-n C. $2^{n+1}-2n-2$ D. 2^n+n

gate2004 algorithms recurrence normal

1.15.19 Recurrence: GATE2004-IT-57

https://gateoverflow.in/3700

Consider a list of recursive algorithms and a list of recurrence relations as shown below. Each recurrence relation corresponds to exactly one algorithm and is used to derive the time complexity of the algorithm.

	Recursive Algorithm		Recurrence Relation
P	Binary search	I.	$T(n)=T(n-k)+T(k)+cn$
Q.	Merge sort	II.	$T(n)=2T(n-1)+1$
R.	Quick sort	III.	$T(n)=2T(n/2)+cn$
S.	Tower of Hanoi	IV.	$T(n)=T(n/2)+1$

Which of the following is the correct match between the algorithms and their recurrence relations?

A. P-II, Q-III, R-IV, S-I
B. P-IV, Q-III, R-I, S-II
C. P-III, Q-II, R-IV, S-I
D. P-IV, Q-II, R-I, S-III

gate2004-it algorithms recurrence normal

1.15.20 Recurrence: GATE2005-37

https://gateoverflow.in/1373

Suppose $T(n)=2T(\frac{n}{2})+n$, $T(0)=T(1)=1$

Which one of the following is FALSE?

A. $T(n)=O(n^2)$
B. $T(n)=\Theta(n\log n)$
C. $T(n)=\Omega(n^2)$
D. $T(n)=O(n\log n)$

gate2005 algorithms asymptotic-notations recurrence normal

1.15.21 Recurrence: GATE2005-IT-51

https://gateoverflow.in/3812

Let $T(n)$ be a function defined by the recurrence
$T(n)=2T(n/2)+\sqrt{n}$ for $n\geq 2$ and
$T(1)=1$

Which of the following statements is **TRUE**?

A. $T(n)=\Theta(\log n)$
B. $T(n)=\Theta(\sqrt{n})$
C. $T(n)=\Theta(n)$
D. $T(n)=\Theta(n\log n)$

gate2005-it algorithms recurrence easy

1.15.22 Recurrence: GATE2006-51, ISRO2016-34

https://gateoverflow.in/1829

Consider the following recurrence:

$$T(n)=2T\left(\sqrt{n}\right)+1,\ T(1)=1$$

Which one of the following is true?

A. $T(n)=\Theta(\log\log n)$
B. $T(n)=\Theta(\log n)$
C. $T(n)=\Theta(\sqrt{n})$
D. $T(n)=\Theta(n)$

algorithms recurrence isro2016 gate2006

1.15.23 Recurrence: GATE2008-78

https://gateoverflow.in/497

Let x_n denote the number of binary strings of length n that contain no consecutive 0s.

Which of the following recurrences does x_n satisfy?

A. $x_n=2x_{n-1}$
B. $x_n=x_{\lfloor n/2\rfloor}+1$
C. $x_n=x_{\lfloor n/2\rfloor}+n$
D. $x_n=x_{n-1}+x_{n-2}$

© Copyright GATE Overflow. All rights reserved.

gate2008 algorithms recurrence normal

1.15.24 Recurrence: GATE2008-79

https://gateoverflow.in/43485

Let x_n denote the number of binary strings of length n that contain no consecutive 0s.
The value of x_5 is

A. 5 B. 7 C. 8 D. 16

gate2008 algorithms recurrence normal

1.15.25 Recurrence: GATE2008-IT-44

https://gateoverflow.in/3354

When $n = 2^{2k}$ for some $k \geqslant 0$, the recurrence relation
$T(n) = \sqrt{(2)}T(n/2) + \sqrt{n}$, $T(1) = 1$
evaluates to :

A. $\sqrt{(n)}(\log n + 1)$
B. $\sqrt{(n)} \log n$
C. $\sqrt{(n)} \log \sqrt{(n)}$
D. $n \log \sqrt{n}$

gate2008-it algorithms recurrence normal

1.15.26 Recurrence: GATE2009-35

https://gateoverflow.in/1321

The running time of an algorithm is represented by the following recurrence relation:

$$T(n) = \begin{cases} n & n \leq 3 \\ T(\frac{n}{3}) + cn & \text{otherwise} \end{cases}$$

Which one of the following represents the time complexity of the algorithm?

A. $\Theta(n)$
B. $\Theta(n \log n)$
C. $\Theta(n^2)$
D. $\Theta(n^2 \log n)$

gate2009 algorithms recurrence time-complexity normal

1.15.27 Recurrence: GATE2012-16

https://gateoverflow.in/48

The recurrence relation capturing the optimal execution time of the $Towers\ of\ Hanoi$ problem with n discs is

A. $T(n) = 2T(n-2) + 2$
B. $T(n) = 2T(n-1) + n$
C. $T(n) = 2T(n/2) + 1$
D. $T(n) = 2T(n-1) + 1$

gate2012 algorithms easy recurrence

1.15.28 Recurrence: GATE2014-2-13

https://gateoverflow.in/1968

Which one of the following correctly determines the solution of the recurrence relation with $T(1) = 1$?

$$T(n) = 2T\left(\frac{n}{2}\right) + \log n$$

A. $\Theta(n)$ B. $\Theta(n \log n)$ C. $\Theta(n^2)$ D. $\Theta(\log n)$

gate2014-2 algorithms recurrence normal

1.15.29 Recurrence: GATE2015-1-2

https://gateoverflow.in/8017

Which one of the following is the recurrence equation for the worst case time complexity of the quick sort algorithm for sorting n (≥ 2) numbers? In the recurrence equations given in the options below, c is a constant.

A. $T(n) = 2T(n/2) + cn$
B. $T(n) = T(n-1) + T(1) + cn$
C. $T(n) = 2T(n-1) + cn$
D. $T(n) = T(n/2) + cn$

gate2015-1 algorithms recurrence sorting easy

1.15.30 Recurrence: GATE2015-1-49

https://gateoverflow.in/8355

Let a_n represent the number of bit strings of length n containing two consecutive 1s. What is the recurrence relation for a_n?

A. $a_{n-2} + a_{n-1} + 2^{n-2}$
B. $a_{n-2} + 2a_{n-1} + 2^{n-2}$

© Copyright GATE Overflow. All rights reserved.

C. $2a_{n-2}+a_{n-1}+2^{n-2}$ D. $2a_{n-2}+2a_{n-1}+2^{n-2}$

gate2015-1 algorithms recurrence normal

1.15.31 Recurrence: GATE2015-3-39

https://gateoverflow.in/8498

Consider the following recursive C function.

```
void get(int n)
{
    if (n<1) return;
    get (n-1);
    get (n-3);
    printf("%d", n);
}
```

If $get(6)$ function is being called in $main()$ then how many times will the $get()$ function be invoked before returning to the $main()$?

A. 15 B. 25 C. 35 D. 45

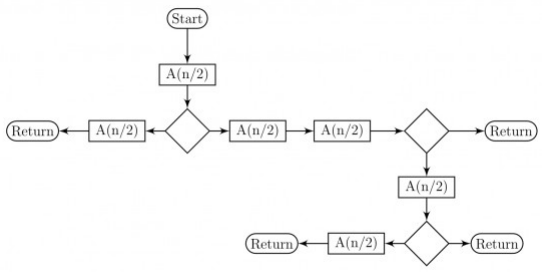
gate2015-3 algorithms recurrence normal

1.15.32 Recurrence: GATE2016-2-39

https://gateoverflow.in/39581

The given diagram shows the flowchart for a recursive function $A(n)$. Assume that all statements, except for the recursive calls, have $O(1)$ time complexity. If the worst case time complexity of this function is $O(n^{\alpha})$, then the least possible value (accurate up to two decimal positions) of α is _____.

Flow chart for Recursive Function $A(n)$.



gate2016-2 algorithms time-complexity recurrence normal numerical-answers

1.15.33 Recurrence: GATE2017-2-30

https://gateoverflow.in/118623

Consider the recurrence function

$$T(n)=\begin{cases}2T(\sqrt{n})+1, & n>2\\ 2, & 0<n\leq 2\end{cases}$$

Then $T(n)$ in terms of Θ notation is

A. $\Theta(\log\log n)$ B. $\Theta(\log n)$
C. $\Theta(\sqrt{n})$ D. $\Theta(n)$

gate2017-2 algorithms recurrence

1.15.34 Recurrence: TIFR2014-B-11

https://gateoverflow.in/27308

Consider the following recurrence relation:

$$T(n)=\begin{cases}T\left(\frac{n}{k}\right)+T\left(\frac{3n}{4}\right)+n & \text{if } n\geq 2\\ 1 & \text{if } n=1\end{cases}$$

Which of the following statements is FALSE?

a. $T(n)$ is $O(n^{3/2})$ when $k=3$. b. $T(n)$ is $O(n\log n)$ when $k=3$.
c. $T(n)$ is $O(n\log n)$ when $k=4$. d. $T(n)$ is $O(n\log n)$ when $k=5$.
e. $T(n)$ is $O(n)$ when $k=5$.

tifr2014 algorithms recurrence

© Copyright GATE Overflow. All rights reserved.

1.15.35 Recurrence: TIFR2015-B-1

https://gateoverflow.in/29657

Consider the following recurrence relation:

$$T(n) = \begin{cases} 2T(\lfloor \sqrt{n} \rfloor) + \log n & \text{if } n \geq 2 \\ 1 & \text{if } n = 1 \end{cases}$$

Which of the following statements is TRUE?

a. $T(n)$ is $O(\log n)$.
b. $T(n)$ is $O(\log n. \log\log n)$ but not $O(\log n)$.
c. $T(n)$ is $O(\log^{3/2} n)$ but not $O(\log n. \log\log n)$.
d. $T(n)$ is $O(\log^2 n)$ but not $O(\log^{3/2} n)$.
e. $T(n)$ is $O(\log^2 n. \log\log n)$ but not $O(\log^2 n)$.

tifr2015 algorithms recurrence time-complexity

1.15.36 Recurrence: TIFR2017-A-15

https://gateoverflow.in/95664

Let $T(a,b)$ be the function with two arguments (both nonnegative integral powers of 2) defined by the following reccurence:

- $T(a,b) = T\left(\frac{a}{2}, b\right) + T\left(a, \frac{b}{2}\right) \qquad \text{if } a,b \geq 2$;
- $T(a,1) = T\left(\frac{a}{2}, 1\right) \qquad \text{if } a \geq 2$;
- $T(1,b) = T\left(1, \frac{b}{2}\right) \qquad \text{if } b \geq 2$;
- $T(1,1) = 1$.

What is $T(2^r, 2^s)$?

A. rs
B. $r+s$
C. $\binom{2^r + 2^s}{2^r}$
D. $\binom{r+s}{r}$
E. 2^{r-s} if $r \geq s$, otherwise 2^{s-r}

tifr2017 algorithms recurrence

1.15.37 Recurrence: TIFR2018-B-5

https://gateoverflow.in/179289

Which of the following functions, given by there recurrence, grows the fastest asymptotically ?

A. $T(n) = 4T\left(\frac{n}{2}\right) + 10n$
B. $T(n) = 8T\left(\frac{n}{3}\right) + 24n^2$
C. $T(n) = 16T\left(\frac{n}{4}\right) + 10n^2$
D. $T(n) = 25T\left(\frac{n}{5}\right) + 20(nlogn)^{1.99}$
E. They all are asymptotically the same

tifr2018 asymptotic-notations recurrence

1.16 Searching (8)

1.16.1 Searching: GATE1996-18

https://gateoverflow.in/2770

Consider the following program that attempts to locate an element x in an array $a[]$ using binary search. Assume $N > 1$. The program is erroneous. Under what conditions does the program fail?

```
var i,j,k: integer; x: integer;
    a: array; [1..N] of integer;
begin i:= 1; j:= n;
repeat
    k:(i+j) div 2;
    if a[k] < x then i:= k
    else j:= k
until (a[k] = x) or (i >= j);

if (a[k] = x) then
    writeln ('x is in the array')
else
    writeln ('x is not in the array')
end;
```

© Copyright GATE Overflow. All rights reserved.

gate1996 algorithms searching normal

1.16.2 Searching: GATE1996-2.13, ISRO2016-28

https://gateoverflow.in/2742

The average number of key comparisons required for a successful search for sequential search on n items is

A. $\frac{n}{2}$ B. $\frac{n-1}{2}$ C. $\frac{n+1}{2}$ D. None of the above

gate1996 algorithms easy isro2016 searching

1.16.3 Searching: GATE2002-2.10

https://gateoverflow.in/840

Consider the following algorithm for searching for a given number x in an unsorted array $A[1..n]$ having n distinct values:

1. Choose an i at random from $1..n$
2. If $A[i] = x$, then Stop else Goto 1;

Assuming that x is present in A, what is the expected number of comparisons made by the algorithm before it terminates?

A. n B. $n-1$ C. $2n$ D. $\frac{n}{2}$

gate2002 searching normal

1.16.4 Searching: GATE2008-84

https://gateoverflow.in/394

Consider the following C program that attempts to locate an element x in an array $Y[\,]$ using binary search. The program is erroneous.

```
f (int Y[10] , int x) {
   int u, j, k;
   i= 0; j = 9;
   do {
        k = (i+ j) / 2;
        if( Y[k] < x) i = k;else j = k;
        } while (Y[k] != x) && (i < j)) ;
   if(Y[k] == x) printf(" x is in the array ") ;
   else printf(" x is not in the array ") ;
}
```

On which of the following contents of Y and x does the program fail?

A. Y is $[1\,2\,3\,4\,5\,6\,7\,8\,9\,10]$ and $x < 10$
B. Y is $[1\,3\,5\,7\,9\,11\,13\,15\,17\,19]$ and $x < 1$
C. Y is $[2\,2\,2\,2\,2\,2\,2\,2\,2\,2]$ and $x > 2$
D. Y is $[2\,4\,6\,8\,10\,12\,14\,16\,18\,20]$ and $2 < x < 20$ and x is even

gate2008 algorithms searching normal

1.16.5 Searching: GATE2008-85

https://gateoverflow.in/43508

Consider the following C program that attempts to locate an element x in an array $Y[\,]$ using binary search. The program is erroneous.

```
f (int Y[10] , int x) {
   int u, j, k;
   i= 0; j = 9;
   do {
        k = (i+ j) / 2;
        if( Y[k] < x) i = k;else j = k;
        } while (Y[k] != x) && (i < j)) ;
   if(Y[k] == x) printf(" x is in the array ") ;
   else printf(" x is not in the array ") ;
}
```

The correction needed in the program to make it work properly is

A. Change line 6 to: if $(Y[k] < x) i = k+1$; else $j = k-1$;
B. Change line 6 to: if $(Y[k] < x) i = k-1$; else $j = k+1$;
C. Change line 6 to: if $(Y[k] < x) i = k$; else $j = k$;
D. Change line 7 to: } while $((Y[k] == x) \&\& (i < j))$;

© Copyright GATE Overflow. All rights reserved.

gate2008 algorithms searching normal

1.16.6 Searching: GATE2017-1-48

https://gateoverflow.in/118331

Let A be an array of 31 numbers consisting of a sequence of 0's followed by a sequence of 1's. The problem is to find the smallest index i such that $A[i]$ is 1 by probing the minimum number of locations in A. The *worst case* number of probes performed by an *optimal* algorithm is _____.

gate2017-1 algorithms normal numerical-answers searching

1.16.7 Searching: TIFR2010-B-29

https://gateoverflow.in/18752

Suppose you are given an array A with $2n$ numbers.

The numbers in odd positions are sorted in ascending order, that is, $A[1] \leq A[3] \leq \ldots \leq A[2n-1]$.

The numbers in even positions are sorted in descending order, that is, $A[2] \geq A[4] \geq \ldots \geq A[2n]$.

What is the method you would recommend for determining if a given number is in the array?

A. Sort the array using quick-sort and then use binary search.
B. Merge the sorted lists and perform binary search.
C. Perform a single binary search on the entire array.
D. Perform separate binary searches on the odd positions and the even positions.
E. Search sequentially from the end of the array.

tifr2010 searching

1.16.8 Searching: TIFR2012-B-11

https://gateoverflow.in/25140

Consider the following three version of the binary search program. Assume that the elements of type T can be compared with each other; also assume that the array is sorted.

```
i, j, k : integer;
a : array [1....N] of T;
x : T;

Program 1 :   i := 1; j := N;
              repeat
                    k := (i + j) div 2;
                    if a[k] < x then i := k else j := k
              until (a[k] = x) or (i > j)
Program 2 :   i := 1; j := N;
              repeat
                    k := (i + j) div 2;
                   if x < a[k] then j := k - 1;
                   if a[k] < x then i := k + 1;
              until i > j
Program 3 :  i := 1; j := N
              repeat
                    k := (i + j) div 2;
                   if x < a[k] then j := k else i := k + 1
              until i > j
```

A binary search program is called correct provided it terminates with $a[k] = x$ whenever such an element exists, or it terminates with $a[k] \neq x$ if there exists no array element with value x. Which of the following statements is correct?

A. Only Program 1 is correct
B. Only Program 2 is correct
C. Only Program 1 and 2 are correct.
D. Both Program 2 and 3 are correct
E. All the three programs are wrong

tifr2012 algorithms searching

1.17 Shortest Path (1)

1.17.1 Shortest Path: TIFR2018-B-9

https://gateoverflow.in/179293

Let $G = (V, E)$ be a DIRECTED graph, where each edge e has a positive weight $\omega(e)$, and all vertices can be reached from vertex s. For each vertex v, let $\phi(v)$ be the length of the shortest path from s to v. Let $G' = (V, E)$ be a new weighted graph with the same vertices and edges, but with the edge weight of every edge $e = (u \rightarrow v)$ changed to $\omega'(e) = \omega(e) + \phi(v) - \phi(u)$. Let P be a path from s to a vertex v, and let $\omega(P) = \sum_{e \in P} \omega_e$, and $\omega'(P) = \sum_{e \in P} \omega'_e$.

© Copyright GATE Overflow. All rights reserved.

Which of the following options is NOT NECESSARILY TRUE ?

A. If P is a shortest path in G, then P is a shortest path in G'.
B. If P is a shortest path in F', then P is a shortest path in G.
C. If P is a shortest path in G, then $\omega'(P) = 2 \times \omega(P)$.
D. If P is NOT a shortest path in G, then $\omega'(P) < 2 \times \omega(P)$.
E. All of the above options are necessarily TRUE.

tifr2018 graph-algorithms shortest-path

1.18 Sorting (52)

1.18.1 Sorting: GATE1987-1-xviii

https://gateoverflow.in/80366

Let P be a quicksort program to sort numbers in ascending order. Let t_1 and t_2 be the time taken by the program for the inputs $[1\,2\,3\,4]$ and $[5\,4\,3\,2\,1]$, respectively. Which of the following holds?

A. $t_1 = t_2$
B. $t_1 > t_2$
C. $t_1 < t_2$
D. $t_1 = t_2 + 5 \log 5$

gate1987 algorithms sorting

1.18.2 Sorting: GATE1988-1iii

https://gateoverflow.in/91338

Quicksort is _____ efficient than heapsort in the worst case.

gate1988 algorithms sorting

1.18.3 Sorting: GATE1989-9

https://gateoverflow.in/89083

An input files has 10 records with keys as given below:

$25 \quad 7 \quad 34 \quad 2 \quad 70 \quad 9 \quad 61 \quad 16 \quad 49 \quad 19$

This is to be sorted in non-decreasing order.

i. Sort the input file using QUICKSORT by correctly positioning the first element of the file/subfile. Show the subfiles obtained at all intermediate steps. Use square brackets to demarcate subfiles.
ii. Sort the input file using 2-way- MERGESORT showing all major intermediate steps. Use square brackets to demarcate subfiles.

gate1989 descriptive algorithms sorting

1.18.4 Sorting: GATE1990-3-v

https://gateoverflow.in/84830

Choose the correct alternatives (More than one may be correct).

The complexity of comparision based sorting algorithms is:

A. $\Theta(n \log n)$
B. $\Theta(n)$
C. $\Theta\left(n^2\right)$
D. $\Theta(n\sqrt{n})$

gate1990 normal algorithms sorting

1.18.5 Sorting: GATE1991-01,vii

https://gateoverflow.in/505

The minimum number of comparisons required to sort 5 elements is ____ minimum number of comparison= (log n!)

gate1991 normal algorithms sorting

1.18.6 Sorting: GATE1991-13

https://gateoverflow.in/540

Give an optimal algorithm in pseudo-code for sorting a sequence of n numbers which has only k distinct numbers (k is not known a Priori). Give a brief analysis for the time-complexity of your algorithm.

gate1991 sorting time-complexity algorithms difficult

1.18.7 Sorting: GATE1992-02,ix

https://gateoverflow.in/559

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

© Copyright GATE Overflow. All rights reserved.

Following algorithm(s) can be used to sort n in the range $[1 \ldots n^3]$ in $O(n)$ time

a. Heap sort b. Quick sort c. Merge sort d. Radix sort

gate1992 easy algorithms sorting

1.18.8 Sorting: GATE1992-03,iv

https://gateoverflow.in/581

Assume that the last element of the set is used as partition element in Quicksort. If n distinct elements from the set $[1 \ldots n]$ are to be sorted, give an input for which Quicksort takes maximum time.

gate1992 algorithms sorting easy

1.18.9 Sorting: GATE1994-1.19, ISRO2016-31

https://gateoverflow.in/2462

Algorithm design technique used in quicksort algorithm is?

A. Dynamic programming
B. Backtracking
C. Divide and conquer
D. Greedy method

gate1994 algorithms sorting easy isro2016

1.18.10 Sorting: GATE1995-1.16

https://gateoverflow.in/2603

For merging two sorted lists of sizes m and n into a sorted list of size $m+n$, we require comparisons of

A. $O(m)$ B. $O(n)$ C. $O(m+n)$ D. $O(\log m + \log n)$

gate1995 algorithms sorting normal

1.18.11 Sorting: GATE1995-1.5

https://gateoverflow.in/2592

Merge sort uses:

A. Divide and conquer strategy
B. Backtracking approach
C. Heuristic search
D. Greedy approach

gate1995 algorithms sorting easy

1.18.12 Sorting: GATE1995-12

https://gateoverflow.in/2648

Consider the following sequence of numbers:

$$92, 37, 52, 12, 11, 25$$

Use Bubble sort to arrange the sequence in ascending order. Give the sequence at the end of each of the first five passes.

gate1995 algorithms sorting easy

1.18.13 Sorting: GATE1996-14

https://gateoverflow.in/2766

A two dimensional array $A[1..n][1..n]$ of integers is partially sorted if $\forall i, j \in [1..n-1], A[i][j] < A[i][j+1]$ and $A[i][j] < A[i+1][j]$

Fill in the blanks:

a. The smallest item in the array is at $A[i][j]$ where i=__ and j=__.
b. The smallest item is deleted. Complete the following $O(n)$ procedure to insert item x (which is guaranteed to be smaller than any item in the last row or column) still keeping A partially sorted.

```
procedure insert (x: integer);
var i,j: integer;
begin
    i:=1; j:=1, A[i][j]:=x;
    while (x > __ or x > __) do
        if A[i+1][j] < A[i][j] ___ then begin
            A[i][j]:=A[i+1][j]; i:=i+1;
        end
        else begin
            _____
        end
    A[i][j]:= ____
```

© Copyright GATE Overflow. All rights reserved.

```
end
```

gate1996 algorithms sorting normal

1.18.14 Sorting: GATE1996-2.15

https://gateoverflow.in/2744

Quick-sort is run on two inputs shown below to sort in ascending order taking first element as pivot

i. $1,2,3,\ldots n$
ii. $n,n-1,n-2,\ldots,2,1$

Let C_1 and C_2 be the number of comparisons made for the inputs (i) and (ii) respectively. Then,

A. $C_1 < C_2$
B. $C_1 > C_2$
C. $C_1 = C_2$
D. we cannot say anything for arbitrary n

gate1996 algorithms sorting normal

1.18.15 Sorting: GATE1998-1.22

https://gateoverflow.in/1659

Give the correct matching for the following pairs:

(A)	$O(\log n)$	(P)	Selection sort
(B)	$O(n)$	(Q)	Insertion sort
(C)	$O(n \log n)$	(R)	Binary search
(D)	$O(n^2)$	(S)	Merge sort

A. A-R B-P C-Q D-S
B. A-R B-P C-S D-Q
C. A-P B-R C-S D-Q
D. A-P B-S C-R D-Q

gate1998 algorithms sorting easy

1.18.16 Sorting: GATE1999-1.12

https://gateoverflow.in/1465

A sorting technique is called stable if

A. it takes $O(n \log n)$ time
B. it maintains the relative order of occurrence of non-distinct elements
C. it uses divide and conquer paradigm
D. it takes $O(n)$ space

gate1999 algorithms sorting easy

1.18.17 Sorting: GATE1999-1.14, ISRO2015-42

https://gateoverflow.in/1467

If one uses straight two-way merge sort algorithm to sort the following elements in ascending order:

20, 47, 15, 8, 9, 4, 40, 30, 12, 17

then the order of these elements after second pass of the algorithm is:

A. 8, 9, 15, 20, 47, 4, 12, 17, 30, 40
B. 8, 15, 20, 47, 4, 9, 30, 40, 12, 17
C. 15, 20, 47, 4, 8, 9, 12, 30, 40, 17
D. 4, 8, 9, 15, 20, 47, 12, 17, 30, 40

gate1999 algorithms sorting normal isro2015

1.18.18 Sorting: GATE1999-8

https://gateoverflow.in/1507

Let A be an $n \times n$ matrix such that the elements in each row and each column are arranged in ascending order. Draw a decision tree, which finds 1^{st}, 2^{nd} and 3^{rd} smallest elements in minimum number of comparisons.

gate1999 algorithms sorting normal descriptive

© Copyright GATE Overflow. All rights reserved.

1.18.19 Sorting: GATE2000-17

https://gateoverflow.in/688

An array contains four occurrences of 0, five occurrences of 1, and three occurrences of 2 in any order. The array is to be sorted using swap operations (elements that are swapped need to be adjacent).

a. What is the minimum number of swaps needed to sort such an array in the worst case?
b. Give an ordering of elements in the above array so that the minimum number of swaps needed to sort the array is maximum.

gate2000 algorithms sorting normal descriptive

1.18.20 Sorting: GATE2001-1.14

https://gateoverflow.in/707

Randomized quicksort is an extension of quicksort where the pivot is chosen randomly. What is the worst case complexity of sorting n numbers using Randomized quicksort?

A. $O(n)$ B. $O(n \log n)$ C. $O(n^2)$ D. $O(n!)$

gate2001 algorithms sorting time-complexity easy

1.18.21 Sorting: GATE2003-22

https://gateoverflow.in/912

The unusual $\Theta(n^2)$ implementation of Insertion Sort to sort an array uses linear search to identify the position where an element is to be inserted into the already sorted part of the array. If, instead, we use binary search to identify the position, the worst case running time will

A. remain $\Theta(n^2)$
B. become $\Theta(n(\log n)^2)$
C. become $\Theta(n \log n)$
D. become $\Theta(n)$

gate2003 algorithms sorting time-complexity normal

1.18.22 Sorting: GATE2003-61

https://gateoverflow.in/949

In a permutation $a_1 \ldots a_n$, of n distinct integers, an inversion is a pair (a_i, a_j) such that $i < j$ and $a_i > a_j$.

If all permutations are equally likely, what is the expected number of inversions in a randomly chosen permutation of $1 \ldots n$?

A. $\frac{n(n-1)}{2}$ B. $\frac{n(n-1)}{4}$ C. $\frac{n(n+1)}{4}$ D. $2n\lceil \log_2 n \rceil$

gate2003 algorithms sorting normal

1.18.23 Sorting: GATE2003-62

https://gateoverflow.in/43576

In a permutation $a_1 \ldots a_n$, of n distinct integers, an inversion is a pair (a_i, a_j) such that $i < j$ and $a_i > a_j$.

What would be the worst case time complexity of the Insertion Sort algorithm, if the inputs are restricted to permutations of $1 \ldots n$ with at most n inversions?

A. $\Theta(n^2)$
B. $\Theta(n \log n)$
C. $\Theta(n^{1.5})$
D. $\Theta(n)$

gate2003 algorithms sorting normal

1.18.24 Sorting: GATE2004-29

https://gateoverflow.in/1026

The tightest lower bound on the number of comparisons, in the worst case, for comparison-based sorting is of the order of

A. n B. n^2 C. $n \log n$ D. $n \log^2 n$

gate2004 algorithms sorting asymptotic-notations easy

1.18.25 Sorting: GATE2005-39

https://gateoverflow.in/784

Suppose there are $\lceil \log n \rceil$ sorted lists of $\lfloor n / \log n \rfloor$ elements each. The time complexity of producing a sorted list of all these elements is: (Hint:Use a heap data structure)

A. $O(n \log \log n)$
B. $\Theta(n \log n)$
C. $\Omega(n \log n)$
D. $\Omega\left(n^{3/2}\right)$

© Copyright GATE Overflow. All rights reserved.

gate2005 algorithms sorting normal

1.18.26 Sorting: GATE2005-IT-59

https://gateoverflow.in/3820

Let a and b be two sorted arrays containing n integers each, in non-decreasing order. Let c be a sorted array containing $2n$ integers obtained by merging the two arrays a and b. Assuming the arrays are indexed starting from 0, consider the following four statements

I. $a[i] \geq b[i] \Rightarrow c[2i] \geq a[i]$
II. $a[i] \geq b[i] \Rightarrow c[2i] \geq b[i]$
III. $a[i] \geq b[i] \Rightarrow c[2i] \leq a[i]$
IV. $a[i] \geq b[i] \Rightarrow c[2i] \leq b[i]$

Which of the following is TRUE?

A. only I and II B. only I and IV C. only II and III D. only III and IV

gate2005-it algorithms sorting normal

1.18.27 Sorting: GATE2006-14, ISRO2011-14

https://gateoverflow.in/975

Which one of the following in place sorting algorithms needs the minimum number of swaps?

A. Quick sort B. Insertion sort C. Selection sort D. Heap sort

gate2006 algorithms sorting easy isro2011

1.18.28 Sorting: GATE2006-52

https://gateoverflow.in/1830

The median of n elements can be found in $O(n)$ time. Which one of the following is correct about the complexity of quick sort, in which median is selected as pivot?

A. $\Theta(n)$
B. $\Theta(n \log n)$
C. $\Theta(n^2)$
D. $\Theta(n^3)$

gate2006 algorithms sorting easy

1.18.29 Sorting: GATE2007-14

https://gateoverflow.in/1212

Which of the following sorting algorithms has the lowest worse-case complexity?

A. Merge sort B. Bubble sort C. Quick sort D. Selection sort

gate2007 algorithms sorting time-complexity easy

1.18.30 Sorting: GATE2008-43

https://gateoverflow.in/455

Consider the Quicksort algorithm. Suppose there is a procedure for finding a pivot element which splits the list into two sub-lists each of which contains at least one-fifth of the elements. Let $T(n)$ be the number of comparisons required to sort n elements. Then

A. $T(n) \leq 2T(n/5) + n$
B. $T(n) \leq T(n/5) + T(4n/5) + n$
C. $T(n) \leq 2T(4n/5) + n$
D. $T(n) \leq 2T(n/2) + n$

gate2008 algorithms sorting easy

1.18.31 Sorting: GATE2008-IT-43

https://gateoverflow.in/3353

If we use Radix Sort to sort n integers in the range $\left(n^{k/2}, n^k\right]$, for some $k > 0$ which is independent of n, the time taken would be?

A. $\Theta(n)$ B. $\Theta(kn)$ C. $\Theta(n \log n)$ D. $\Theta(n^2)$

gate2008-it algorithms sorting normal

1.18.32 Sorting: GATE2009-11

https://gateoverflow.in/1303

What is the number of swaps required to sort n elements using selection sort, in the worst case?

© Copyright GATE Overflow. All rights reserved.

A. $\Theta(n)$
B. $\Theta(n \log n)$
C. $\Theta(n^2)$
D. $\Theta(n^2 \log n)$

gate2009 algorithms sorting easy

1.18.33 Sorting: GATE2009-39

https://gateoverflow.in/1325

In quick-sort, for sorting n elements, the $(n/4)^{th}$ smallest element is selected as pivot using an $O(n)$ time algorithm. What is the worst case time complexity of the quick sort?

A. $\Theta(n)$
B. $\Theta(n \log n)$
C. $\Theta(n^2)$
D. $\Theta(n^2 \log n)$

gate2009 algorithms sorting normal

1.18.34 Sorting: GATE2012-39

https://gateoverflow.in/1762

A list of n strings, each of length n, is sorted into lexicographic order using the merge-sort algorithm. The worst case running time of this computation is

A. $O(n \log n)$ B. $O(n^2 \log n)$ C. $O(n^2 + \log n)$ D. $O(n^2)$

gate2012 algorithms sorting normal

1.18.35 Sorting: GATE2013-30

https://gateoverflow.in/1541

The number of elements that can be sorted in $\Theta(\log n)$ time using heap sort is

A. $\Theta(1)$
B. $\Theta(\sqrt{\log n})$
C. $\Theta(\frac{\log n}{\log \log n})$
D. $\Theta(\log n)$

gate2013 algorithms sorting normal

1.18.36 Sorting: GATE2013-6

https://gateoverflow.in/1415

Which one of the following is the tightest upper bound that represents the number of swaps required to sort n numbers using selection sort?

A. $O(\log n)$ B. $O(n)$ C. $O(n \log n)$ D. $O(n^2)$

gate2013 algorithms sorting easy

1.18.37 Sorting: GATE2014-1-14

https://gateoverflow.in/1780

Let P be quicksort program to sort numbers in ascending order using the first element as the pivot. Let t_1 and t_2 be the number of comparisons made by P for the inputs $[1\,2\,3\,4\,5]$ and $[4\,1\,5\,3\,2]$ respectively. Which one of the following holds?

A. $t_1 = 5$ B. $t_1 < t_2$ C. $t_1 > t_2$ D. $t_1 = t_2$

gate2014-1 algorithms sorting easy

1.18.38 Sorting: GATE2014-2-38

https://gateoverflow.in/1997

Suppose P, Q, R, S, T are sorted sequences having lengths $20, 24, 30, 35, 50$ respectively. They are to be merged into a single sequence by merging together two sequences at a time. The number of comparisons that will be needed in the worst case by the optimal algorithm for doing this is ____.

gate2014-2 algorithms sorting normal numerical-answers

1.18.39 Sorting: GATE2014-3-14

https://gateoverflow.in/2048

You have an array of n elements. Suppose you implement quicksort by always choosing the central element of the array as the pivot. Then the tightest upper bound for the worst case performance is

A. $O(n^2)$ B. $O(n \log n)$ C. $\Theta(n \log n)$ D. $O(n^3)$

gate2014-3 algorithms sorting easy

© Copyright GATE Overflow. All rights reserved.

1.18.40 Sorting: GATE2015-2-45

https://gateoverflow.in/8243

Suppose you are provided with the following function declaration in the C programming language.

```
int partition(int a[], int n);
```

The function treats the first element of $a[\,]$ as a pivot and rearranges the array so that all elements less than or equal to the pivot is in the left part of the array, and all elements greater than the pivot is in the right part. In addition, it moves the pivot so that the pivot is the last element of the left part. The return value is the number of elements in the left part.

The following partially given function in the C programming language is used to find the k^{th} smallest element in an array $a[\,]$ of size n using the partition function. We assume $k \leq n$.

```
int kth_smallest (int a[], int n, int k)
{
    int left_end = partition (a, n);
    if (left_end+1==k) {
        return a[left_end];
    }
    if (left_end+1 > k) {
        return kth_smallest (___________);
    } else {
        return kth_smallest (___________);
    }
}
```

The missing arguments lists are respectively

A. $(a,$ left_end$,k)$ and $(a+$left_end$+1, n-$left_end$-1, k-$left_end$-1)$
B. $(a,$ left_end$,k)$ and $(a, n-$left_end$-1, k-$left_end$-1)$
C. $(a+$ left_end$+1, n-$left_end$-1, k-$left_end$-1)$ and $(a,$ left_end$,k)$
D. $(a, n-$left_end$-1, k-$left_end$-1)$ and $(a,$left_end$,k)$

gate2015-2 algorithms normal sorting

1.18.41 Sorting: GATE2015-3-27

https://gateoverflow.in/8480

Assume that a mergesort algorithm in the worst case takes 30 seconds for an input of size 64. Which of the following most closely approximates the maximum input size of a problem that can be solved in 6 minutes?

A. 256 B. 512 C. 1024 D. 2018

gate2015-3 algorithms sorting

1.18.42 Sorting: GATE2016-1-13

https://gateoverflow.in/39660

The worst case running times of *Insertion sort* , *Merge sort* and *Quick sort*, respectively are:

A. $\Theta(n \log n)$, $\Theta(n \log n)$ and $\Theta(n^2)$
B. $\Theta(n^2)$, $\Theta(n^2)$ and $\Theta(n \log n)$
C. $\Theta(n^2)$, $\Theta(n \log n)$ and $\Theta(n \log n)$
D. $\Theta(n^2)$, $\Theta(n \log n)$ and $\Theta(n^2)$

gate2016-1 algorithms sorting easy

1.18.43 Sorting: GATE2016-2-13

https://gateoverflow.in/39561

Assume that the algorithms considered here sort the input sequences in ascending order. If the input is already in the ascending order, which of the following are **TRUE**?

I. Quicksort runs in $\Theta(n^2)$ time
II. Bubblesort runs in $\Theta(n^2)$ time
III. Mergesort runs in $\Theta(n)$ time
IV. Insertion sort runs in $\Theta(n)$ time

A. I and II only B. I and III only C. II and IV only D. I and IV only

gate2016-2 algorithms sorting time-complexity normal ambiguous

© Copyright GATE Overflow. All rights reserved.

1.18.44 Sorting: TIFR2010-B-23

https://gateoverflow.in/18623

Suppose you are given n numbers and you sort them in descending order as follows:

First find the maximum. Remove this element from the list and find the maximum of the remaining elements, remove this element, and so on, until all elements are exhausted. How many comparisons does this method require in the worst case?

A. Linear in n.
B. $O\left(n^2\right)$ but not better.
C. $O(n \log n)$
D. Same as heap sort.
E. $O\left(n^{1.5}\right)$ but not better.

tifr2010 algorithms time-complexity sorting

1.18.45 Sorting: TIFR2010-B-27

https://gateoverflow.in/19036

Consider the Insertion Sort procedure given below, which sorts an array L of size $n (\geq 2)$ in ascending order:

```
begin
    for xindex:= 2 to n do
        x := L [xindex];
        j:= xindex - 1;
        while j > 0 and L[j] > x do
            L[j + 1]:= L[j];
            j:= j - 1;
        end {while}
        L [j + 1]:=X;
    end{for}
end
```

It is known that insertion sort makes at most $n(n-1)/2$ comparisons. Which of the following is true?

A. There is no input on which insertion Sort makes $n(n-1)/2$ comparisons.
B. Insertion Sort makes $n(n-1)/2$ comparisons when the input is already sorted in ascending order.
C. Insertion Sort makes $n(n-1)/2$ comparisons only when the input is sorted in descending order.
D. There are more than one input orderings where insertion sort makes $n(n-1)/2$ comparisons.
E. Insertion Sort makes $n(n-1)/2$ comparisons whenever all the elements of L are not distinct.

tifr2010 algorithms sorting

1.18.46 Sorting: TIFR2011-B-21

https://gateoverflow.in/20324

Let $S = \{x_1, \ldots, x_n\}$ be a set of n numbers. Consider the problem of storing the elements of S in an array $A[1 \ldots n]$ such that the following min-heap property is maintained for all $2 \leq i \leq n : A[\lfloor i/2 \rfloor] \leq A[i]$. (Note that $\lfloor x \rfloor$ is the largest integer that is at most x). Which of the following statements is TRUE?

A. This problem can be solved in $O(\log n)$ time.
B. This problem can be solved in $O(n)$ time but not in $O(\log n)$ time.
C. This problem can be solved in $O(n \log n)$ time but not in $O(n)$ time.
D. This problem can be solved in $O\left(n^2\right)$ time but not in $O(n \log n)$ time.
E. None of the above.

tifr2011 algorithms sorting

1.18.47 Sorting: TIFR2011-B-31

https://gateoverflow.in/20617

Given a set of $n = 2^k$ distinct numbers, we would like to determine the smallest and the second smallest using comparisons. Which of the following statements is TRUE?

A. Both these elements can be determined using $2k$ comparisons.
B. Both these elements can be determined using $n-2$ comparisons.
C. Both these elements can be determined using $n+k-2$ comparisons.
D. $2n-3$ comparisons are necessary to determine these two elements.
E. nk comparisons are necessary to determine these two elements.

tifr2011 algorithms sorting

© Copyright GATE Overflow. All rights reserved.

1.18.48 Sorting: TIFR2011-B-39

https://gateoverflow.in/20935

The first n cells of an array L contain positive integers sorted in decreasing order, and the remaining $m-n$ cells all contain 0. Then, given an integer x, in how many comparisons can one find the position of x in L?

A. At least n comparisons are necessary in the worst case.
B. At least $\log m$ comparisons are necessary in the worst case.
C. $O(\log(m-n))$ comparisons suffice.
D. $O(\log n)$ comparisons suffice.
E. $O(\log(m/n))$ comparisons suffice.

tifr2011 algorithms sorting

1.18.49 Sorting: TIFR2012-B-13

https://gateoverflow.in/25207

An array A contains n integers. We wish to sort A in ascending order. We are told that initially no element of A is more than a distance k away from its final position in the sorted list. Assume that n and k are large and k is much smaller than n. Which of the following is true for the worst case complexity of sorting A?

A. A can be sorted with constant $.kn$ comparison but not with fewer comparisons.
B. A cannot be sorted with less than constant $.n\log n$ comparisons.
C. A can be sorted with constant $.n$ comparisons.
D. A can be sorted with constant $.n\log k$ comparisons but not with fewer comparisons.
E. A can be sorted with constant $.k^2n$ comparisons but not fewer.

tifr2012 algorithms sorting

1.18.50 Sorting: TIFR2012-B-14

https://gateoverflow.in/25209

Consider the quick sort algorithm on a set of n numbers, where in every recursive subroutine of the algorithm, the algorithm chooses the median of that set as the pivot. Then which of the following statements is TRUE?

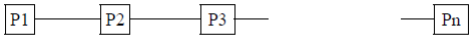
A. The running time of the algorithm is $\Theta(n)$.
B. The running time of the algorithm is $\Theta(n\log n)$.
C. The running time of the algorithm is $\Theta(n^{1.5})$.
D. The running time of the algorithm is $\Theta(n^2)$.
E. None of the above.

tifr2012 algorithms sorting

1.18.51 Sorting: TIFR2013-B-20

https://gateoverflow.in/25878

Suppose n processors are connected in a linear array as shown below. Each processor has a number. The processors need to exchange numbers so that the numbers eventually appear in ascending order (the processor P1 should have the minimum value and the the processor Pn should have the maximum value).



The algorithm to be employed is the following. Odd numbered processors and even numbered processors are activated alternate steps; assume that in the first step all the even numbered processors are activated. When a processor is activated, the number it holds is compared with the number held by its right-hand neighbour (if one exists) and the smaller of the two numbers is retained by the activated processor and the bigger stored in its right hand neighbour.
How long does it take for the processors to sort the values?

A. $n\log n$ steps
B. n^2 steps
C. n steps
D. $n^{1.5}$ steps
E. The algorithm is not guaranteed to sort

tifr2013 algorithms sorting

At first look, it appears that each step will take O(n) time and total n steps will be required in worst case so n*n=O(n^2) but twist here is that at any step All the even numbered (or odd numbered) processors are working(Comparing it's value with its right neighbour and swapping values if required) simultaneously. So at each step, a constant amount of time (O(1)) is required. so T.C will be n*O(1)=O(n) only.

1.18.52 Sorting: TIFR2017-B-7

https://gateoverflow.in/95699

An array of n distinct elements is said to be un-sorted if for every index i such that $2 \leq i \leq n-1$, either $A[i] > max\{A[i-1], A[i+1]\}$, or $A[i] < min\{A[i-1], A[i+1]\}$. What is the time-complexity of the fastest algorithm that takes as input a sorted array A with n distinct elements, and un-sorts A?

© Copyright GATE Overflow. All rights reserved.

A. $O(n \log n)$ but not $O(n)$
B. $O(n)$ but not $O(\sqrt{n})$
C. $O(\sqrt{n})$ but not $O(\log n)$
D. $O(\log n)$ but not $O(1)$
E. $O(1)$

tifr2017 algorithms sorting

1.19 Spanning Tree (31)

1.19.1 Spanning Tree: GATE1991-03,vi

https://gateoverflow.in/521

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

Kruskal's algorithm for finding a minimum spanning tree of a weighted graph G with n vertices and m edges has the time complexity of:

A. $O(n^2)$ B. $O(mn)$ C. $O(m+n)$ D. $O(m \log n)$ E. $O(m^2)$

gate1991 algorithms spanning-tree

1.19.2 Spanning Tree: GATE1992-01,ix

https://gateoverflow.in/549

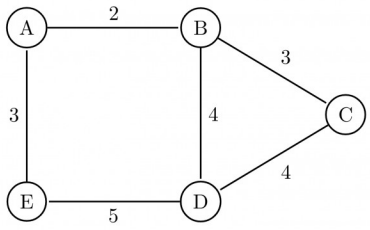
Complexity of Kruskal's algorithm for finding the minimum spanning tree of an undirected graph containing n vertices and m edges if the edges are sorted is _____

gate1992 spanning-tree algorithms time-complexity easy

1.19.3 Spanning Tree: GATE1995-22

https://gateoverflow.in/2660

How many minimum spanning trees does the following graph have? Draw them. (Weights are assigned to edges).



gate1995 algorithms graph-algorithms spanning-tree easy

1.19.4 Spanning Tree: GATE1996-16

https://gateoverflow.in/2768

A complete, undirected, weighted graph G is given on the vertex $\{0,1,\ldots,n-1\}$ for any fixed 'n'. Draw the minimum spanning tree of G if

A. the weight of the edge (u,v) is $|u-v|$
B. the weight of the edge (u,v) is $u+v$

gate1996 algorithms graph-algorithms spanning-tree normal

1.19.5 Spanning Tree: GATE1997-9

https://gateoverflow.in/2269

Consider a graph whose vertices are points in the plane with integer co-ordinates (x,y) such that $1 \leq x \leq n$ and $1 \leq y \leq n$, where $n \geq 2$ is an integer. Two vertices (x_1,y_1) and (x_2,y_2) are adjacent iff $|x_1 - x_2| \leq 1$ and $|y_1 - y_2| \leq 1$. The weight of an edge $\{(x_1,y_1),(x_2,y_2)\}$ is $\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$

A. What is the weight of a minimum weight-spanning tree in this graph? Write only the answer without any explanations.
B. What is the weight of a maximum weight-spanning tree in this graph? Write only the answer without any explanations.

gate1997 algorithms spanning-tree normal

1.19.6 Spanning Tree: GATE2000-2.18

https://gateoverflow.in/665

Let G be an undirected connected graph with distinct edge weights. Let e_{max} be the edge with maximum weight and

© Copyright GATE Overflow. All rights reserved.

e_{min} the edge with minimum weight. Which of the following statements is false?

A. Every minimum spanning tree of G must contain e_{min}
B. If e_{max} is in a minimum spanning tree, then its removal must disconnect G
C. No minimum spanning tree contains e_{max}
D. G has a unique minimum spanning tree

gate2000 algorithms spanning-tree normal

1.19.7 Spanning Tree: GATE2001-15

https://gateoverflow.in/756

Consider a weighted undirected graph with vertex set $V = \{n1, n2, n3, n4, n5, n6\}$ and edge set $E = \{(n1, n2, 2), (n1, n3, 8), (n1, n6, 3), (n2, n4, 4), (n2, n5, 12), (n3, n4, 7), (n4, n5, 9), (n4, n6, 4)\}$. The third value in each tuple represents the weight of the edge specified in the tuple.

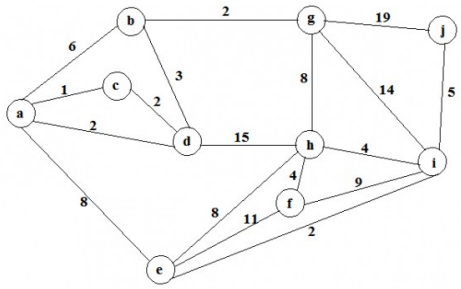
A. List the edges of a minimum spanning tree of the graph.
B. How many distinct minimum spanning trees does this graph have?
C. Is the minimum among the edge weights of a minimum spanning tree unique over all possible minimum spanning trees of a graph?
D. Is the maximum among the edge weights of a minimum spanning tree unique over all possible minimum spanning tree of a graph?

gate2001 algorithms spanning-tree normal descriptive

1.19.8 Spanning Tree: GATE2003-68

https://gateoverflow.in/955

What is the weight of a minimum spanning tree of the following graph?



A. 29 B. 31 C. 38 D. 41

gate2003 algorithms spanning-tree normal

1.19.9 Spanning Tree: GATE2005-6

https://gateoverflow.in/1348

An undirected graph G has n nodes. its adjacency matrix is given by an $n \times n$ square matrix whose (i) diagonal elements are 0's and (ii) non-diagonal elements are 1's. Which one of the following is TRUE?

A. Graph G has no minimum spanning tree (MST)
B. Graph G has unique MST of cost $n-1$
C. Graph G has multiple distinct MSTs, each of cost $n-1$
D. Graph G has multiple spanning trees of different costs

gate2005 algorithms spanning-tree normal

1.19.10 Spanning Tree: GATE2005-IT-52

https://gateoverflow.in/3813

Let G be a weighted undirected graph and e be an edge with maximum weight in G. Suppose there is a minimum weight spanning tree in G containing the edge e. Which of the following statements is always TRUE?

A. There exists a cutset in G having all edges of maximum weight.
B. There exists a cycle in G having all edges of maximum weight.
C. Edge e cannot be contained in a cycle.

© Copyright GATE Overflow. All rights reserved.

D. All edges in G have the same weight.

gate2005-it algorithms spanning-tree normal

1.19.11 Spanning Tree: GATE2006-11

https://gateoverflow.in/890

Consider a weighted complete graph G on the vertex set $\{v_1, v_2, \ldots . v_n\}$ such that the weight of the edge (v_i, v_j) is $2|i-j|$. The weight of a minimum spanning tree of G is:

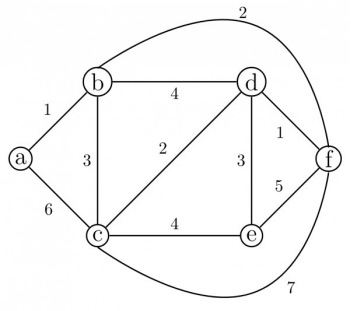
A. $n-1$ B. $2n-2$ C. $\binom{n}{2}$ D. n^2

gate2006 algorithms spanning-tree normal

1.19.12 Spanning Tree: GATE2006-47

https://gateoverflow.in/1823

Consider the following graph:



Which one of the following cannot be the sequence of edges added, **in that order**, to a minimum spanning tree using Kruskal's algorithm?

A. $(a-b),(d-f),(b-f),(d-c),(d-e)$
B. $(a-b),(d-f),(d-c),(b-f),(d-e)$
C. $(d-f),(a-b),(d-c),(b-f),(d-e)$
D. $(d-f),(a-b),(b-f),(d-e),(d-c)$

gate2006 algorithms graph-algorithms spanning-tree normal

1.19.13 Spanning Tree: GATE2007-49

https://gateoverflow.in/1247

Let w be the minimum weight among all edge weights in an undirected connected graph. Let e be a specific edge of weight w. Which of the following is FALSE?

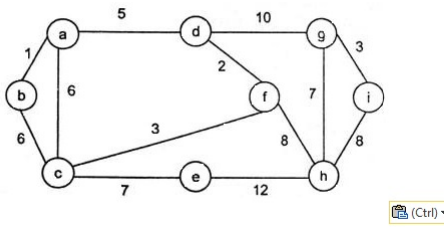
A. There is a minimum spanning tree containing e
B. If e is not in a minimum spanning tree T, then in the cycle formed by adding e to T, all edges have the same weight.
C. Every minimum spanning tree has an edge of weight w
D. e is present in every minimum spanning tree

gate2007 algorithms spanning-tree normal

1.19.14 Spanning Tree: GATE2008-IT-45

https://gateoverflow.in/3355

For the undirected, weighted graph given below, which of the following sequences of edges represents a correct execution of Prim's algorithm to construct a Minimum Spanning Tree?



A. (a, b), (d, f), (f, c), (g, i), (d, a), (g, h), (c, e), (f, h)

© Copyright GATE Overflow. All rights reserved.

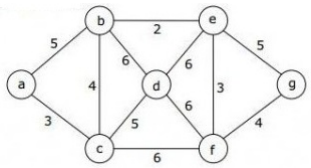
B. $(c, e), (c, f), (f, d), (d, a), (a, b), (g, h), (h, f), (g, i)$
C. $(d, f), (f, c), (d, a), (a, b), (c, e), (f, h), (g, h), (g, i)$
D. $(h, g), (g, i), (h, f), (f, c), (f, d), (d, a), (a, b), (c, e)$

gate2008-it algorithms graph-algorithms spanning-tree normal

1.19.15 Spanning Tree: GATE2009-38

https://gateoverflow.in/1324

Consider the following graph:



Which one of the following is NOT the sequence of edges added to the minimum spanning tree using Kruskal's algorithm?

A. $(b, e)\ (e, f)\ (a, c)\ (b, c)\ (f, g)\ (c, d)$
B. $(b, e)\ (e, f)\ (a, c)\ (f, g)\ (b, c)\ (c, d)$
C. $(b, e)\ (a, c)\ (e, f)\ (b, c)\ (f, g)\ (c, d)$
D. $(b, e)\ (e, f)\ (b, c)\ (a, c)\ (f, g)\ (c, d)$

gate2009 algorithms spanning-tree normal

1.19.16 Spanning Tree: GATE2010-50

https://gateoverflow.in/2355

Consider a complete undirected graph with vertex set $\{0,1,2,3,4\}$. Entry W_{ij} in the matrix W below is the weight of the edge $\{i,j\}$

$$W = \begin{pmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{pmatrix}$$

What is the minimum possible weight of a spanning tree T in this graph such that vertex 0 is a leaf node in the tree T?

A. 7 B. 8 C. 9 D. 10

gate2010 algorithms spanning-tree normal

1.19.17 Spanning Tree: GATE2010-51

https://gateoverflow.in/43328

Consider a complete undirected graph with vertex set $\{0,1,2,3,4\}$. Entry W_{ij} in the matrix W below is the weight of the edge $\{i,j\}$

$$W = \begin{pmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{pmatrix}$$

What is the minimum possible weight of a path P from vertex 1 to vertex 2 in this graph such that P contains at most 3 edges?

A. 7 B. 8 C. 9 D. 10

gate2010 normal algorithms spanning-tree

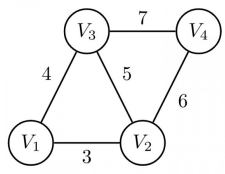
1.19.18 Spanning Tree: GATE2011-54

https://gateoverflow.in/2162

An undirected graph $G(V, E)$ contains $n\ (n > 2)$ nodes named $v_1, v_2, \ldots, v_n$. Two nodes v_i, v_j are connected if and

© Copyright GATE Overflow. All rights reserved.

only if $0 < | i-j | \leq 2$. Each edge (v_i, v_j) is assigned a weight $i+j$. A sample graph with $n=4$ is shown below.



What will be the cost of the minimum spanning tree (MST) of such a graph with n nodes?

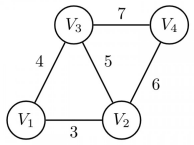
A. $\frac{1}{12}(11n^2-5n)$ B. n^2-n+1 C. $6n-11$ D. $2n+1$

gate2011 algorithms graph-algorithms spanning-tree normal

1.19.19 Spanning Tree: GATE2011-55

https://gateoverflow.in/43325

An undirected graph $G(V,E)$ contains $n\,(n>2)$ nodes named $v_1, v_2, \ldots, v_n$. Two nodes v_i, v_j are connected if and only if $0 < | i-j | \leq 2$. Each edge (v_i, v_j) is assigned a weight $i+j$. A sample graph with $n=4$ is shown below.



The length of the path from v_5 to v_6 in the MST of previous question with $n=10$ is

A. 11 B. 25 C. 31 D. 41

gate2011 algorithms graph-algorithms spanning-tree normal

1.19.20 Spanning Tree: GATE2012-29

https://gateoverflow.in/786

Let G be a weighted graph with edge weights greater than one and G' be the graph constructed by squaring the weights of edges in G. Let T and T' be the minimum spanning trees of G and G', respectively, with total weights t and t'. Which of the following statements is **TRUE**?

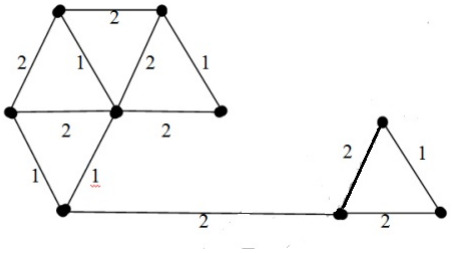
A. $T' = T$ with total weight $t' = t^2$
B. $T' = T$ with total weight $t' < t^2$
C. $T' \neq T$ but total weight $t' = t^2$
D. None of the above

gate2012 algorithms spanning-tree normal marks-to-all

1.19.21 Spanning Tree: GATE2014-2-52

https://gateoverflow.in/2019

The number of distinct minimum spanning trees for the weighted graph below is _____



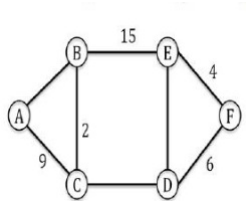
gate2014-2 algorithms spanning-tree numerical-answers normal

1.19.22 Spanning Tree: GATE2015-1-43

https://gateoverflow.in/8313

The graph shown below has 8 edges with distinct integer edge weights. The minimum spanning tree (**MST**) is of weight 36 and contains the edges: $\{(A,C),(B,C),(B,E),(E,F),(D,F)\}$. The edge weights of only those edges which are in the **MST** are given in the figure shown below. The minimum possible sum of weights of all 8 edges of this graph is_____.

© Copyright GATE Overflow. All rights reserved.



gate2015-1 algorithms spanning-tree normal numerical-answers

1.19.23 Spanning Tree: GATE2015-3-40

https://gateoverflow.in/8499

Let G be a connected undirected graph of 100 vertices and 300 edges. The weight of a minimum spanning tree of G is 500. When the weight of each edge of G is increased by five, the weight of a minimum spanning tree becomes _____.

gate2015-3 algorithms spanning-tree easy numerical-answers

1.19.24 Spanning Tree: GATE2016-1-14

https://gateoverflow.in/39673

Let G be a weighted connected undirected graph with distinct positive edge weights. If every edge weight is increased by the same value, then which of the following statements is/are TRUE?

- P: Minimum spanning tree of G does not change.
- Q: Shortest path between any pair of vertices does not change.

A. P only B. Q only C. Neither P nor Q D. Both P and Q

gate2016-1 algorithms spanning-tree normal

1.19.25 Spanning Tree: GATE2016-1-39

https://gateoverflow.in/39725

Let G be a complete undirected graph on 4 vertices, having 6 edges with weights being $1,2,3,4,5,$ and 6. The maximum possible weight that a minimum weight spanning tree of G can have is _____

gate2016-1 algorithms spanning-tree normal numerical-answers

1.19.26 Spanning Tree: GATE2016-1-40

https://gateoverflow.in/39727

$G=(V,E)$ is an undirected simple graph in which each edge has a distinct weight, and e is a particular edge of G. Which of the following statements about the minimum spanning trees $(MSTs)$ of G is/are TRUE?

I. If e is the lightest edge of some cycle in G, then every MST of G includes e.
II. If e is the heaviest edge of some cycle in G, then every MST of G excludes e.

1. Every MST includes lightest weight always.

2. Some of the MST includes or some of the MST excludes the heaviest weight.

A. I only. B. II only. C. Both I and II. D. Neither I nor II.

gate2016-1 algorithms spanning-tree normal

1.19.27 Spanning Tree: TIFR2011-B-35

https://gateoverflow.in/20842

Let G be a connected simple graph (no self-loops or parallel edges) on $n \geq 3$ vertices, with distinct edge weights. Let $e_1, e_2, \ldots, e_m$ be an ordering of the edges in decreasing order of weight. Which of the following statements is FALSE?

A. The edge e_1 has to be present in every maximum weight spanning tree.
B. Both e_1 and e_2 have to be present in every maximum weight spanning tree.
C. The edge e_m has to be present in every minimum weight spanning tree.
D. The edge e_m is never present in any maximum weight spanning tree.
E. G has a unique maximum weight spanning tree.

tifr2011 algorithms graph-algorithms spanning-tree

1.19.28 Spanning Tree: TIFR2013-B-17

https://gateoverflow.in/25860

In a connected weighted graph with n vertices, all the edges have distinct positive integer weights. Then, the maximum number of minimum weight spanning trees in the graph is

a. 1 b. n

© Copyright GATE Overflow. All rights reserved.

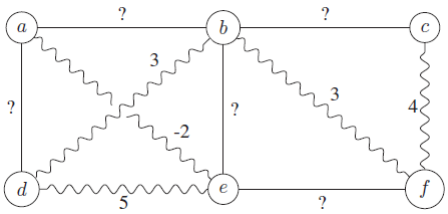
c. equal to number of edges in the graph.
d. equal to maximum weight of an edge of the graph.
e. n^{n-2}

tifr2013 spanning-tree

1.19.29 Spanning Tree: TIFR2014-B-4

https://gateoverflow.in/27174

Consider the following undirected graph with some edge costs missing.



Suppose the wavy edges form a Minimum Cost Spanning Tree for G. Then, which of the following inequalities NEED NOT hold?

a. $\text{cost}(a,b) \geq 6$.
b. $\text{cost}(b,e) \geq 5$.
c. $\text{cost}(e,f) \geq 5$.
d. $\text{cost}(a,d) \geq 4$.
e. $\text{cost}(b,c) \geq 4$.

tifr2014 algorithms graph-algorithms spanning-tree

1.19.30 Spanning Tree: TIFR2014-B-5

https://gateoverflow.in/27180

Let $G = (V,E)$ be an undirected connected simple (i.e., no parallel edges or self-loops) graph with the weight function $w : E \to \mathbb{R}$ on its edge set. Let $w(e_1) < w(e_2) < \cdots < w(e_m)$, where $E = \{e_1, e_2, \ldots, e_m\}$. Suppose T is a minimum spanning tree of G. Which of the following statements is FALSE?

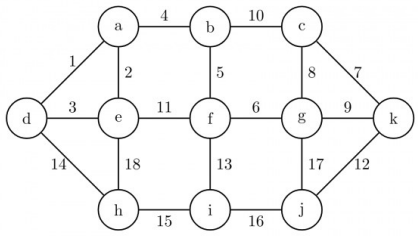
A. The tree T has to contain the edge e_1.
B. The tree T has to contain the edge e_2.
C. The minimum weight edge incident on each vertex has to be present in T.
D. T is the unique minimum spanning tree in G.
E. If we replace each edge weight $w_i = w(e_i)$ by its square w_i^2 , then T must still be a minimum spanning tree of this new instance.

tifr2014 algorithms spanning-tree

1.19.31 Spanning Tree: TIFR2015-B-2

https://gateoverflow.in/29844

Consider the following undirected connected graph G with weights on its edges as given in the figure below. A minimum spanning tree is a spanning tree of least weight and a maximum spanning tree is one with largest weight. A second best minimum spanning tree whose weight is the smallest among all spanning trees that are not minimum spanning trees in G.



Which of the following statements is TRUE in the above graph? (Note that all the edge weights are distinct in the above graph)

A. There is more than one minimum spanning tree and similarly, there is more than one maximum spanning tree here.
B. There is a unique minimum spanning tree, however there is more than one maximum spanning tree here.
C. There is more than one minimum spanning tree, however there is a unique maximum spanning tree here.
D. There is more than one minimum spanning tree and similarly, there is more than one second-best minimum spanning tree here.
E. There is unique minimum spanning tree, however there is more than one second-best minimum spanning tree here.

© Copyright GATE Overflow. All rights reserved.

tifr2015 spanning-tree algorithms graph-algorithms

1.20 Time Complexity (33)

1.20.1 Time Complexity: GATE1988-6i

https://gateoverflow.in/94363

Given below is the sketch of a program that represents the path in a two-person game tree by the sequence of active procedure calls at any time. The program assumes that the payoffs are real number in a limited range; that the constant INF is larger than any positive payoff and its negation is smaller than any negative payoff and that there is a function “payoff” and that computes the payoff for any board that is a leaf. The type “boardtype” has been suitably declared to represent board positions. It is player-1’s move if mode = MAX and player-2’s move if mode=MIN. The type modetype =(MAX, MIN). The functions “min” and “max” find the minimum and maximum of two real numbers.

```
function search(B: boardtype; mode: modetype): real;
    var
        C:boardtype; {a child of board B}
        value:real;
    begin
        if B is a leaf then
            return (payoff(B))
        else
            begin
            if mode = MAX then value :=-INF
            else
                value:INF;
            for each child C of board B do
            if mode = MAX then
                value:=max (value, search (C, MIN))
            else
                value:=min(value, search(C, MAX))
            return(value)
            end
    end; (search)
```

Comment on the working principle of the above program. Suggest a possible mechanism for reducing the amount of search.

gate1988 normal descriptive algorithms time-complexity

1.20.2 Time Complexity: GATE1989-2-iii

https://gateoverflow.in/87080

Match the pairs in the following:

(A)	$O(\log n)$	(p)	Heapsort
(B)	$O(n)$	(q)	Depth-first search
(C)	$O(n \log n)$	(r)	Binary search
(D)	$O(n^2)$	(s)	Selection of the k^{th} smallest element in a set of n elements

gate1989 match-the-following algorithms time-complexity

1.20.3 Time Complexity: GATE1993-8.7

https://gateoverflow.in/2305

$\sum_{1 \leq k \leq n} O(n)$, where $O(n)$ stands for order n is:

A. $O(n)$ B. $O(n^2)$ C. $O(n^3)$ D. $O(3n^2)$ E. $O(1.5n^2)$

gate1993 algorithms time-complexity easy

1.20.4 Time Complexity: GATE1999-1.13

https://gateoverflow.in/1466

Suppose we want to arrange the n numbers stored in any array such that all negative values occur before all positive ones. Minimum number of exchanges required in the worst case is

A. $n-1$ B. n C. $n+1$ D. None of the above

gate1999 algorithms time-complexity normal

© Copyright GATE Overflow. All rights reserved.

3 Programming and DS: DS (212)

Arrays, Stacks, Queues, Linked lists, Trees, Binary search trees, Binary heaps, Graphs.

Mark Distribution in Previous GATE

Year	2019	2018	2017-1	2017-2	2016-1	2016-2	Minimum	Average	Maximum
1 Mark Count	0	2	3	1	1	1	0	1.3	3
2 Marks Count	2	0	0	1	3	3	0	1.5	3
Total Marks	4	2	3	3	7	7	**2**	**4.3**	**7**

3.1 Abstract Data Type (1)

3.1.1 Abstract Data Type: GATE2005-2

https://gateoverflow.in/1344

An Abstract Data Type (ADT) is:

A. same as an abstract class
B. a data type that cannot be instantiated
C. a data type for which only the operations defined on it can be used, but none else
D. all of the above

gate2005 data-structure normal abstract-data-type

3.2 Arrays (13)

3.2.1 Arrays: GATE1993-12

https://gateoverflow.in/2309

The following Pascal program segments finds the largest number in a two-dimensional integer array $A[0\ldots n-1, 0\ldots n-1]$ using a single loop. Fill up the boxes to complete the program and write against $\boxed{A}, \boxed{B}, \boxed{C}$ and $\boxed{D}$ in your answer book Assume that max is a variable to store the largest value and i, j are the indices to the array.

```
begin
    max:=|A|, i:=0, j:=0;
    while |B| do
    begin
        if A[i, j]>max then max:=A[i, j];
        if |C| then j:=j+1;
        else begin
            j:=0;
            i:=|D|
        end
    end
end
```

gate1993 data-structure arrays normal

3.2.2 Arrays: GATE1994-1.11

https://gateoverflow.in/2452

In a compact single dimensional array representation for lower triangular matrices (i.e all the elements above the diagonal are zero) of size $n \times n$, non-zero elements, (i.e elements of lower triangle) of each row are stored one after another, starting from the first row, the index of the $(i,j)^{th}$ element of the lower triangular matrix in this new representation is:

A. $i+j$ B. $i+j-1$ C. $(j-1)+\frac{i(i-1)}{2}$ D. $i+\frac{j(j-1)}{2}$

gate1994 data-structure arrays normal

3.2.3 Arrays: GATE1994-25

https://gateoverflow.in/2521

An array A contains n integers in non-decreasing order, $A[1] \leq A[2] \leq \cdots \leq A[n]$. Describe, using Pascal like pseudo code, a linear time algorithm to find i, j, such that $A[i] + A[j] = a$ given integer M, if such i, j exist.

gate1994 data-structure arrays normal

3.2.4 Arrays: GATE1997-17

https://gateoverflow.in/2277

An array A contains $n \geq 1$ positive integers in the locations $A[1], A[2], \ldots A[n]$. The following program fragment

© Copyright GATE Overflow. All rights reserved.

prints the length of a shortest sequence of consecutive elements of A, $A[i], A[i+1], \ldots, A[j]$ such that the sum of their values is $\geq M$, a given positive number. It prints '$n+1$' if no such sequence exists. Complete the program by filling in the boxes. In each case use the simplest possible expression. Write only the line number and the contents of the box.

```
begin
i:=1;j:=1;
sum := □
min:=n; finish:=false;
while not finish do
    if □ then
        if j=n then finish:=true
        else
        begin
            j:=j+1;
            sum:= □
        end
    else
    begin
        if(j-i) < min then min:=j-i;
        sum:=sum -A[i];
        i:=i+1;
    end
    writeln (min +1);
end.
```

gate1997 data-structure arrays normal

3.2.5 Arrays: GATE1998-2.14

https://gateoverflow.in/1686

Let A be a two dimensional array declared as follows:

```
A: array [1 …. 10] [1 ….. 15] of integer;
```

Assuming that each integer takes one memory location, the array is stored in row-major order and the first element of the array is stored at location 100, what is the address of the element $A[i][j]$?

A. $15i+j+84$ B. $15j+i+84$ C. $10i+j+89$ D. $10j+i+89$

gate1998 data-structure arrays easy

3.2.6 Arrays: GATE2000-1.2

https://gateoverflow.in/625

An $n \times n$ array v is defined as follows:

$v[i,j] = i-j$ for all $i,j, i \leq n, 1 \leq j \leq n$

The sum of the elements of the array v is

A. 0 B. $n-1$ C. n^2-3n+2 D. $n^2\frac{(n+1)}{2}$

gate2000 data-structure arrays easy

3.2.7 Arrays: GATE2000-15

https://gateoverflow.in/686

Suppose you are given arrays $p[1.....N]$ and $q[1.....N]$ both uninitialized, that is, each location may contain an arbitrary value), and a variable count, initialized to 0. Consider the following procedures set and is_set:

```
set(i) {
    count = count + 1;
    q[count] = i;
    p[i] = count;
}
is_set(i) {
    if (p[i] ≤ 0 or p[i] > count)
        return false;
    if (q[p[i]] ≠ i)
        return false;
    return true;
}
```

A. Suppose we make the following sequence of calls:
$set(7); set(3); set(9);$
After these sequence of calls, what is the value of count, and what do $q[1], q[2], q[3], p[7], p[3]$ and $p[9]$ contain?

B. Complete the following statement "The first count elements of _____contain values i such that set (_____) has been called".

© Copyright GATE Overflow. All rights reserved.

C. Show that if $set(i)$ has not been called for some i, then regardless of what $p[i]$ contains, $is\_set(i)$ will return false.

gate2000 data-structure arrays easy descriptive

3.2.8 Arrays: GATE2005-5

https://gateoverflow.in/1347

A program P reads in 500 integers in the range $[0,100]$ representing the scores of 500 students. It then prints the frequency of each score above 50. What would be the best way for P to store the frequencies?

A. An array of 50 numbers
B. An array of 100 numbers
C. An array of 500 numbers
D. A dynamically allocated array of 550 numbers

gate2005 data-structure arrays easy

3.2.9 Arrays: GATE2013-50

https://gateoverflow.in/1557

The procedure given below is required to find and replace certain characters inside an input character string supplied in array A. The characters to be replaced are supplied in array $oldc$, while their respective replacement characters are supplied in array $newc$. Array A has a fixed length of five characters, while arrays $oldc$ and $newc$ contain three characters each. However, the procedure is flawed.

```
void find_and_replace (char *A, char *oldc, char *newc) {
    for (int i=0; i<5; i++)
        for (int j=0; j<3; j++)
            if (A[i] == oldc[j])
                A[i] = newc[j];
}
```

The procedure is tested with the following four test cases.

1. $oldc =$ "abc", $newc =$ "dab"
2. $oldc =$ "cde", $newc =$ "bcd"
3. $oldc =$ "bca", $newc =$ "cda"
4. $oldc =$ "abc", $newc =$ "bac"

The tester now tests the program on all input strings of length five consisting of characters 'a', 'b', 'c', 'd' and 'e' with duplicates allowed. If the tester carries out this testing with the four test cases given above, how many test cases will be able to capture the flaw?

A. Only one B. Only two C. Only three D. All four

gate2013 data-structure arrays normal

3.2.10 Arrays: GATE2013-51

https://gateoverflow.in/43291

The procedure given below is required to find and replace certain characters inside an input character string supplied in array A. The characters to be replaced are supplied in array $oldc$, while their respective replacement characters are supplied in array $newc$. Array A has a fixed length of five characters, while arrays $oldc$ and $newc$ contain three characters each. However, the procedure is flawed.

```
void find_and_replace (char *A, char *oldc, char *newc) {
    for (int i=0; i<5; i++)
        for (int j=0; j<3; j++)
            if (A[i] == oldc[j])
                A[i] = newc[j];
}
```

The procedure is tested with the following four test cases.

1. $oldc =$ "abc", $newc =$ "dab"
2. $oldc =$ "cde", $newc =$ "bcd"
3. $oldc =$ "bca", $newc =$ "cda"
4. $oldc =$ "abc", $newc =$ "bac"

If array A is made to hold the string "$abcde$", which of the above four test cases will be successful in exposing the flaw in this procedure?

A. None B. 2 only C. 3 and 4 only D. 4 only

© Copyright GATE Overflow. All rights reserved.

gate2013 data-structure arrays normal

3.2.11 Arrays: GATE2014-3-42

https://gateoverflow.in/2076

Consider the C function given below. Assume that the array $listA$ contains $n(>0)$ elements, sorted in ascending order.

```
int ProcessArray(int *listA, int x, int n)
{
        int i, j, k;
        i = 0;      j = n-1;
         do {
               k = (i+j)/2;
               if (x <= listA[k]) j = k-1;
               if (listA[k] <= x) i = k+1;
             }
        while (i <= j);
        if (listA[k] == x) return(k);
        else    return -1;
}
```

Which one of the following statements about the function $ProcessArray$ is **CORRECT**?

A. It will run into an infinite loop when x is not in $listA$.
B. It is an implementation of binary search.
C. It will always find the maximum element in $listA$.
D. It will return -1 even when x is present in $listA$.

gate2014-3 data-structure arrays easy

3.2.12 Arrays: GATE2015-2-31

https://gateoverflow.in/8148

A Young tableau is a $2D$ array of integers increasing from left to right and from top to bottom. Any unfilled entries are marked with ∞, and hence there cannot be any entry to the right of, or below a ∞. The following Young tableau consists of unique entries.

1	2	5	14
3	4	6	23
10	12	18	25
31	∞	∞	∞

When an element is removed from a Young tableau, other elements should be moved into its place so that the resulting table is still a Young tableau (unfilled entries may be filled with a ∞). The minimum number of entries (other than 1) to be shifted, to remove 1 from the given Young tableau is _____.

gate2015-2 databases arrays normal numerical-answers

3.2.13 Arrays: TIFR2011-B-30

https://gateoverflow.in/20611

Consider an array $A[1...n]$. It consists of a permutation of numbers $1....n$. Now compute another array $B[1...n]$ as follows: $B[A[i]] := i$ for all i. Which of the following is true?

A. B will be a sorted array.
B. B is a permutation of array A.
C. Doing the same transformation twice will not give the same array.
D. B is not a permutation of array A.
E. None of the above.

tifr2011 data-structure arrays

3.3 Binary Search Tree (29)

3.3.1 Binary Search Tree: GATE1996-2.14

https://gateoverflow.in/2743

A binary search tree is generated by inserting in order the following integers:

$$50, 15, 62, 5, 20, 58, 91, 3, 8, 37, 60, 24$$

The number of nodes in the left subtree and right subtree of the root respectively is

© Copyright GATE Overflow. All rights reserved.

A. $(4,7)$ B. $(7,4)$ C. $(8,3)$ D. $(3,8)$

gate1996 data-structure binary-search-tree normal

3.3.2 Binary Search Tree: GATE1996-4

https://gateoverflow.in/2756

A binary search tree is used to locate the number 43. Which of the following probe sequences are possible and which are not? Explain.

(a) $61\ 52\ 14\ 17\ 40\ 43$
(b) $2\ 3\ 50\ 40\ 60\ 43$
(c) $10\ 65\ 31\ 48\ 37\ 43$
(d) $81\ 61\ 52\ 14\ 41\ 43$
(e) $17\ 77\ 27\ 66\ 18\ 43$

gate1996 data-structure binary-search-tree normal

3.3.3 Binary Search Tree: GATE2001-14

https://gateoverflow.in/755

A. Insert the following keys one by one into a binary search tree in the order specified.

$$15,32,20,9,3,25,12,1$$

Show the final binary search tree after the insertions.
B. Draw the binary search tree after deleting 15 from it.
C. Complete the statements $S1$, $S2$ and $S3$ in the following function so that the function computes the depth of a binary tree rooted at t.

```
typedef struct tnode{
    int key;
    struct tnode *left, *right;
} *Tree;

int depth (Tree t)
{
    int x, y;
    if (t == NULL) return 0;
    x = depth (t -> left);
S1:     ____________;

S2:     if (x > y) return ___________;

S3:     else return _______;

}
```

gate2001 data-structure binary-search-tree normal descriptive

3.3.4 Binary Search Tree: GATE2003-19, ISRO2009-24

https://gateoverflow.in/909

Suppose the numbers $7,5,1,8,3,6,0,9,4,2$ are inserted in that order into an initially empty binary search tree. The binary search tree uses the usual ordering on natural numbers. What is the in-order traversal sequence of the resultant tree?

A. $7\ 5\ 1\ 0\ 3\ 2\ 4\ 6\ 8\ 9$
B. $0\ 2\ 4\ 3\ 1\ 6\ 5\ 9\ 8\ 7$
C. $0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9$
D. $9\ 8\ 6\ 4\ 2\ 3\ 0\ 1\ 5\ 7$

gate2003 binary-search-tree easy isro2009

3.3.5 Binary Search Tree: GATE2003-6

https://gateoverflow.in/897

Let $T(n)$ be the number of different binary search trees on n distinct elements.
Then $T(n) = \sum_{k=1}^{n} T(k-1)T(x)$, where x is

© Copyright GATE Overflow. All rights reserved.

A. $n-k+1$ B. $n-k$ C. $n-k-1$ D. $n-k-2$

gate2003 normal binary-search-tree

3.3.6 Binary Search Tree: GATE2003-63, ISRO2009-25

https://gateoverflow.in/950

A data structure is required for storing a set of integers such that each of the following operations can be done in $O(\log n)$ time, where n is the number of elements in the set.

I. Deletion of the smallest element

II. Insertion of an element if it is not already present in the set

Which of the following data structures can be used for this purpose?

A. A heap can be used but not a balanced binary search tree so all operations done in O(log n)

B. A balanced binary search tree can be used but not a heap

C. Both balanced binary search tree and heap can be used

D. Neither balanced search tree nor heap can be used

gate2003 data-structure easy isro2009 binary-search-tree

3.3.7 Binary Search Tree: GATE2004-4, ISRO2009-26

https://gateoverflow.in/1001

The following numbers are inserted into an empty binary search tree in the given order: $10,1,3,5,15,12,16$. What is the height of the binary search tree (the height is the maximum distance of a leaf node from the root)?

A. 2 B. 3 C. 4 D. 6

gate2004 data-structure binary-search-tree easy isro2009

3.3.8 Binary Search Tree: GATE2004-85

https://gateoverflow.in/1079

A program takes as input a balanced binary search tree with n leaf nodes and computes the value of a function $g(x)$ for each node x. If the cost of computing $g(x)$ is:

$$\min\left(\begin{array}{c}\text{number of leaf-nodes}\\ \text{in left-subtree of } x\end{array}, \begin{array}{c}\text{number of leaf-nodes}\\ \text{in right-subtree of } x\end{array}\right)$$

Then the worst-case time complexity of the program is?

A. $\Theta(n)$ B. $\Theta(n \log n)$

C. $\Theta(n^2)$ D. $\Theta(n^2 \log n)$

gate2004 binary-search-tree normal data-structure

3.3.9 Binary Search Tree: GATE2005-IT-12

https://gateoverflow.in/3757

The numbers $1,2,\ldots.n$ are inserted in a binary search tree in some order. In the resulting tree, the right subtree of the root contains p nodes. The first number to be inserted in the tree must be

A. p B. $p+1$ C. $n-p$ D. $n-p+1$

gate2005-it data-structure normal binary-search-tree

3.3.10 Binary Search Tree: GATE2005-IT-55

https://gateoverflow.in/3816

A binary search tree contains the numbers $1,2,3,4,5,6,7,8$. When the tree is traversed in pre-order and the values in each node printed out, the sequence of values obtained is $5,3,1,2,4,6,8,7$. If the tree is traversed in post-order, the sequence obtained would be

A. $8,7,6,5,4,3,2,1$ B. $1,2,3,4,8,7,6,5$

C. $2,1,4,3,6,7,8,5$ D. $2,1,4,3,7,8,6,5$

gate2005-it data-structure binary-search-tree normal

© Copyright GATE Overflow. All rights reserved.

3.3.11 Binary Search Tree: GATE2006-IT-45

https://gateoverflow.in/3588

Suppose that we have numbers between 1 and 100 in a binary search tree and want to search for the number 55. Which of the following sequences CANNOT be the sequence of nodes examined?

A. $\{10, 75, 64, 43, 60, 57, 55\}$
B. $\{90, 12, 68, 34, 62, 45, 55\}$
C. $\{9, 85, 47, 68, 43, 57, 55\}$
D. $\{79, 14, 72, 56, 16, 53, 55\}$

gate2006-it data-structure binary-search-tree normal

3.3.12 Binary Search Tree: GATE2007-IT-29

https://gateoverflow.in/3462

When searching for the key value 60 in a binary search tree, nodes containing the key values $10, 20, 40, 50, 70, 80, 90$ are traversed, not necessarily in the order given. How many different orders are possible in which these key values can occur on the search path from the root to the node containing the value 60?

A. 35 B. 64 C. 128 D. 5040

7! / 3!4!

gate2007-it data-structure binary-search-tree normal

3.3.13 Binary Search Tree: GATE2008-46

https://gateoverflow.in/458

You are given the postorder traversal, P, of a binary search tree on the n elements $1, 2, \ldots, n$. You have to determine the unique binary search tree that has P as its postorder traversal. What is the time complexity of the most efficient algorithm for doing this?

A. $\Theta(\log n)$
B. $\Theta(n)$
C. $\Theta(n \log n)$
D. None of the above, as the tree cannot be uniquely determined

gate2008 data-structure binary-search-tree normal

3.3.14 Binary Search Tree: GATE2008-IT-12

https://gateoverflow.in/3272

Which of the following is TRUE?

A. The cost of searching an AVL tree is $\Theta(\log n)$ but that of a binary search tree is $O(n)$
B. The cost of searching an AVL tree is $\Theta(\log n)$ but that of a complete binary tree is $\Theta(n \log n)$
C. The cost of searching a binary search tree is $O(\log n)$ but that of an AVL tree is $\Theta(n)$
D. The cost of searching an AVL tree is $\Theta(n \log n)$ but that of a binary search tree is $O(n)$

gate2008-it data-structure binary-search-tree easy

3.3.15 Binary Search Tree: GATE2008-IT-71

https://gateoverflow.in/3385

A Binary Search Tree (BST) stores values in the range 37 to 573. Consider the following sequence of keys.

I. $81, 537, 102, 439, 285, 376, 305$
II. $52, 97, 121, 195, 242, 381, 472$
III. $142, 248, 520, 386, 345, 270, 307$
IV. $550, 149, 507, 395, 463, 402, 270$

Suppose the BST has been unsuccessfully searched for key 273. Which all of the above sequences list nodes in the order in which we could have encountered them in the search?

A. II and III only B. I and III only C. III and IV only D. III only

gate2008-it data-structure binary-search-tree normal

3.3.16 Binary Search Tree: GATE2008-IT-72

https://gateoverflow.in/3386

A Binary Search Tree (BST) stores values in the range 37 to 573. Consider the following sequence of keys.

I. $81, 537, 102, 439, 285, 376, 305$
II. $52, 97, 121, 195, 242, 381, 472$
III. $142, 248, 520, 386, 345, 270, 307$
IV. $550, 149, 507, 395, 463, 402, 270$

Which of the following statements is TRUE?

© Copyright GATE Overflow. All rights reserved.

A. I, II and IV are inorder sequences of three different BSTs
B. I is a preorder sequence of some BST with 439 as the root
C. II is an inorder sequence of some BST where 121 is the root and 52 is a leaf
D. IV is a postorder sequence of some BST with 149 as the root

gate2008-it data-structure binary-search-tree easy

3.3.17 Binary Search Tree: GATE2008-IT-73 https://gateoverflow.in/3387

How many distinct BSTs can be constructed with 3 distinct keys?

A. 4 B. 5 C. 6 D. 9

gate2008-it data-structure binary-search-tree normal

3.3.18 Binary Search Tree: GATE2009-37,ISRO-DEC2017-55 https://gateoverflow.in/1323

What is the maximum height of any AVL-tree with 7 nodes? Assume that the height of a tree with a single node is 0.

A. 2 B. 3 C. 4 D. 5

gate2009 data-structure binary-search-tree normal isrodec2017

3.3.19 Binary Search Tree: GATE2012-5 https://gateoverflow.in/37

The worst case running time to search for an element in a balanced binary search tree with $n2^n$ elements is

A. $\Theta(n \log n)$
B. $\Theta(n2^n)$
C. $\Theta(n)$
D. $\Theta(\log n)$

gate2012 data-structure normal binary-search-tree

3.3.20 Binary Search Tree: GATE2013-43 https://gateoverflow.in/1554

The preorder traversal sequence of a binary search tree is $30,20,10,15,25,23,39,35,42$. Which one of the following is the postorder traversal sequence of the same tree?

A. $10,20,15,23,25,35,42,39,30$
B. $15,10,25,23,20,42,35,39,30$
C. $15,20,10,23,25,42,35,39,30$
D. $15,10,23,25,20,35,42,39,30$

gate2013 data-structure binary-search-tree normal

3.3.21 Binary Search Tree: GATE2013-7 https://gateoverflow.in/1416

Which one of the following is the tightest upper bound that represents the time complexity of inserting an object into a binary search tree of n nodes?

A. $O(1)$ B. $O(\log n)$ C. $O(n)$ D. $O(n \log n)$

gate2013 data-structure easy binary-search-tree

3.3.22 Binary Search Tree: GATE2014-3-39 https://gateoverflow.in/2073

Suppose we have a balanced binary search tree T holding n numbers. We are given two numbers L and H and wish to sum up all the numbers in T that lie between L and H. Suppose there are m such numbers in T. If the tightest upper bound on the time to compute the sum is $O(n^a \log^b n + m^c \log^d n)$, the value of $a + 10b + 100c + 1000d$ is _____.

gate2014-3 data-structure binary-search-tree numerical-answers normal O(log n+ m) = 110

3.3.23 Binary Search Tree: GATE2015-1-10 https://gateoverflow.in/8129

Which of the following is/are correct in order traversal sequence(s) of binary search tree(s)?

I. $3,5,7,8,15,19,25$
II. $5,8,9,12,10,15,25$
III. $2,7,10,8,14,16,20$
IV. $4,6,7,9,18,20,25$

A. I and IV only B. II and III only C. II and IV only D. II only

© Copyright GATE Overflow. All rights reserved.

gate2015-1 data-structure binary-search-tree easy

3.3.24 Binary Search Tree: GATE2015-1-23

https://gateoverflow.in/8221

What are the worst-case complexities of insertion and deletion of a key in a binary search tree?

A. $\Theta(\log n)$ for both insertion and deletion
B. $\Theta(n)$ for both insertion and deletion
C. $\Theta(n)$ for insertion and $\Theta(\log n)$ for deletion
D. $\Theta(\log n)$ for insertion and $\Theta(n)$ for deletion

gate2015-1 data-structure binary-search-tree easy

3.3.25 Binary Search Tree: GATE2015-3-13

https://gateoverflow.in/8409

While inserting the elements $71, 65, 84, 69, 67, 83$ in an empty binary search tree (BST) in the sequence shown, the element in the lowest level is

A. 65 B. 67 C. 69 D. 83

gate2015-3 data-structure binary-search-tree easy

3.3.26 Binary Search Tree: GATE2016-2-40

https://gateoverflow.in/39586

The number of ways in which the numbers $1, 2, 3, 4, 5, 6, 7$ can be inserted in an empty binary search tree, such that the resulting tree has height 6, is _____.

Note: The height of a tree with a single node is 0.

at each level we have exactly 2 possible options like 1 and 7 for root- one corresponding to making it left skewed and other right skewed. And this is the same for all levels up to 6 giving 2^{6}=64 possible ways.

gate2016-2 data-structure binary-search-tree normal numerical-answers

3.3.27 Binary Search Tree: GATE2017-1-6

https://gateoverflow.in/118286

Let T be a binary search tree with 15 nodes. The minimum and maximum possible heights of T are:

Note: The height of a tree with a single node is 0.

A. 4 and 15 respectively.
B. 3 and 14 respectively.
C. 4 and 14 respectively.
D. 3 and 15 respectively.

gate2017-1 data-structure binary-search-tree easy

3.3.28 Binary Search Tree: GATE2017-2-36

https://gateoverflow.in/118378

The pre-order traversal of a binary search tree is given by $12, 8, 6, 2, 7, 9, 10, 16, 15, 19, 17, 20$. Then the post-order traversal of this tree is

A. $2, 6, 7, 8, 9, 10, 12, 15, 16, 17, 19, 20$
B. $2, 7, 6, 10, 9, 8, 15, 17, 20, 19, 16, 12$
C. $7, 2, 6, 8, 9, 10, 20, 17, 19, 15, 16, 12$
D. $7, 6, 2, 10, 9, 8, 15, 16, 17, 20, 19, 12$

gate2017-2 data-structure binary-search-tree

3.3.29 Binary Search Tree: TIFR2010-B-26

https://gateoverflow.in/18749

Suppose there is a balanced binary search tree with n nodes, where at each node, in addition to the key, we store the number of elements in the sub tree rooted at that node.

Now, given two elements a and b, such that $a < b$, we want to find the number of elements x in the tree that lie between a and b, that is, $a \leq x \leq b$. This can be done with (choose the best solution).

A. $O(\log n)$ comparisons and $O(\log n)$ additions.
B. $O(\log n)$ comparisons but no further additions.
C. $O(\sqrt{n})$ comparisons but $O(\log n)$ additions.
D. $O(\log n)$ comparisons but a constant number of additions.
E. $O(n)$ comparisons and $O(n)$ additions, using depth-first- search.

© Copyright GATE Overflow. All rights reserved.

tifr2010 binary-search-tree

3.4 Binary Tree (56)

3.4.1 Binary Tree: GATE1987-2c

https://gateoverflow.in/80579

State whether the following statements are TRUE or FALSE:

It is possible to construct a binary tree uniquely whose pre-order and post-order traversals are given?

gate1987 binary-tree data-structure normal

3.4.2 Binary Tree: GATE1987-2g

https://gateoverflow.in/80588

State whether the following statements are TRUE or FALSE:

If the number of leaves in a tree is not a power of 2, then the tree is not a binary tree.

gate1987 data-structure binary-tree

3.4.3 Binary Tree: GATE1987-7b

https://gateoverflow.in/82427

Construct a binary tree whose preorder traversal is

- K L N M P R Q S T

and inorder traversal is

- N L K P R M S Q T

gate1987 data-structure binary-tree

3.4.4 Binary Tree: GATE1988-7i

https://gateoverflow.in/94366

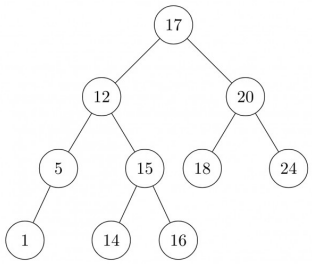
Define the height of a binary tree or subtree and also define a height-balanced (AVL) tree.

gate1988 normal descriptive data-structure binary-tree

3.4.5 Binary Tree: GATE1988-7ii

https://gateoverflow.in/94367

Mark the balance factor of each on the tree given on the below figure and state whether it is height-balanced.



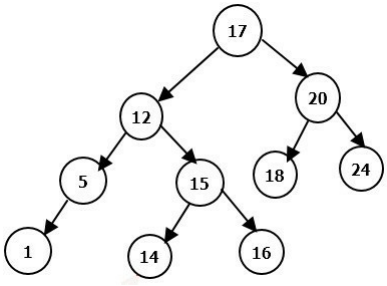
gate1988 normal descriptive binary-tree

3.4.6 Binary Tree: GATE1988-7iii

https://gateoverflow.in/94368

Consider the tree given in the below figure, insert 13 and show the new balance factors that would arise if the tree is not rebalanced. Finally, carry out the required rebalancing of the tree and show the new tree with the balance factors on each mode.

© Copyright GATE Overflow. All rights reserved.



gate1988 normal descriptive data-structure binary-tree

3.4.7 Binary Tree: GATE1990-3-iv

https://gateoverflow.in/84828

Choose the correct alternatives (More than one may be correct).

The total external path length, EPL, of a binary tree with n external nodes is, $EPL = \sum_w Iw$, where I_w is the path length of external node w),

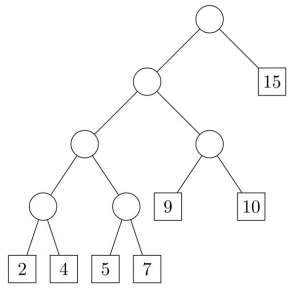
A. $\leq n^2$ always.
B. $\geq n \log_2 n$ always.
C. Equal to n^2 always.
D. $O(n)$ for some special trees.

gate1990 normal data-structure binary-tree

3.4.8 Binary Tree: GATE1991-01,viii

https://gateoverflow.in/506

The weighted external path length of the binary tree in figure is _____

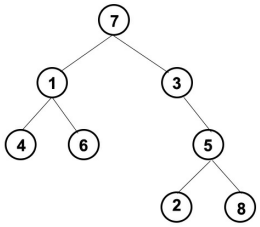


gate1991 binary-tree data-structure normal

3.4.9 Binary Tree: GATE1991-1,ix

https://gateoverflow.in/502

If the binary tree in figure is traversed in inorder, then the order in which the nodes will be visited is _____

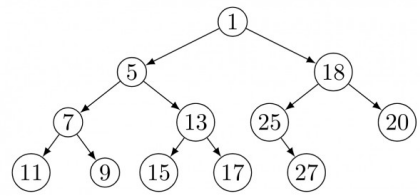


gate1991 binary-tree easy data-structure

3.4.10 Binary Tree: GATE1991-14,a

https://gateoverflow.in/541

Consider the binary tree in the figure below:



(a). What structure is represented by the binary tree?

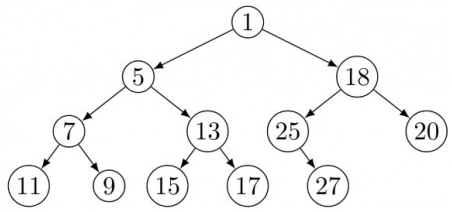
© Copyright GATE Overflow. All rights reserved.

gate1991 data-structure binary-tree time-complexity normal

3.4.11 Binary Tree: GATE1991-14,b

https://gateoverflow.in/43026

Consider the binary tree in the figure below:



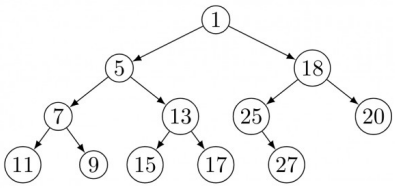
Give different steps for deleting the node with key 5 so that the structure is preserved.

gate1991 data-structure binary-tree normal

3.4.12 Binary Tree: GATE1991-14,c

https://gateoverflow.in/43027

Consider the binary tree in the figure below:



Outline a procedure in Pseudo-code to delete an arbitrary node from such a binary tree with n nodes that preserves the structures. What is the worst-case-time-complexity of your procedure?

gate1991 normal data-structure binary-tree time-complexity

3.4.13 Binary Tree: GATE1993-16

https://gateoverflow.in/2313

Prove by the principal of mathematical induction that for any binary tree, in which every non-leaf node has 2-descendants, the number of leaves in the tree is one more than the number of non-leaf nodes.

gate1993 data-structure binary-tree normal

3.4.14 Binary Tree: GATE1994-8

https://gateoverflow.in/2504

A rooted tree with 12 nodes has its nodes numbered 1 to 12 in pre-order. When the tree is traversed in post-order, the nodes are visited in the order $3,5,4,2,7,8,6,10,11,12,9,1$.

Reconstruct the original tree from this information, that is, find the parent of each node, and show the tree diagrammatically.

gate1994 data-structure binary-tree normal

3.4.15 Binary Tree: GATE1995-1.17

https://gateoverflow.in/2604

A binary tree T has n leaf nodes. The number of nodes of degree 2 in T is

A. $\log_2 n$ B. $n-1$ C. n D. 2^n

gate1995 data-structure binary-tree normal

3.4.16 Binary Tree: GATE1995-6

https://gateoverflow.in/2667

What is the number of binary trees with 3 nodes which when traversed in post-order give the sequence A,B,C? Draw all these binary trees.

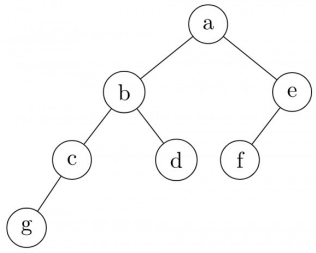
gate1995 data-structure binary-tree normal

© Copyright GATE Overflow. All rights reserved.

3.4.17 Binary Tree: GATE1996-1.14

https://gateoverflow.in/2718

In the balanced binary tree in the below figure, how many nodes will become unbalanced when a node is inserted as a child of the node "g"?



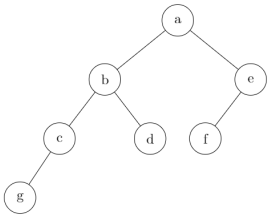
A. 1 B. 3 C. 7 D. 8

gate1996 data-structure binary-tree normal

3.4.18 Binary Tree: GATE1996-1.15

https://gateoverflow.in/2719

Which of the following sequences denotes the post order traversal sequence of the below tree?



A. $f\ e\ g\ c\ d\ b\ a$
B. $g\ c\ b\ d\ a\ f\ e$
C. $g\ c\ d\ b\ f\ e\ a$
D. $f\ e\ d\ g\ c\ b\ a$

gate1996 data-structure binary-tree easy

3.4.19 Binary Tree: GATE1997-16

https://gateoverflow.in/2276

A size-balanced binary tree is a binary tree in which for every node the difference between the number of nodes in the left and right subtree is at most 1. The distance of a node from the root is the length of the path from the root to the node. The height of a binary tree is the maximum distance of a leaf node from the root.

A. Prove, by using induction on h, that a size-balance binary tree of height h contains at least 2^h nodes.
B. In a size-balanced binary tree of height $h \geqslant 1$, how many nodes are at distance $h-1$ from the root? Write only the answer without any explanations.

gate1997 data-structure binary-tree normal

3.4.20 Binary Tree: GATE1997-4.5

https://gateoverflow.in/2246

A binary search tree contains the value $1,2,3,4,5,6,7,8$. The tree is traversed in pre-order and the values are printed out. Which of the following sequences is a valid output?

A. $5\ 3\ 1\ 2\ 4\ 7\ 8\ 6$
B. $5\ 3\ 1\ 2\ 6\ 4\ 8\ 7$
C. $5\ 3\ 2\ 4\ 1\ 6\ 7\ 8$
D. $5\ 3\ 1\ 2\ 4\ 7\ 6\ 8$

gate1997 data-structure binary-tree normal

3.4.21 Binary Tree: GATE1998-20

https://gateoverflow.in/1734

Draw the binary tree with node labels $\mathrm{a, b, c, d, e, f}$ and g for which the inorder and postorder traversals result in the following sequences:

Inorder: $\mathrm{a\ f\ b\ c\ d\ g\ e}$

Postorder: $\mathrm{a\ f\ c\ g\ e\ d\ b}$

gate1998 data-structure binary-tree descriptive

© Copyright GATE Overflow. All rights reserved.

3.4.22 Binary Tree: GATE2000-1.14

https://gateoverflow.in/637

Consider the following nested representation of binary trees: $(X\,Y\,Z)$ indicates Y and Z are the left and right subtrees, respectively, of node X. Note that Y and Z may be $NULL$, or further nested. Which of the following represents a valid binary tree?

A. $(1\,2\,(4\,5\,6\,7))$
B. $(1\,(2\,3\,4)\,5\,6)\,7)$
C. $(1\,(2\,3\,4)\,(5\,6\,7))$
D. $(1\,(2\,3\,NULL)\,(4\,5))$

gate2000 data-structure binary-tree easy

3.4.23 Binary Tree: GATE2000-2.16

https://gateoverflow.in/663

Let LASTPOST, LASTIN and LASTPRE denote the last vertex visited `in a postorder, inorder and preorder traversal respectively, of a complete binary tree. Which of the following is always true?

A. LASTIN = LASTPOST
B. LASTIN = LASTPRE
C. LASTPRE = LASTPOST
D. None of the above

gate2000 data-structure binary-tree normal

3.4.24 Binary Tree: GATE2002-2.12

https://gateoverflow.in/842

A weight-balanced tree is a binary tree in which for each node, the number of nodes in the left sub tree is at least half and at most twice the number of nodes in the right sub tree. The maximum possible height (number of nodes on the path from the root to the furthest leaf) of such a tree on n nodes is best described by which of the following?

A. $\log_2 n$
B. $\log_{\frac{4}{3}} n$
C. $\log_3 n$
D. $\log_{\frac{3}{2}} n$

gate2002 data-structure binary-tree normal

3.4.25 Binary Tree: GATE2002-6

https://gateoverflow.in/859

Draw all binary trees having exactly three nodes labeled A, B and C on which preorder traversal gives the sequence C, B, A.

gate2002 data-structure binary-tree easy descriptive

3.4.26 Binary Tree: GATE2004-35

https://gateoverflow.in/1032

Consider the label sequences obtained by the following pairs of traversals on a labeled binary tree. Which of these pairs identify a tree uniquely?

I. preorder and postorder
II. inorder and postorder
III. preorder and inorder
IV. level order and postorder

A. I only
B. II, III
C. III only
D. IV only

gate2004 data-structure binary-tree normal

3.4.27 Binary Tree: GATE2004-43

https://gateoverflow.in/1040

Consider the following C program segment

```
struct CellNode{
    struct CellNode *leftChild
    int element;
    struct CellNode *rightChild;
    };

int Dosomething (struct CellNode *ptr)
{
    int value = 0;
    if(ptr != NULL)
    {
        if (ptr -> leftChild != NULL)
            value = 1 + DoSomething (ptr -> leftChild);
        if (ptr -> rightChild != NULL)
            value = max(value, 1 + Dosomething (ptr -> rightChild));
    }
    return(value);
```

© Copyright GATE Overflow. All rights reserved.

```
}
```

The value returned by the function $\mathrm{DoSomething}$ when a pointer to the root of a non-empty tree is passed as argument is

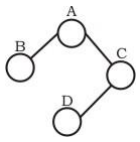
A. The number of leaf nodes in the tree
B. The number of nodes in the tree
C. The number of internal nodes in the tree
D. The height of the tree

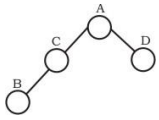
gate2004 data-structure binary-tree normal

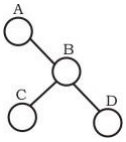
3.4.28 Binary Tree: GATE2004-IT-54

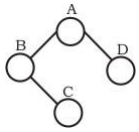
https://gateoverflow.in/3697

Which one of the following binary trees has its inorder and preorder traversals as $BCAD$ and $ABCD$, respectively?

A. 

B. 

C. 

D. 

gate2004-it binary-tree easy data-structure

3.4.29 Binary Tree: GATE2005-33

https://gateoverflow.in/1369

Postorder traversal of a given binary search tree, T produces the following sequence of keys

$10, 9, 23, 22, 27, 25, 15, 50, 95, 60, 40, 29$

Which one of the following sequences of keys can be the result of an in-order traversal of the tree T?

A. $9, 10, 15, 22, 23, 25, 27, 29, 40, 50, 60, 95$
B. $9, 10, 15, 22, 40, 50, 60, 95, 23, 25, 27, 29$
C. $29, 15, 9, 10, 25, 22, 23, 27, 40, 60, 50, 95$
D. $95, 50, 60, 40, 27, 23, 22, 25, 10, 9, 15, 29$

gate2005 data-structure binary-tree easy

3.4.30 Binary Tree: GATE2005-IT-50

https://gateoverflow.in/3811

In a binary tree, for every node the difference between the number of nodes in the left and right subtrees is at most 2. If the height of the tree is $h > 0$, then the minimum number of nodes in the tree is

A. 2^{h-1} B. $2^{h-1}+1$ C. $2^h - 1$ D. 2^h

gate2005-it data-structure binary-tree normal

3.4.31 Binary Tree: GATE2006-13

https://gateoverflow.in/974

A scheme for storing binary trees in an array X is as follows. Indexing of X starts at 1 instead of 0. the root is stored at $X[1]$. For a node stored at $X[i]$, the left child, if any, is stored in $X[2i]$ and the right child, if any, in $X[2i+1]$. To be able to store any binary tree on n vertices the minimum size of X should be

A. $\log_2 n$ B. n C. $2n+1$ D. $2^n - 1$

gate2006 data-structure binary-tree normal

3.4.32 Binary Tree: GATE2006-IT-71

https://gateoverflow.in/3615

An array X of n distinct integers is interpreted as a complete binary tree. The index of the first element of the array is 0. The index of the parent of element $X[i], i \neq 0$, is?

A. $\left\lfloor \frac{i}{2} \right\rfloor$ B. $\left\lceil \frac{i-1}{2} \right\rceil$

C. $\left\lceil \frac{i}{2} \right\rceil$ D. $\left\lceil \frac{i}{2} \right\rceil - 1$

gate2006-it data-structure binary-tree normal

© Copyright GATE Overflow. All rights reserved.

3.4.33 Binary Tree: GATE2006-IT-73

https://gateoverflow.in/3617

An array X of n distinct integers is interpreted as a complete binary tree. The index of the first element of the array is 0. If the root node is at level 0, the level of element $X[i]$, $i \neq 0$, is

A. $\lfloor \log_2 i \rfloor$
B. $\lceil \log_2 (i+1) \rceil$
C. $\lfloor \log_2 (i+1) \rfloor$
D. $\lceil \log_2 i \rceil$

gate2006-it data-structure binary-tree normal

3.4.34 Binary Tree: GATE2006-IT-9

https://gateoverflow.in/3548

In a binary tree, the number of internal nodes of degree 1 is 5, and the number of internal nodes of degree 2 is 10. The number of leaf nodes in the binary tree is

A. 10 B. 11 C. 12 D. 15

gate2006-it data-structure binary-tree normal

3.4.35 Binary Tree: GATE2007-12

https://gateoverflow.in/1210

The height of a binary tree is the maximum number of edges in any root to leaf path. The maximum number of nodes in a binary tree of height h is:

A. $2^h - 1$ B. $2^{h-1} - 1$ C. $2^{h+1} - 1$ D. 2^{h+1}

gate2007 data-structure binary-tree easy

3.4.36 Binary Tree: GATE2007-13

https://gateoverflow.in/1211

The maximum number of binary trees that can be formed with three unlabeled nodes is:

A. 1 B. 5 C. 4 D. 3

gate2007 data-structure binary-tree normal

3.4.37 Binary Tree: GATE2007-39, UGCNET-June2015-II-22

https://gateoverflow.in/1237

The inorder and preorder traversal of a binary tree are

$d\ b\ e\ a\ f\ c\ g$ and $a\ b\ d\ e\ c\ f\ g$, respectively

The postorder traversal of the binary tree is:

A. $d\ e\ b\ f\ g\ c\ a$
B. $e\ d\ b\ g\ f\ c\ a$
C. $e\ d\ b\ f\ g\ c\ a$
D. $d\ e\ f\ g\ b\ c\ a$

gate2007 data-structure binary-tree normal ugcnetjune2015ii

3.4.38 Binary Tree: GATE2007-46

https://gateoverflow.in/1244

Consider the following C program segment where $CellNode$ represents a node in a binary tree:

```
struct CellNode {
    struct CellNode *leftChild;
    int element;
    struct CellNode *rightChild;
};

int Getvalue (struct CellNode *ptr) {
    int value = 0;
    if (ptr != NULL) {
        if ((ptr->leftChild == NULL) &&
            (ptr->rightChild == NULL))
            value = 1;
        else
            value = value + GetValue(ptr->leftChild)
                      + GetValue(ptr->rightChild);
    }
    return(value);
}
```

The value returned by $GetValue$ when a pointer to the root of a binary tree is passed as its argument is:

© Copyright GATE Overflow. All rights reserved.

A. the number of nodes in the tree
B. the number of internal nodes in the tree
C. the number of leaf nodes in the tree
D. the height of the tree

gate2007 data-structure binary-tree normal

3.4.39 Binary Tree: GATE2008-IT-46

https://gateoverflow.in/3356

The following three are known to be the preorder, inorder and postorder sequences of a binary tree. But it is not known which is which.

I. $MBCAFHPYK$
II. $KAMCBYPFH$
III. $MABCKYFPH$

Pick the true statement from the following.

A. I and II are preorder and inorder sequences, respectively
B. I and III are preorder and postorder sequences, respectively
C. II is the inorder sequence, but nothing more can be said about the other two sequences
D. II and III are the preorder and inorder sequences, respectively

gate2008-it data-structure normal binary-tree

3.4.40 Binary Tree: GATE2008-IT-76

https://gateoverflow.in/3390

A binary tree with $n > 1$ nodes has n_1, n_2 and n_3 nodes of degree one, two and three respectively. The degree of a node is defined as the number of its neighbours.

n_3 can be expressed as

A. $n_1 + n_2 - 1$
B. $n_1 - 2$
C. $[((n_1 + n_2)/2)]$
D. $n_2 - 1$

gate2008-it data-structure binary-tree normal

3.4.41 Binary Tree: GATE2008-IT-77

https://gateoverflow.in/3391

A binary tree with $n > 1$ nodes has n_1, n_2 and n_3 nodes of degree one, two and three respectively. The degree of a node is defined as the number of its neighbours.

Starting with the above tree, while there remains a node v of degree two in the tree, add an edge between the two neighbours of v and then remove v from the tree. How many edges will remain at the end of the process?

A. $2 * n_1 - 3$
B. $n_2 + 2 * n_1 - 2$
C. $n_3 - n_2$
D. $n_2 + n_1 - 2$

gate2008-it data-structure binary-tree normal

3.4.42 Binary Tree: GATE2010-10

https://gateoverflow.in/2183

In a binary tree with n nodes, every node has an odd number of descendants. Every node is considered to be its own descendant. What is the number of nodes in the tree that have exactly one child?

A. 0
B. 1
C. $\frac{(n-1)}{2}$
D. $n-1$

gate2010 data-structure binary-tree normal

3.4.43 Binary Tree: GATE2011-29

https://gateoverflow.in/2131

We are given a set of n distinct elements and an unlabeled binary tree with n nodes. In how many ways can we populate the tree with the given set so that it becomes a binary search tree?

A. 0
B. 1
C. $n!$
D. $\frac{1}{n+1}.^{2n}C_n$

gate2011 binary-tree normal

3.4.44 Binary Tree: GATE2012-47

https://gateoverflow.in/2163

The height of a tree is defined as the number of edges on the longest path in the tree. The function shown in the pseudo-code below is invoked as height (root) to compute the height of a binary tree rooted at the tree pointer root.

© Copyright GATE Overflow. All rights reserved.

```
int height(treeptr n)
{ if(n == NULL) return -1;
  if(n -> left == NULL)
     if(n -> right == NULL) return 0;
     else return B1; // Box 1

  else{h1 = height(n -> left);
       if(n -> right == NULL) return (1+h1);
       else{h2 = height(n -> right);
            return B2; // Box 2
            }
       }
}
```

The appropriate expressions for the two boxes **B1** and **B2** are:

A. **B1**: $(1 + \text{height}(n \rightarrow \text{right}))$; **B2**: $(1 + \max(h1, h2))$
B. **B1**: $(\text{height}(n \rightarrow \text{right}))$; **B2**: $(1 + \max(h1, h2))$
C. **B1**: $\text{height}(n \rightarrow \text{right})$; **B2**: $\max(h1, h2)$
D. **B1**: $(1 + \text{height}(n \rightarrow \text{right}))$; **B2**: $\max(h1, h2)$

gate2012 data-structure binary-tree normal

3.4.45 Binary Tree: GATE2014-1-12

https://gateoverflow.in/1776

Consider a rooted n node binary tree represented using pointers. The best upper bound on the time required to determine the number of subtrees having exactly 4 nodes is $O(n^a \log^b n)$. Then the value of $a + 10b$ is _____.

gate2014-1 data-structure binary-tree numerical-answers normal

3.4.46 Binary Tree: GATE2015-1-25

https://gateoverflow.in/8223

The height of a tree is the length of the longest root-to-leaf path in it. The maximum and minimum number of nodes in a binary tree of height 5 are

A. 63 and 6, respectively
B. 64 and 5, respectively
C. 32 and 6, respectively
D. 31 and 5, respectively

gate2015-1 data-structure binary-tree easy

3.4.47 Binary Tree: GATE2015-2-10

https://gateoverflow.in/8059

A binary tree T has 20 leaves. The number of nodes in T having two children is _____.

gate2015-2 data-structure binary-tree normal numerical-answers

3.4.48 Binary Tree: GATE2015-3-25

https://gateoverflow.in/8428

Consider a binary tree T that has 200 leaf nodes. Then the number of nodes in T that have exactly two children are _____.

gate2015-3 data-structure binary-tree normal numerical-answers

3.4.49 Binary Tree: GATE2016-2-36

https://gateoverflow.in/39597

Consider the following New-order strategy for traversing a binary tree:

- Visit the root;
- Visit the right subtree using New-order;
- Visit the left subtree using New-order;

The New-order traversal of the expression tree corresponding to the reverse polish expression

```
3 4 * 5 - 2 ^ 6 7 * 1 + -
```

is given by:

A. $+ - 1\ 6\ 7 * 2 \wedge 5\ -\ 3\ 4 *$
B. $- + 1 * 6\ 7 \wedge 2 - 5 * 3\ 4$
C. $- + 1 * 7\ 6 \wedge 2\ -\ 5\ *\ 4\ 3$
D. $1\ 7\ 6 * + 2\ 5\ 4\ 3\ *\ - \wedge -$

© Copyright GATE Overflow. All rights reserved.

gate2016-2 data-structure binary-tree normal

3.4.50 Binary Tree: GATE2018-20

https://gateoverflow.in/204094

The postorder traversal of a binary tree is $8,9,6,7,4,5,2,3,1$. The inorder traversal of the same tree is $8,6,9,4,7,2,5,1,3$. The height of a tree is the length of the longest path from the root to any leaf. The height of the binary tree above is _____

gate2018 data-structure binary-tree numerical-answers

3.4.51 Binary Tree: GATE2019-46

https://gateoverflow.in/302802

Let T be a full binary tree with 8 leaves. (A full binary tree has every level full.) Suppose two leaves a and b of T are chosen uniformly and independently at random. The expected value of the distance between a and b in T (ie., the number of edges in the unique path between a and b) is (rounded off to 2 decimal places) _____.

So, expected path length

=0×864+2×864+4×1664+6×3264=27264=4.25

gate2019 numerical-answers data-structure binary-tree

3.4.52 Binary Tree: TIFR2012-B-16

https://gateoverflow.in/25214

Consider a complete binary tree of height n, where each edge is one Ohm resistor. Suppose all the leaves of the tree are tied together. Approximately how much is the effective resistance from the root to this bunch of leaves for very large n?

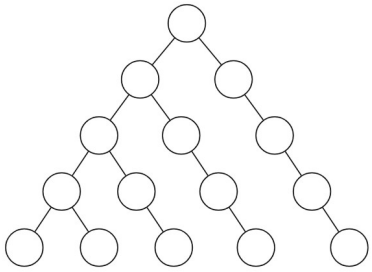
a. Exponential in n.
b. Cubic in n.
c. Linear in n.
d. Logarithmic in n.
e. Of the order square root of n.

tifr2012 binary-tree

3.4.53 Binary Tree: TIFR2013-B-13

https://gateoverflow.in/25775

Given a binary tree of the following form and having n nodes, the height of the tree is



a. $\Theta(\log n)$
b. $\Theta(n)$
c. $\Theta(\sqrt{n})$
d. $\Theta(n/\log n)$
e. None of the above.

tifr2013 binary-tree data-structure

3.4.54 Binary Tree: TIFR2014-B-1

https://gateoverflow.in/27133

Let T be a rooted binary tree whose vertices are labelled with symbols a,b,c,d,e,f,g,h,i,j,k . Suppose the in-order (visit left subtree, visit root, visit right subtree) and post-order (visit left subtree, visit right subtree, visit root) traversals of T produce the following sequences.

in-order:a,b,c,d,e,f,g,h,i,j,k

post-order:a,c,b,e,f,h,j,k,i,g,d

How many leaves does the tree have?

a. THREE.
b. FOUR.
c. FIVE.
d. SIX.
e. Cannot be determined uniquely from the given information.

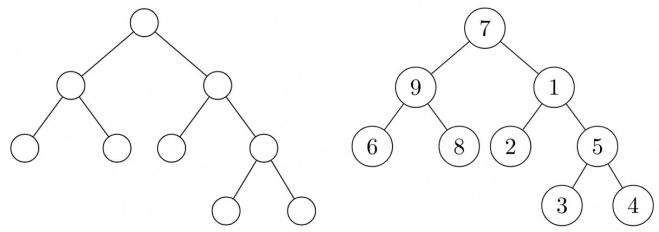
tifr2014 binary-tree data-structure easy

© Copyright GATE Overflow. All rights reserved.

3.4.55 Binary Tree: TIFR2015-B-4

https://gateoverflow.in/29849

First, consider the tree on the left.



On the right, the nine nodes of the tree have been assigned numbers from the set $\{1,2,\ldots,9\}$ so that for every node, the numbers in its left subtree and right subtree lie in disjoint intervals (that is, all numbers in one subtree are less than all numbers in the other subtree). How many such assignments are possible? Hint: Fix a value for the root and ask what values can then appear in its left and right subtrees.

A. $2^9 = 512$ B. $2^4.3^2.5.9 = 6480$ C. $2^3.3.5.9 = 1080$ D. $2^4 = 16$ E. $2^3.3^3 = 216$

tifr2015 binary-tree permutation-and-combination

3.4.56 Binary Tree: TIFR2018-B-6

https://gateoverflow.in/179290

Consider the following implementation of a binary tree data strucrure. The operator $+$ denotes list-concatenation. That is, $[a,b,c]+[d,e]=[a,b,c,d,e]$.

```
struct TreeNode:
 int value
 TreeNode leftChild
 TreeNode rightChild

function preOrder(T):
 if T == null:
  return []
 else:
  return [T.value] + preOrder(T.leftChild) + preOrder(T.rightChild)

function inOrder(T):
 if T == null:
  return []
 else:
  return inOrder(T.leftChild) + [T.value] + inOrder(T.rightChild)

function postOrder(T):
 if T == null:
  return []
 else:
  return postOrder(T.leftChild) + postOrder(T.rightChild) + [T.value]
```

For some T the functions inOrder(T) and preOrder(T) return the following:

$\text{inOrder(T)}: \quad [12,10,6,9,7,2,15,5,1,13,4,3,8,14,11]$

$\text{preOrder(T)}: \quad [5,2,10,12,9,6,7,15,13,1,3,4,14,8,11]$

What does postOrder(T) return ?

A. $[12,6,10,7,15,2,9,1,4,13,8,11,14,3,5]$
B. $[11,8,14,4,3,1,13,15,7,6,9,12,10,2,5]$
C. $[11,14,8,3,4,13,1,5,15,2,7,9,6,10,12]$
D. $[12,6,7,9,10,15,2,1,4,8,11,14,3,13,5]$
E. Cannot be uniquely determined from given information.

tifr2018 data-structure binary-tree

3.5 Graph Search (1)

3.5.1 Graph Search: GATE1989-3-ixa

https://gateoverflow.in/87143

Answer the following:

Which one of the following statements (s) is/are FALSE?

© Copyright GATE Overflow. All rights reserved.

A. Overlaying is used to run a program, which is longer than the address space of the computer.
B. Optimal binary search tree construction can be performed efficiently by using dynamic programming.
C. Depth first search cannot be used to find connected components of a graph.
D. Given the prefix and postfix walls over a binary tree, the binary tree can be uniquely constructed.

normal gate1989 binary-tree graph-search

3.6 Graphs (6)

3.6.1 Graphs: GATE1992-03,iii

https://gateoverflow.in/580

How many edges can there be in a forest with p components having n vertices in all?

gate1992 data-structure graphs easy

3.6.2 Graphs: GATE1997-6.2

https://gateoverflow.in/2258

Let G be the graph with 100 vertices numbered 1 to 100. Two vertices i and j are adjacent if $|i-j|=8$ or $|i-j|=12$. The number of connected components in G is

A. 8 B. 4 C. 12 D. 25

gate1997 data-structure normal graphs

3.6.3 Graphs: GATE2008-42

https://gateoverflow.in/1872

G is a graph on n vertices and $2n-2$ edges. The edges of G can be partitioned into two edge-disjoint spanning trees. Which of the following is NOT true for G?

A. For every subset of k vertices, the induced subgraph has at most $2k-2$ edges.
B. The minimum cut in G has at least 2 edges.
C. There are at least 2 edge-disjoint paths between every pair of vertices.
D. There are at least 2 vertex-disjoint paths between every pair of vertices.

gate2008 data-structure graphs normal

3.6.4 Graphs: GATE2008-IT-4

https://gateoverflow.in/3264

What is the size of the smallest MIS (Maximal Independent Set) of a chain of nine nodes?

A. 5 B. 4 C. 3 D. 2

gate2008-it data-structure normal graphs

3.6.5 Graphs: GATE2014-1-3

https://gateoverflow.in/1754

Let $G=(V,E)$ be a directed graph where V is the set of vertices and E the set of edges. Then which one of the following graphs has the same strongly connected components as G ?

A. $G_1=(V,E_1)$ where $E_1=\{(u,v) \mid (u,v) \notin E\}$
B. $G_2=(V,E_2)$ where $E_2=\{(u,v) \mid (v,u) \in E\}$
C. $G_3=(V,E_3)$ where $E_3=\{(u,v) \mid$ there is a path of length ≤ 2 from u to v in $E\}$
D. $G_4=(V_4,E)$ where V_4 is the set of vertices in G which are not isolated

gate2014-1 data-structure graphs ambiguous

3.6.6 Graphs: GATE2016-1-38

https://gateoverflow.in/39731

Consider the weighted undirected graph with 4 vertices, where the weight of edge $\{i,j\}$ is given by the entry W_{ij} in the matrix W.

$$W=\begin{bmatrix} 0 & 2 & 8 & 5 \\ 2 & 0 & 5 & 8 \\ 8 & 5 & 0 & x \\ 5 & 8 & x & 0 \end{bmatrix}$$

© Copyright GATE Overflow. All rights reserved.

The largest possible integer value of x, for which at least one shortest path between some pair of vertices will contain the edge with weight x is _____.

gate2016-1 data-structure graphs normal numerical-answers

3.7 Hashing (17)

3.7.1 Hashing: GATE1989-1-vii, ISRO2015-14

https://gateoverflow.in/10905

A hash table with ten buckets with one slot per bucket is shown in the following figure. The symbols $S1$ to $S7$ initially entered using a hashing function with linear probing. The maximum number of comparisons needed in searching an item that is not present is

0	S7
1	S1
2	
3	S4
4	S2
5	
6	S5
7	
8	S6
9	S3

A. 4 B. 5 C. 6 D. 3

hashing isro2015 gate1989 data-structure normal

3.7.2 Hashing: GATE1996-1.13

https://gateoverflow.in/2717

An advantage of chained hash table (external hashing) over the open addressing scheme is

A. Worst case complexity of search operations is less
B. Space used is less
C. Deletion is easier
D. None of the above

gate1996 data-structure hashing normal

3.7.3 Hashing: GATE1996-15

https://gateoverflow.in/2767

Insert the characters of the string $K\ R\ P\ C\ S\ N\ Y\ T\ J\ M$ into a hash table of size 10.
Use the hash function

$$h(x) = (ord(x) - ord(\text{" a "}) + 1) \mod 10$$

and linear probing to resolve collisions.

A. Which insertions cause collisions?
B. Display the final hash table.

gate1996 data-structure hashing normal

3.7.4 Hashing: GATE1997-12

https://gateoverflow.in/2272

Consider a hash table with n buckets, where external (overflow) chaining is used to resolve collisions. The hash function is such that the probability that a key value is hashed to a particular bucket is $\frac{1}{n}$. The hash table is initially empty and K distinct values are inserted in the table.

A. What is the probability that bucket number 1 is empty after the K^{th} insertion?
B. What is the probability that no collision has occurred in any of the K insertions?
C. What is the probability that the first collision occurs at the K^{th} insertion?

gate1997 data-structure hashing probability normal

© Copyright GATE Overflow. All rights reserved.

3.7.5 Hashing: GATE2004-7

https://gateoverflow.in/1004

Given the following input $(4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199)$ and the hash function $x \bmod 10$, which of the following statements are true?

I. $9679, 1989, 4199$ hash to the same value
II. $1471, 6171$ hash to the same value
III. All elements hash to the same value
IV. Each element hashes to a different value

A. I only B. II only C. I and II only D. III or IV

gate2004 data-structure hashing easy

3.7.6 Hashing: GATE2005-IT-16

https://gateoverflow.in/3761

A hash table contains 10 buckets and uses linear probing to resolve collisions. The key values are integers and the hash function used is key % 10. If the values $43, 165, 62, 123, 142$ are inserted in the table, in what location would the key value 142 be inserted?

A. 2 B. 3 C. 4 D. 6

gate2005-it data-structure hashing easy

3.7.7 Hashing: GATE2006-IT-20

https://gateoverflow.in/3559

Which of the following statement(s) is TRUE?

I. A hash function takes a message of arbitrary length and generates a fixed length code.
II. A hash function takes a message of fixed length and generates a code of variable length.
III. A hash function may give the same hash value for distinct messages.

A. I only B. II and III only C. I and III only D. II only

gate2006-it data-structure hashing normal

3.7.8 Hashing: GATE2007-40

https://gateoverflow.in/1238

Consider a hash table of size seven, with starting index zero, and a hash function $(3x + 4) \mod 7$. Assuming the hash table is initially empty, which of the following is the contents of the table when the sequence $1, 3, 8, 10$ is inserted into the table using closed hashing? Note that − denotes an empty location in the table.

A. $8, –, –, –, –, –, 10$
B. $1, 8, 10, –, –, –, 3$
C. $1, –, –, –, –, –, 3$
D. $1, 10, 8, –, –, –, 3$

gate2007 data-structure hashing easy

3.7.9 Hashing: GATE2007-IT-28

https://gateoverflow.in/3461

Consider a hash function that distributes keys uniformly. The hash table size is 20. After hashing of how many keys will the probability that any new key hashed collides with an existing one exceed 0.5.

A. 5 B. 6 C. 7 D. 10

gate2007-it data-structure hashing probability normal

3.7.10 Hashing: GATE2008-IT-48

https://gateoverflow.in/3358

Consider a hash table of size 11 that uses open addressing with linear probing. Let $h(k) = k \mod 11$ be the hash function used. A sequence of records with keys

43 36 92 87 11 4 71 13 14

is inserted into an initially empty hash table, the bins of which are indexed from zero to ten. What is the index of the bin into which the last record is inserted?

A. 3 B. 4 C. 6 D. 7

gate2008-it data-structure hashing normal

© Copyright GATE Overflow. All rights reserved.

3.7.11 Hashing: GATE2009-36

https://gateoverflow.in/1322

The keys $12,18,13,2,3,23,5$ and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function $h(k) = k \mod 10$ and linear probing. What is the resultant hash table?

A.

0	
1	
2	2
3	23
4	
5	15
6	
7	
8	18
9	

B.

0	
1	
2	12
3	13
4	
5	5
6	
7	
8	18
9	

C.

0	
1	
2	12
3	13
4	2
5	3
6	23
7	5
8	18
9	15

D.

0	
1	
2	2,12
3	13,3,23
4	
5	5,15
6	
7	
8	18
9	

gate2009 data-structure hashing normal

3.7.12 Hashing: GATE2010-52

https://gateoverflow.in/2360

A hash table of length 10 uses open addressing with hash function $h(k) = k \mod 10$, and linear probing. After inserting 6 values into an empty hash table, the table is shown as below

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

Which one of the following choices gives a possible order in which the key values could have been inserted in the table?

A. $46,42,34,52,23,33$
B. $34,42,23,52,33,46$
C. $46,34,42,23,52,33$
D. $42,46,33,23,34,52$

gate2010 data-structure hashing normal

3.7.13 Hashing: GATE2010-53

https://gateoverflow.in/43327

A hash table of length 10 uses open addressing with hash function $h(k) = k \, mod \, 10$, and linear probing. After inserting 6 values into an empty hash table, the table is shown as below

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

How many different insertion sequences of the key values using the same hash function and linear probing will result in the hash table shown above?

© Copyright GATE Overflow. All rights reserved.

A. 10 B. 20 C. 30 D. 40

data-structure hashing normal gate2010

3.7.14 Hashing: GATE2014-1-40

https://gateoverflow.in/1918

Consider a hash table with 9 slots. The hash function is $h(k) = k \mod 9$. The collisions are resolved by chaining. The following 9 keys are inserted in the order: $5, 28, 19, 15, 20, 33, 12, 17, 10$. The maximum, minimum, and average chain lengths in the hash table, respectively, are

A. $3, 0,$ and 1 B. $3, 3,$ and 3 C. $4, 0,$ and 1 D. $3, 0,$ and 2

gate2014-1 data-structure hashing normal

3.7.15 Hashing: GATE2014-3-40

https://gateoverflow.in/2074

Consider a hash table with 100 slots. Collisions are resolved using chaining. Assuming simple uniform hashing, what is the probability that the first 3 slots are unfilled after the first 3 insertions?

A. $(97 \times 97 \times 97)/100^3$
B. $(99 \times 98 \times 97)/100^3$
C. $(97 \times 96 \times 95)/100^3$
D. $(97 \times 96 \times 95/(3! \times 100^3)$

gate2014-3 data-structure hashing probability normal

3.7.16 Hashing: GATE2015-2-33

https://gateoverflow.in/8152

Which one of the following hash functions on integers will distribute keys most uniformly over 10 buckets numbered 0 to 9 for i ranging from 0 to 2020?

A. $h(i) = i^2 \mod 10$
B. $h(i) = i^3 \mod 10$
C. $h(i) = (11 * i^2) \mod 10$
D. $h(i) = (12 * i^2) \mod 10$

gate2015-2 data-structure hashing normal

3.7.17 Hashing: GATE2015-3-17

https://gateoverflow.in/8414

Given that hash table T with 25 slots that stores 2000 elements, the load factor a for T is _____.

gate2015-3 data-structure hashing normal numerical-answers

3.8 Heap (25)

3.8.1 Heap: GATE1990-2-viii

https://gateoverflow.in/83993

Match the pairs in the following questions:

(a)	A heap construction	(p)	$\Omega(n \log_{10} n)$
(b)	Constructing Hashtable with linear probing	(q)	$O(n)$
(c)	AVL tree construction	(r)	$O(n^2)$
(d)	Digital trie construction	(s)	$O(n \log_{10} n)$

gate1990 match-the-following data-structure heap

3.8.2 Heap: GATE1996-2.11

https://gateoverflow.in/2740

The minimum number of interchanges needed to convert the array into a max-heap is

$89, 19, 40, 17, 12, 10, 2, 5, 7, 11, 6, 9, 70$

A. 0 B. 1 C. 2 D. 3

gate1996 data-structure heap easy

3.8.3 Heap: GATE1999-12

https://gateoverflow.in/1511

A. In binary tree, a full node is defined to be a node with 2 children. Use induction on the height of the binary tree to prove that the number of full nodes plus one is equal to the number of leaves.

© Copyright GATE Overflow. All rights reserved.

B. Draw the min-heap that results from insertion of the following elements in order into an initially empty min-heap: $7,6,5,4,3,2,1$. Show the result after the deletion of the root of this heap.

gate1999 data-structure heap normal

3.8.4 Heap: GATE2001-1.15

https://gateoverflow.in/708

Consider any array representation of an n element binary heap where the elements are stored from index 1 to index n of the array. For the element stored at index i of the array $(i \leq n)$, the index of the parent is

A. $i-1$ B. $\lfloor \frac{i}{2} \rfloor$ C. $\lceil \frac{i}{2} \rceil$ D. $\frac{(i+1)}{2}$

gate2001 data-structure heap easy

3.8.5 Heap: GATE2003-23

https://gateoverflow.in/1110

In a min-heap with n elements with the smallest element at the root, the 7^{th} smallest element can be found in time

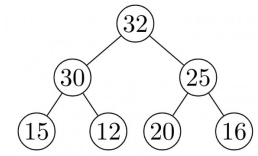
A. $\Theta(n \log n)$ B. $\Theta(n)$
C. $\Theta(\log n)$ D. $\Theta(1)$

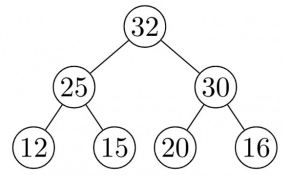
gate2003 data-structure heap

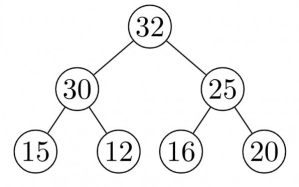
3.8.6 Heap: GATE2004-37

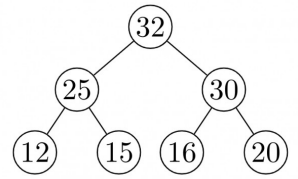
https://gateoverflow.in/1034

The elements $32,15,20,30,12,25,16,$ are inserted one by one in the given order into a maxHeap. The resultant maxHeap is

A.


B.


C.


D.


gate2004 data-structure heap normal

3.8.7 Heap: GATE2004-IT-53

https://gateoverflow.in/3696

An array of integers of size n can be converted into a heap by adjusting the heaps rooted at each internal node of the complete binary tree starting at the node $\lfloor (n-1)/2 \rfloor$, and doing this adjustment up to the root node (root node is at index 0) in the order $\lfloor (n-1)/2 \rfloor$, $\lfloor (n-3)/2 \rfloor$, ..., 0. The time required to construct a heap in this manner is

A. $O(\log n)$ B. $O(n)$ C. $O(n \log \log n)$ D. $O(n \log n)$

By using Build Heap method we can create heap from complete binary tree.

which will take O(n).

gate2004-it data-structure heap normal

3.8.8 Heap: GATE2005-34

https://gateoverflow.in/1370

A priority queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is: $10,8,5,3,2$. Two new elements 1 and 7 are inserted into the heap in that order. The level-order traversal of the heap after the insertion of the elements is:

A. $10,8,7,5,3,2,1$ B. $10,8,7,2,3,1,5$
C. $10,8,7,1,2,3,5$ D. $10,8,7,3,2,1,5$

gate2005 data-structure heap normal

3.8.9 Heap: GATE2006-10

https://gateoverflow.in/889

In a binary max heap containing n numbers, the smallest element can be found in time

A. $O(n)$ B. $O(\log n)$

© Copyright GATE Overflow. All rights reserved.

C. $O(\log \log n)$ D. $O(1)$

gate2006 data-structure heap easy

3.8.10 Heap: GATE2006-76

https://gateoverflow.in/1852

Statement for Linked Answer Questions 76 & 77:

A 3-ary max heap is like a binary max heap, but instead of 2 children, nodes have 3 children. A 3-ary heap can be represented by an array as follows: The root is stored in the first location, $a[0]$, nodes in the next level, from left to right, is stored from $a[1]$ to $a[3]$. The nodes from the second level of the tree from left to right are stored from $a[4]$ location onward. An item x can be inserted into a 3-ary heap containing n items by placing x in the location $a[n]$ and pushing it up the tree to satisfy the heap property.

76. Which one of the following is a valid sequence of elements in an array representing 3-ary max heap?

A. $1,3,5,6,8,9$ B. $9,6,3,1,8,5$
C. $9,3,6,8,5,1$ D. $9,5,6,8,3,1$

gate2006 data-structure heap normal

3.8.11 Heap: GATE2006-77

https://gateoverflow.in/87191

Statement for Linked Answer Questions 76 & 77:

A 3-ary max heap is like a binary max heap, but instead of 2 children, nodes have 3 children. A 3-ary heap can be represented by an array as follows: The root is stored in the first location, $a[0]$, nodes in the next level, from left to right, is stored from $a[1]$ to $a[3]$. The nodes from the second level of the tree from left to right are stored from $a[4]$ location onward. An item x can be inserted into a 3-ary heap containing n items by placing x in the location $a[n]$ and pushing it up the tree to satisfy the heap property.

77. Suppose the elements $7, 2, 10$ and 4 are inserted, in that order, into the valid 3-ary max heap found in the previous question, Q.76. Which one of the following is the sequence of items in the array representing the resultant heap?

A. $10,7,9,8,3,1,5,2,6,4$ B. $10,9,8,7,6,5,4,3,2,1$
C. $10,9,4,5,7,6,8,2,1,3$ D. $10,8,6,9,7,2,3,4,1,5$

gate2006 data-structure heap normal

3.8.12 Heap: GATE2006-IT-44

https://gateoverflow.in/3587

Which of the following sequences of array elements forms a heap?

A. $\{23,17,14,6,13,10,1,12,7,5\}$ B. $\{23,17,14,6,13,10,1,5,7,12\}$
C. $\{23,17,14,7,13,10,1,5,6,12\}$ D. $\{23,17,14,7,13,10,1,12,5,7\}$

gate2006-it data-structure heap easy

3.8.13 Heap: GATE2006-IT-72

https://gateoverflow.in/3616

An array X of n distinct integers is interpreted as a complete binary tree. The index of the first element of the array is 0. If only the root node does not satisfy the heap property, the algorithm to convert the complete binary tree into a heap has the best asymptotic time complexity of

A. $O(n)$ B. $O(\log n)$ C. $O(n \log n)$ D. $O(n \log \log n)$

gate2006-it data-structure heap easy

3.8.14 Heap: GATE2007-47

https://gateoverflow.in/1245

Consider the process of inserting an element into a $Max\ Heap$, where the $Max\ Heap$ is represented by an $array$. Suppose we perform a binary search on the path from the new leaf to the root to find the position for the newly inserted element, the number of $comparisons$ performed is:

A. $\Theta(\log_2 n)$ B. $\Theta(\log_2 \log_2 n)$
C. $\Theta(n)$ D. $\Theta(n \log_2 n)$

gate2007 data-structure heap normal

© Copyright GATE Overflow. All rights reserved.

3.8.15 Heap: GATE2009-59

https://gateoverflow.in/1341

Consider a binary max-heap implemented using an array.
Which one of the following array represents a binary max-heap?

A. $\{25,12,16,13,10,8,14\}$
B. $\{25,14,13,16,10,8,12\}$
C. $\{25,14,16,13,10,8,12\}$
D. $\{25,14,12,13,10,8,16\}$

gate2009 data-structure heap normal

3.8.16 Heap: GATE2009-60

https://gateoverflow.in/43466

Consider a binary max-heap implemented using an array.
What is the content of the array after two delete operations on $\{25,14,16,13,10,8,12\}$

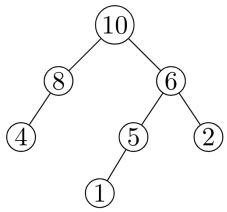
A. $\{14,13,12,10,8\}$
B. $\{14,12,13,8,10\}$
C. $\{14,13,8,12,10\}$
D. $\{14,13,12,8,10\}$

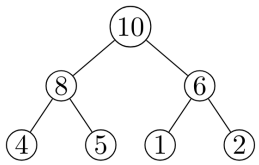
gate2009 data-structure heap normal

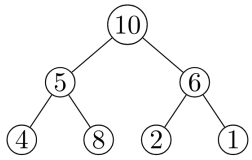
3.8.17 Heap: GATE2011-23

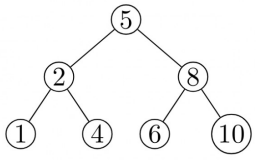
https://gateoverflow.in/2125

A max-heap is a heap where the value of each parent is greater than or equal to the value of its children. Which of the following is a max-heap?

A.


B.


C.


D.


gate2011 data-structure heap easy

3.8.18 Heap: GATE2014-2-12

https://gateoverflow.in/1967

A priority queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is: $10,8,5,3,2$. Two new elements 1 and 7 are inserted into the heap in that order. The level-order traversal of the heap after the insertion of the elements is:

A. $10,8,7,3,2,1,5$
B. $10,8,7,2,3,1,5$
C. $10,8,7,1,2,3,5$
D. $10,8,7,5,3,2,1$

gate2014-2 data-structure heap normal

3.8.19 Heap: GATE2015-1-32

https://gateoverflow.in/8273

Consider a max heap, represented by the array: $40,30,20,10,15,16,17,8,4$.

Array index	1	2	3	4	5	6	7	8	9
Value	40	30	20	10	15	16	17	8	4

Now consider that a value 35 is inserted into this heap. After insertion, the new heap is

A. $40,30,20,10,15,16,17,8,4,35$
B. $40,35,20,10,30,16,17,8,4,15$
C. $40,30,20,10,35,16,17,8,4,15$
D. $40,35,20,10,15,16,17,8,4,30$

gate2015-1 data-structure heap easy

3.8.20 Heap: GATE2015-2-17

https://gateoverflow.in/8091

Consider a complete binary tree where the left and right subtrees of the root are max-heaps. The lower bound for the number of operations to convert the tree to a heap is

© Copyright GATE Overflow. All rights reserved.

A. $\Omega(\log n)$
B. $\Omega(n)$
C. $\Omega(n \log n)$
D. $\Omega(n^2)$

gate2015-2 data-structure heap normal

3.8.21 Heap: GATE2015-3-19

https://gateoverflow.in/8418

Consider the following array of elements.

$\langle 89, 19, 50, 17, 12, 15, 2, 5, 7, 11, 6, 9, 100 \rangle$

The minimum number of interchanges needed to convert it into a max-heap is

A. 4
B. 5
C. 2
D. 3

gate2015-3 data-structure heap normal

3.8.22 Heap: GATE2016-1-37

https://gateoverflow.in/39706

An operator $delete(i)$ for a binary heap data structure is to be designed to delete the item in the i-th node. Assume that the heap is implemented in an array and i refers to the i-th index of the array. If the heap tree has depth d (number of edges on the path from the root to the farthest leaf), then what is the time complexity to re-fix the heap efficiently after the removal of the element?

A. $O(1)$
B. $O(d)$ but not $O(1)$
C. $O(2^d)$ but not $O(d)$
D. $O(d\, 2^d)$ but not $O(2^d)$

gate2016-1 data-structure heap normal

3.8.23 Heap: GATE2016-2-34

https://gateoverflow.in/39585

A complete binary min-heap is made by including each integer in $[1, 1023]$ exactly once. The depth of a node in the heap is the length of the path from the root of the heap to that node. Thus, the root is at depth 0. The maximum depth at which integer 9 can appear is _____.

gate2016-2 data-structure heap normal numerical-answers

3.8.24 Heap: GATE2019-40

https://gateoverflow.in/302808

Consider the following statements:

I. The smallest element in a max-heap is always at a leaf node
II. The second largest element in a max-heap is always a child of a root node
III. A max-heap can be constructed from a binary search tree in $\theta(n)$ time
IV. A binary search tree can be constructed from a max-heap in $\theta(n)$ time

Which of te above statements are TRUE?

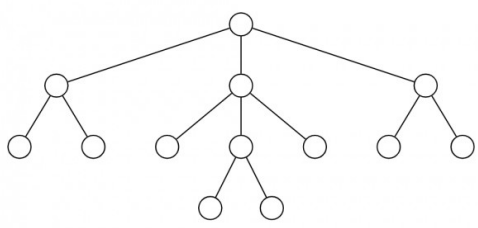
A. I, II and III
B. I, II and IV
C. I, III and IV
D. II, III and IV

gate2019 data-structure heap

3.8.25 Heap: TIFR2014-B-19

https://gateoverflow.in/27352

Consider the following tree with 13 nodes.



Suppose the nodes of the tree are randomly assigned distinct labels from $\{1, 2, \ldots, 13\}$, each permutation being equally likely. What is the probability that the labels form a min-heap (i.e., every node receives the minimum label in its subtree)?

A. $\left(\frac{1}{6!}\right)\left(\frac{1}{3!}\right)^2$
B. $\left(\frac{1}{3!}\right)^2\left(\frac{1}{2!}\right)^3$
C. $\left(\frac{1}{13}\right)\left(\frac{1}{6}\right)\left(\frac{1}{3}\right)^3$
D. $\frac{2}{13}$
E. $\frac{1}{2^{13}}$

tifr2014 heap

© Copyright GATE Overflow. All rights reserved.

3.9 Infix Postfix (2)

3.9.1 Infix Postfix: GATE1997-1.7

https://gateoverflow.in/2223

Which of the following is essential for converting an infix expression to the postfix form efficiently?

A. An operator stack
B. An operand stack
C. An operand stack and an operator stack
D. A parse tree

gate1997 normal infix-postfix stack data-structure

3.9.2 Infix Postfix: GATE1998-19b

https://gateoverflow.in/15708

Compute the post fix equivalent of the following expression $3^* \log(x+1) - \frac{a}{2}$

gate1998 stack infix-postfix

3.10 Linked Lists (19)

3.10.1 Linked Lists: GATE1987-1-xv

https://gateoverflow.in/80298

In a circular linked list oraganisation, insertion of a record involves modification of

A. One pointer.
B. Two pointers.
C. Multiple pointers.
D. No pointer.

gate1987 data-structure linked-lists

3.10.2 Linked Lists: GATE1987-6a

https://gateoverflow.in/82419

A list of n elements is commonly written as a sequence of n elements enclosed in a pair of square brackets. For example. $[10,20,30]$ is a list of three elements and $[]$ is a nil list. Five functions are defined below:

- $car(l)$ returns the first element of its argument list l;
- $cdr(l)$ returns the list obtained by removing the first element of the argument list l;
- $glue(a,l)$ returns a list m such that $car(m) = a$ and $cdr(m) = l$.
- $f(x,y) \equiv$ if $x = []$ then y
  else $glue(car(x), f(cdr(x),y))$;
- $g(x) \equiv$ if $x = []$ then $[]$
  else $f(g(cdr(x))\,,\, glue(car(x),[]))$

What do the following compute?

(a) $f([32,16,8],[9,11,12])$

(b) $g([5,1,8,9])$

gate1987 data-structure linked-lists

3.10.3 Linked Lists: GATE1993-13

https://gateoverflow.in/2310

Consider a singly linked list having n nodes. The data items $d_1, d_2, \ldots d_n$ are stored in these n nodes. Let X be a pointer to the j^{th} node $(1 \leq j \leq n)$ in which d_j is stored. A new data item d stored in node with address Y is to be inserted. Give an algorithm to insert d into the list to obtain a list having items $d_1, d_2, \ldots, d_j, d, \ldots, d_n$ in order without using the header.

gate1993 data-structure linked-lists normal

3.10.4 Linked Lists: GATE1994-1.17, UGCNET-Sep2013-II-32

https://gateoverflow.in/2460

Linked lists are not suitable data structures for which one of the following problems?

A. Insertion sort
B. Binary search
C. Radix sort
D. Polynomial manipulation

gate1994 data-structure linked-lists normal ugcnetsep2013ii

© Copyright GATE Overflow. All rights reserved.

3.10.5 Linked Lists: GATE1995-2.22

https://gateoverflow.in/2634

Which of the following statements is true?

I. As the number of entries in a hash table increases, the number of collisions increases.
II. Recursive programs are efficient
III. The worst case complexity for Quicksort is $O(n^2)$
IV. Binary search using a linear linked list is efficient

A. I and II B. II and III C. I and IV D. I and III

gate1995 data-structure linked-lists hashing

3.10.6 Linked Lists: GATE1997-1.4

https://gateoverflow.in/2220

The concatenation of two lists is to be performed on $O(1)$ time. Which of the following implementations of a list should be used?

A. Singly linked list
B. Doubly linked list
C. Circular doubly linked list
D. Array implementation of list

gate1997 data-structure linked-lists easy

3.10.7 Linked Lists: GATE1997-18

https://gateoverflow.in/2278

Consider the following piece of 'C' code fragment that removes duplicates from an ordered list of integers.

```
Node *remove-duplicates (Node* head, int *j)
{
    Node *t1, *t2; *j=0;
    t1 = head;
    if (t1! = NULL)
        t2 = t1 ->next;
    else return head;
    *j = 1;
    if(t2 == NULL) return head;
    while (t2 != NULL)
    {
        if (t1.val != t2.val) ----------------> (S1)
        {
            (*j)++;
            t1 -> next = t2;
            t1 = t2; -----> (S2)
        }
            t2 = t2 ->next;
    }
    t1 -> next = NULL;
    return head;
}
```

Assume the list contains n elements ($n \geq 2$) in the following questions.

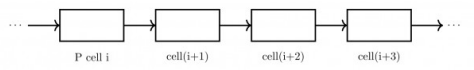
a. How many times is the comparison in statement $S1$ made?
b. What is the minimum and the maximum number of times statements marked $S2$ get executed?
c. What is the significance of the value in the integer pointed to by j when the function completes?

gate1997 data-structure linked-lists normal

3.10.8 Linked Lists: GATE1998-19a

https://gateoverflow.in/1733

a. Let p be a pointer as shown in the figure in a single linked list.



What do the following assignment statements achieve?

```
q: = p -> next
p -> next:= q -> next
```

© Copyright GATE Overflow. All rights reserved.

```
q -> next:=(q -> next) -> next
(p -> next) -> next:= q
```

gate1998 data-structure linked-lists normal

3.10.9 Linked Lists: GATE1999-11b

https://gateoverflow.in/93575

Write a constant time algorithm to insert a node with data D just before the node with address p of a singly linked list.

gate1999 data-structure linked-lists

3.10.10 Linked Lists: GATE2002-1.5

https://gateoverflow.in/809

In the worst case, the number of comparisons needed to search a single linked list of length n for a given element is

A. $\log n$ B. $\frac{n}{2}$ C. $\log_2 n - 1$ D. n

gate2002 easy data-structure linked-lists

3.10.11 Linked Lists: GATE2003-90

https://gateoverflow.in/973

Consider the function f defined below.

```
struct item {
        int data;
        struct item * next;
};
int f(struct item *p) {
    return ((p == NULL) || (p->next == NULL)||
        ((p->data <= p ->next -> data) &&
        f(p->next)));
}
```

For a given linked list p, the function f returns 1 if and only if

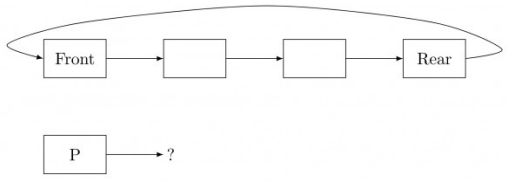
A. the list is empty or has exactly one element
B. the elements in the list are sorted in non-decreasing order of data value
C. the elements in the list are sorted in non-increasing order of data value
D. not all elements in the list have the same data value

gate2003 data-structure linked-lists normal

3.10.12 Linked Lists: GATE2004-36

https://gateoverflow.in/1033

A circularly linked list is used to represent a Queue. A single variable p is used to access the Queue. To which node should p point such that both the operations enQueue and deQueue can be performed in constant time?



A. rear node
B. front node
C. not possible with a single pointer
D. node next to front

gate2004 data-structure linked-lists normal

3.10.13 Linked Lists: GATE2004-40

https://gateoverflow.in/1037

Suppose each set is represented as a linked list with elements in arbitrary order. Which of the operations among union, intersection, membership, cardinality will be the slowest?

A. union only
B. intersection, membership
C. membership, cardinality
D. union, intersection

gate2004 data-structure linked-lists normal

© Copyright GATE Overflow. All rights reserved.

3.10.14 Linked Lists: GATE2004-IT-13

https://gateoverflow.in/3654

Let P be a singly linked list. Let Q be the pointer to an intermediate node x in the list. What is the worst-case time complexity of the best-known algorithm to delete the node x from the list ?

A. $O(n)$ B. $O(\log^2 n)$ C. $O(\log n)$ D. $O(1)$

gate2004-it data-structure linked-lists normal ambiguous

3.10.15 Linked Lists: GATE2005-IT-54

https://gateoverflow.in/3815

The following C function takes a singly-linked list of integers as a parameter and rearranges the elements of the list. The list is represented as pointer to a structure. The function is called with the list containing the integers $1,2,3,4,5,6,7$ in the given order. What will be the contents of the list after the function completes execution?

```
struct node {int value; struct node *next;);
void rearrange (struct node *list) {
    struct node *p, *q;
    int temp;
    if (!list || !list -> next) return;
    p = list; q = list -> next;
    while (q) {
        temp = p -> value;
        p -> value = q -> value;
        q -> value = temp;
        p = q -> next;
        q = p ? p -> next : 0;
      }
}
```

A. $1,2,3,4,5,6,7$
B. $2,1,4,3,6,5,7$
C. $1,3,2,5,4,7,6$
D. $2,3,4,5,6,7,1$

gate2005-it data-structure linked-lists normal

3.10.16 Linked Lists: GATE2008-62

https://gateoverflow.in/485

The following C function takes a single-linked list of integers as a parameter and rearranges the elements of the list. The function is called with the list containing the integers $1,2,3,4,5,6,7$ in the given order. What will be the contents of the list after function completes execution?

```
struct node {
    int value;
    struct node *next;
};

void rearrange(struct node *list) {
    struct node *p, *q;
    int temp;
    if (!list || !list -> next) return;
    p = list; q = list -> next;
    while(q) {
        temp = p -> value; p->value =  q -> value;
        q->value = temp; p = q ->next;
        q = p? p ->next : 0;
    }
}
```

A. $1,2,3,4,5,6,7$
B. $2,1,4,3,6,5,7$
C. $1,3,2,5,4,7,6$
D. $2,3,4,5,6,7,1$

gate2008 data-structure linked-lists normal

3.10.17 Linked Lists: GATE2010-36

https://gateoverflow.in/2337

The following C function takes a singly-linked list as input argument. It modifies the list by moving the last element to the front of the list and returns the modified list. Some part of the code is left blank.

```
typedef struct node
{
    int value;
    struct node *next;
} node;
Node *move_to-front(Node *head)
{
    Node *p, *q;
    if ((head == NULL) || (head -> next == NULL))
        return head;
```

© Copyright GATE Overflow. All rights reserved.

```
    q = NULL;
    p = head;
    while (p->next != NULL)
    {
        q=p;
        p=p->next;
    }
    _____________

    return head;

}
```

Choose the correct alternative to replace the blank line.

A. $q = NULL; p \rightarrow next = head; head = p$;
B. $q \rightarrow next = NULL; head = p; p \rightarrow next = head$;
C. $head = p; p \rightarrow next = q; q \rightarrow next = NULL$;
D. $q \rightarrow next = NULL; p \rightarrow next = head; head = p$;

gate2010 data-structure linked-lists normal

3.10.18 Linked Lists: GATE2016-2-15

https://gateoverflow.in/39557

N items are stored in a sorted doubly linked list. For a *delete* operation, a pointer is provided to the record to be deleted. For a *decrease-key* operation, a pointer is provided to the record on which the operation is to be performed.

An algorithm performs the following operations on the list in this order: $\Theta(N)$ *delete*, $O(\log N)$ *insert*, $O(\log N)$ *find*, and $\Theta(N)$ *decrease-key*. What is the time complexity of all these operations put together?

A. $O(\log^2 N)$ B. $O(N)$ C. $O(N^2)$ D. $\Theta(N^2 \log N)$

gate2016-2 data-structure linked-lists time-complexity normal

3.10.19 Linked Lists: GATE2017-1-08

https://gateoverflow.in/118711

Consider the C code fragment given below.

```
typedef struct node {
    int data;
    node* next;
} node;

void join(node* m, node* n) {
    node* p = n;
    while(p->next != NULL) {
        p = p->next;
    }
    p->next = m;
}
```

Assuming that m and n point to valid NULL-terminated linked lists, invocation of join will

A. append list m to the end of list n for all inputs.
B. either cause a null pointer dereference or append list m to the end of list n.
C. cause a null pointer dereference for all inputs.
D. append list n to the end of list m for all inputs.

gate2017-1 data-structure linked-lists normal

3.11 Priority Queue (1)

3.11.1 Priority Queue: GATE1997-4.7

https://gateoverflow.in/2248

A priority queue Q is used to implement a stack that stores characters. PUSH (C) is implemented as INSERT (Q, C, K) where K is an appropriate integer key chosen by the implementation. POP is implemented as DELETEMIN (Q). For a sequence of operations, the keys chosen are in

A. non-increasing order
B. non-decreasing order
C. strictly increasing order
D. strictly decreasing order

gate1997 data-structure stack normal priority-queue

© Copyright GATE Overflow. All rights reserved.

3.12 Queues (12)

3.12.1 Queues: GATE1992-09

https://gateoverflow.in/588

Suggest a data structure for representing a subset S of integers from 1 to n. Following operations on the set S are to be performed in constant time (independent of cardinality of S).

i. MEMBER (X) : Check whether X is in the set S or not
ii. FIND-ONE (S) : If S is not empty, return one element of the set S (any arbitrary element will do)
iii. ADD (X) : Add integer X to set S
ii. DELETE (X) : Delete integer X from S

Give pictorial examples of your data structure. Give routines for these operations in an English like language. You may assume that the data structure has been suitable initialized. Clearly state your assumptions regarding initialization.

gate1992 data-structure normal descriptive queues

3.12.2 Queues: GATE1994-26

https://gateoverflow.in/2522

A queue Q containing n items and an empty stack S are given. It is required to transfer all the items from the queue to the stack, so that the item at the front of queue is on the TOP of the stack, and the order of all other items are preserved. Show how this can be done in $O(n)$ time using only a constant amount of additional storage. Note that the only operations which can be performed on the queue and stack are Delete, Insert, Push and Pop. Do not assume any implementation of the queue or stack.

gate1994 data-structure queues stack normal

3.12.3 Queues: GATE1996-1.12

https://gateoverflow.in/2716

Consider the following statements:

i. First-in-first out types of computations are efficiently supported by STACKS.
ii. Implementing LISTS on linked lists is more efficient than implementing LISTS on an array for almost all the basic LIST operations.
iii. Implementing QUEUES on a circular array is more efficient than implementing QUEUES on a linear array with two indices.
iv. Last-in-first-out type of computations are efficiently supported by QUEUES.

A. (ii) and (iii) are true
B. (i) and (ii) are true
C. (iii) and (iv) are true
D. (ii) and (iv) are true

gate1996 data-structure easy queues stack linked-lists

3.12.4 Queues: GATE2001-2.16

https://gateoverflow.in/734

What is the minimum number of stacks of size n required to implement a queue of size n?

A. One B. Two C. Three D. Four

gate2001 data-structure easy stack queues

3.12.5 Queues: GATE2006-49

https://gateoverflow.in/1826

An implementation of a queue Q, using two stacks $S1$ and $S2$, is given below:

```
void insert (Q, x) {
    push (S1, x);
}
void delete (Q) {
    if (stack-empty(S2)) then
        if (stack-empty(S1)) then {
            print("Q is empty");
            return;
        }
        else while (!(stack-empty(S1))){
            x=pop(S1);
            push(S2,x);
        }
```

© Copyright GATE Overflow. All rights reserved.

```
    x=pop(S2);
}
```

let n *insert* and $m(\leq n)$ *delete* operations be performed in an arbitrary order on an empty queue Q. Let x and y be the number of *push* and *pop* operations performed respectively in the process. Which one of the following is true for all m and n?

A. $n+m \leq x < 2n$ and $2m \leq y \leq n+m$
B. $n+m \leq x < 2n$ and $2m \leq y \leq 2n$
C. $2m \leq x < 2n$ and $2m \leq y \leq n+m$
D. $2m \leq x < 2n$ and $2m \leq y \leq 2n$

gate2006 data-structure queues stack normal

3.12.6 Queues: GATE2007-IT-30

https://gateoverflow.in/3463

Suppose you are given an implementation of a queue of integers. The operations that can be performed on the queue are:

i. $isEmpty(Q)$ — returns true if the queue is empty, false otherwise.
ii. $delete(Q)$ — deletes the element at the front of the queue and returns its value.
iii. $insert(Q,i)$ — inserts the integer i at the rear of the queue.

Consider the following function:

```
void f (queue Q) {
int i ;
if (!isEmpty(Q)) {
   i = delete(Q);
   f(Q);
   insert(Q, i);
  }
}
```

What operation is performed by the above function f?

A. Leaves the queue Q unchanged
B. Reverses the order of the elements in the queue Q
C. Deletes the element at the front of the queue Q and inserts it at the rear keeping the other elements in the same order
D. Empties the queue Q

gate2007-it data-structure queues normal

3.12.7 Queues: GATE2012-35

https://gateoverflow.in/1756

Suppose a circular queue of capacity $(n-1)$ elements is implemented with an array of n elements. Assume that the insertion and deletion operations are carried out using REAR and FRONT as array index variables, respectively. Initially, $REAR = FRONT = 0$. The conditions to detect queue full and queue empty are

A. full: $(REAR+1) \mod n == FRONT$
empty: $REAR == FRONT$

B. full: $(REAR+1) \mod n == FRONT$
empty: $(FRONT+1) \mod n == REAR$

C. full: $REAR == FRONT$
empty: $(REAR+1) \mod n == FRONT$

D. full: $(FRONT+1) \mod n == REAR$
empty: $REAR == FRONT$

gate2012 data-structure queues normal

3.12.8 Queues: GATE2013-44

https://gateoverflow.in/61

Consider the following operation along with Enqueue and Dequeue operations on queues, where k is a global parameter.

```
MultiDequeue(Q){
    m = k
    while (Q is not empty) and (m > 0) {
        Dequeue(Q)
        m = m - 1
    }
}
```

What is the worst case time complexity of a sequence of n queue operations on an initially empty

© Copyright GATE Overflow. All rights reserved.

queue?

A. $\Theta(n)$ B. $\Theta(n+k)$ C. $\Theta(nk)$ D. $\Theta(n^2)$

gate2013 data-structure algorithms normal queues

3.12.9 Queues: GATE2016-1-10

https://gateoverflow.in/39667

A queue is implemented using an array such that ENQUEUE and DEQUEUE operations are performed efficiently. Which one of the following statements is **CORRECT** (n refers to the number of items in the queue) ?

A. Both operations can be performed in $O(1)$ time.
B. At most one operation can be performed in $O(1)$ time but the worst case time for the operation will be $\Omega(n)$.
C. The worst case time complexity for both operations will be $\Omega(n)$.
D. Worst case time complexity for both operations will be $\Omega(\log n)$

gate2016-1 data-structure queues normal

3.12.10 Queues: GATE2016-1-41

https://gateoverflow.in/39684

Let Q denote a queue containing sixteen numbers and S be an empty stack. $Head(Q)$ returns the element at the head of the queue Q without removing it from Q. Similarly $Top(S)$ returns the element at the top of S without removing it from S. Consider the algorithm given below.

```
while Q is not Empty do
  if S is Empty OR Top(S) ≤ Head (Q) then
     x:= Dequeue (Q);
     Push (S, x);
  else
     x:= Pop(S);
     Enqueue (Q, x);
  end
 end
```

The maximum possible number of iterations of the **while** loop in the algorithm is _____.

gate2016-1 data-structure queues difficult numerical-answers

3.12.11 Queues: GATE2017-2-13

https://gateoverflow.in/118253

A circular queue has been implemented using a singly linked list where each node consists of a value and a single pointer pointing to the next node. We maintain exactly two external pointers FRONT and REAR pointing to the front node and the rear node of the queue, respectively. Which of the following statements is/are CORRECT for such a circular queue, so that insertion and deletion operations can be performed in $O(1)$ time?

I. Next pointer of front node points to the rear node.
II. Next pointer of rear node points to the front node.

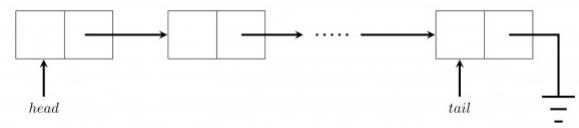
A. (I) only.
B. (II) only.
C. Both (I) and (II).
D. Neither (I) nor (II).

gate2017-2 data-structure queues

3.12.12 Queues: GATE2018-3

https://gateoverflow.in/204077

A queue is implemented using a non-circular singly linked list. The queue has a head pointer and a tail pointer, as shown in the figure. Let n denote the number of nodes in the queue. Let 'enqueue' be implemented by inserting a new node at the head, and 'dequeue' be implemented by deletion of a node from the tail.



Which one of the following is the time complexity of the most time-efficient implementation of 'enqueue' and 'dequeue, respectively, for this data structure?

A. $\Theta(1), \Theta(1)$
B. $\Theta(1), \Theta(n)$
C. $\Theta(n), \Theta(1)$
D. $\Theta(n), \Theta(n)$

© Copyright GATE Overflow. All rights reserved.

gate2018 algorithms data-structure queues normal linked-lists

3.13 Stack (16)

3.13.1 Stack: GATE1991-03,vii

https://gateoverflow.in/522

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

The following sequence of operations is performed on a stack:

PUSH (10), PUSH (20), POP, PUSH (10), PUSH (20), POP, POP, POP, PUSH (20), POP

The sequence of values popped out is

A. $20, 10, 20, 10, 20$
B. $20, 20, 10, 10, 20$
C. $10, 20, 20, 10, 20$
D. $20, 20, 10, 20, 10$

gate1991 data-structure stack easy

3.13.2 Stack: GATE1994-1.14

https://gateoverflow.in/2457

Which of the following permutations can be obtained in the output (in the same order) using a stack assuming that the input is the sequence $1, 2, 3, 4, 5$ in that order?

A. $3, 4, 5, 1, 2$
B. $3, 4, 5, 2, 1$
C. $1, 5, 2, 3, 4$
D. $5, 4, 3, 1, 2$

gate1994 data-structure stack normal

3.13.3 Stack: GATE1995-2.21

https://gateoverflow.in/2633

The postfix expression for the infix expression $A + B * (C + D)/F + D * E$ is:

A. $AB + CD + *F/D + E*$
B. $ABCD + *F/DE * ++$
C. $A * B + CD/F * DE + +$
D. $A + *BCD/F * DE + +$

gate1995 data-structure stack easy

3.13.4 Stack: GATE2000-13

https://gateoverflow.in/684

Suppose a stack implementation supports, in addition to PUSH and POP, an operation REVERSE, which reverses the order of the elements on the stack.

A. To implement a queue using the above stack implementation, show how to implement ENQUEUE using a single operation and DEQUEUE using a sequence of 3 operations.
B. The following post fix expression, containing single digit operands and arithmetic operators $+$ and $*$, is evaluated using a stack.
   $5\ 2 * 3\ 4 + 5\ 2 * *+$
   Show the contents of the stack
   i. After evaluating $5\ 2 * 3\ 4+$
   ii. After evaluating $5\ 2 * 3\ 4 + 5\ 2$
   iii. At the end of evaluation

gate2000 data-structure stack normal descriptive

3.13.5 Stack: GATE2003-64

https://gateoverflow.in/951

Let **S** be a stack of size $n \geq 1$. Starting with the empty stack, suppose we push the first n natural numbers in sequence, and then perform n pop operations. Assume that Push and Pop operations take X seconds each, and Y seconds elapse between the end of one such stack operation and the start of the next operation. For $m \geq 1$, define the stack-life of m as the time elapsed from the end of $Push(m)$ to the start of the pop operation that removes m from **S**. The average stack-life of an element of this stack is

A. $n(X + Y)$
B. $3Y + 2X$
C. $n(X + Y) - X$
D. $Y + 2X$

gate2003 data-structure stack normal

3.13.6 Stack: GATE2004-3

https://gateoverflow.in/1000

A single array $A[1..MAXSIZE]$ is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables $top1$ and $top2$ $(top < top2)$ point to the location of the topmost element in each of the stacks. If the

© Copyright GATE Overflow. All rights reserved.

space is to be used efficiently, the condition for "stack full" is

A. $(top1 = MAXSIZE/2)$ and $(top2 = MAXSIZE/2 + 1)$ B. $top1 + top2 = MAXSIZE$
C. $(top1 = MAXSIZE/2)$ or $(top2 = MAXSIZE)$ D. $top1 = top2 - 1$

gate2004 data-structure stack easy

3.13.7 Stack: GATE2004-38, ISRO2009-27

https://gateoverflow.in/1035

Assume that the operators $+, -, \times$ are left associative and $\hat{}$ is right associative. The order of precedence (from highest to lowest) is $\hat{}, \times, +, -$. The postfix expression corresponding to the infix expression $a + b \times c - d \hat{} e \hat{} f$ is

A. $abc \times + def \hat{} \hat{} -$ B. $abc \times + de \hat{} f \hat{} -$
C. $ab + c \times d - e \hat{} f \hat{}$ D. $- + a \times bc \hat{} \hat{} def$

gate2004 stack isro2009

3.13.8 Stack: GATE2004-5

https://gateoverflow.in/1002

The best data structure to check whether an arithmetic expression has balanced parentheses is a

A. queue B. stack C. tree D. list

gate2004 data-structure easy stack

3.13.9 Stack: GATE2004-IT-52

https://gateoverflow.in/3695

A program attempts to generate as many permutations as possible of the string, ' $abcd$' by pushing the characters a, b, c, d in the same order onto a stack, but it may pop off the top character at any time. Which one of the following strings CANNOT be generated using this program?

A. $abcd$ B. $dcba$ C. $cbad$ D. $cabd$

gate2004-it data-structure normal stack

3.13.10 Stack: GATE2005-IT-13

https://gateoverflow.in/3758

A function f defined on stacks of integers satisfies the following properties. $f(\emptyset) = 0$ and $f(push(S, i)) = max(f(S), 0) + i$ for all stacks S and integers i.

If a stack S contains the integers $2, -3, 2, -1, 2$ in order from bottom to top, what is $f(S)$?

A. 6 B. 4 C. 3 D. 2

gate2005-it data-structure stack normal

3.13.11 Stack: GATE2007-38, ISRO2016-27

https://gateoverflow.in/1236

The following postfix expression with single digit operands is evaluated using a stack:

$8\ 2\ 3\ \hat{}\ \qquad\qquad /$

Note that $\hat{}$ is the exponentiation operator. The top two elements of the stack after the first $*$ is evaluated are

A. $6, 1$ B. $5, 7$ C. $3, 2$ D. $1, 5$

gate2007 data-structure stack normal isro2016

3.13.12 Stack: GATE2007-IT-32

https://gateoverflow.in/3465

Consider the following C program:

```
#include <stdio.h>
        #define EOF -1
        void push (int); /* push the argument on the stack */
        int pop  (void); /* pop the top of the stack */
        void flagError ();
        int main ()
       {         int c, m, n, r;
                  while ((c = getchar ()) != EOF)
                 { if  (isdigit (c) )
                           push (c);
```

© Copyright GATE Overflow. All rights reserved.

```
                    else if ((c == '+') || (c == '*'))
                   {          m = pop ();
                               n = pop ();
                               r = (c == '+') ? n + m : n*m;
                               push (r);
                     }
                     else if (c != ' ')
                               flagError ();
            }
             printf("% c", pop ());
}
```

What is the output of the program for the following input?
$52 * 332 + *+$

A. 15 B. 25 C. 30 D. 150

gate2007-it stack normal

3.13.13 Stack: GATE2014-2-41

https://gateoverflow.in/2007

Suppose a stack implementation supports an instruction $REVERSE$, which reverses the order of elements on the stack, in addition to the $PUSH$ and POP instructions. Which one of the following statements is TRUE (*with respect to this modified stack*)?

A. A queue cannot be implemented using this stack.
B. A queue can be implemented where $ENQUEUE$ takes a single instruction and $DEQUEUE$ takes a sequence of two instructions.
C. A queue can be implemented where $ENQUEUE$ takes a sequence of three instructions and $DEQUEUE$ takes a single instruction.
D. A queue can be implemented where both $ENQUEUE$ and $DEQUEUE$ take a single instruction each.

gate2014-2 data-structure stack easy

3.13.14 Stack: GATE2015-2-38

https://gateoverflow.in/8164

Consider the C program below

```
#include <stdio.h>
int *A, stkTop;
int stkFunc (int opcode, int val)
{
    static int size=0, stkTop=0;
    switch (opcode) {
        case -1: size = val; break;
        case 0: if (stkTop < size ) A[stkTop++]=val; break;
        default: if (stkTop) return A[--stkTop];
    }
    return -1;
}
int main()
{
    int B[20]; A=B; stkTop = -1;
    stkFunc (-1, 10);
    stkFunc (0, 5);
    stkFunc (0, 10);
    printf ("%d\n", stkFunc(1, 0)+ stkFunc(1, 0));
}
```

The value printed by the above program is _____.

gate2015-2 data-structure stack easy numerical-answers

3.13.15 Stack: GATE2015-3-12

https://gateoverflow.in/8408

The result evaluating the postfix expression $10\,5 + 60\,6/ * 8-$ is

A. 284 B. 213 C. 142 D. 71

gate2015-3 data-structure stack normal

© Copyright GATE Overflow. All rights reserved.

3.13.16 Stack: TIFR2017-B-3

https://gateoverflow.in/95679

We have an implementation that supports the following operations on a stack (in the instructions below, s is the name of the stack).

- isempty(s) : returns True if s is empty, and False otherwise.
- top(s) : returns the top element of the stack, but does not pop the stack; returns null if the stack is empty.
- push(s,x) : places x on top of the stack.
- pop(s) : pops the stack; does nothing if s is empty.

Consider the following code:

```
pop_ray_pop(x):
    s=empty
    for i=1 to length(x):
        if (x[i] == '('):
            push(s, x[i])
        else:
            while (top(s)=='('):
                pop(s)
            end while
            push(s, ')')
        end if
    end for
    while not isempty(s):
        print top(s)
        pop(s)
    end while
```

What is the output of this program when

```
pop_ray_pop("((((()(((())((((")
```

is executed?

A. ((((B.))) ((((C.))) D. ((()())) E. ()()

tifr2017 data-structure stack

3.14 Trees (14)

3.14.1 Trees: GATE1990-13a

https://gateoverflow.in/86224

Consider the height-balanced tree T_t with values stored at only the leaf nodes, shown in Fig.4.

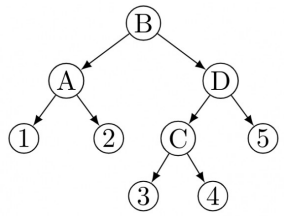


Fig.4

(i) Show how to merge to the tree, T_1 elements from tree T_2 shown in Fig.5 using node D of tree T_1.

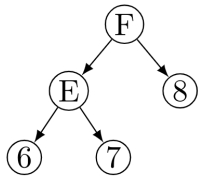


Fig.5

(ii) What is the time complexity of a merge operation of balanced trees T_1 and T_2 where T_1 and T_2 are of height h_1 and h_2 respectively, assuming that rotation schemes are given. Give reasons.

gate1990 descriptive data-structure trees

© Copyright GATE Overflow. All rights reserved.

3.14.2 Trees: GATE1992-02,vii

https://gateoverflow.in/562

Choose the correct alternatives (more than one may be correct) and write the corresponding letters only:

A $2-3$ tree is such that

a. All internal nodes have either 2 or 3 children
b. All paths from root to the leaves have the same length.

The number of internal nodes of a $2-3$ tree having 9 leaves could be

A. 4 B. 5 C. 6 D. 7

gate1992 trees data-structure normal

3.14.3 Trees: GATE1994-5

https://gateoverflow.in/2501

A $3-\text{ary}$ tree is a tree in which every internal node has exactly three children. Use induction to prove that the number of leaves in a $3-\text{ary}$ tree with n internal nodes is $2(n-1)$.

gate1994 data-structure trees proof

3.14.4 Trees: GATE1998-1.24

https://gateoverflow.in/1661

Which of the following statements is false?

A. A tree with a n nodes has $(n–1)$ edges
B. A labeled rooted binary tree can be uniquely constructed given its postorder and preorder traversal results.
C. A complete binary tree with n internal nodes has $(n+1)$ leaves.
D. The maximum number of nodes in a binary tree of height h is $2^{h+1}-1$

gate1998 data-structure trees normal

3.14.5 Trees: GATE1998-2.11

https://gateoverflow.in/1683

A complete n-ary tree is one in which every node has 0 or n sons. If x is the number of internal nodes of a complete n-ary tree, the number of leaves in it is given by

A. $x(n-1)+1$ B. $xn-1$ C. $xn+1$ D. $x(n+1)$

gate1998 data-structure trees normal

3.14.6 Trees: GATE1998-21

https://gateoverflow.in/1735

A. Derive a recurrence relation for the size of the smallest AVL tree with height h.
B. What is the size of the smallest AVL tree with height 8?

gate1998 data-structure trees descriptive numerical-answers

3.14.7 Trees: GATE2002-2.9

https://gateoverflow.in/839

The number of leaf nodes in a rooted tree of n nodes, with each node having 0 or 3 children is:

A. $\frac{n}{2}$ B. $\frac{(n-1)}{3}$ C. $\frac{(n-1)}{2}$ D. $\frac{(2n+1)}{3}$

gate2002 data-structure trees normal

3.14.8 Trees: GATE2004-6

https://gateoverflow.in/1003

Level order traversal of a rooted tree can be done by starting from the root and performing

A. preorder traversal
B. in-order traversal
C. depth first search
D. breadth first search

gate2004 data-structure trees easy

© Copyright GATE Overflow. All rights reserved.

3.14.9 Trees: GATE2005-36

https://gateoverflow.in/1372

In a complete k-ary tree, every internal node has exactly k children. The number of leaves in such a tree with n internal node is:

A. nk B. $(n-1)k+1$ C. $n(k-1)+1$ D. $n(k-1)$

gate2005 data-structure trees normal

3.14.10 Trees: GATE2007-43

https://gateoverflow.in/1241

A complete $n-ary$ tree is a tree in which each node has n children or no children. Let I be the number of internal nodes and L be the number of leaves in a complete $n-ary$ tree. If $L=41$ and $I=10$, what is the value of n?

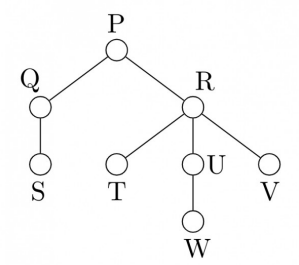
A. 3 B. 4 C. 5 D. 6

gate2007 data-structure trees normal

3.14.11 Trees: GATE2014-3-12

https://gateoverflow.in/2046

Consider the following rooted tree with the vertex labeled P as the root:



The order in which the nodes are visited during an in-order traversal of the tree is

A. $SQPTRWUV$ B. $SQPTUWRV$ C. $SQPTWUVR$ D. $SQPTRUWV$

gate2014-3 data-structure trees easy

3.14.12 Trees: GATE2014-3-41

https://gateoverflow.in/2075

Consider the pseudocode given below. The function $DoSomething()$ takes as argument a pointer to the root of an arbitrary tree represented by the $leftMostChild-rightSibling$ representation. Each node of the tree is of type $treeNode$.

```
typedef struct treeNode* treeptr;

struct treeNode
{
    treeptr leftMostChild, rightSibling;
};

int DoSomething (treeptr tree)
{
    int value=0;
    if (tree != NULL) {
        if (tree->leftMostChild == NULL)
            value = 1;
        else
        value = DoSomething(tree->leftMostChild);
        value = value + DoSomething(tree->rightSibling);
    }
    return(value);
}
```

When the pointer to the root of a tree is passed as the argument to $DoSomething$, the value returned by the function corresponds to the

A. number of internal nodes in the tree.
B. height of the tree.
C. number of nodes without a right sibling in the tree.
D. number of leaf nodes in the tree

gate2014-3 data-structure trees normal

© Copyright GATE Overflow. All rights reserved.

3.14.13 Trees: GATE2017-1-20

https://gateoverflow.in/118300

Let T be a tree with 10 vertices. The sum of the degrees of all the vertices in T is _____

gate2017-1 data-structure trees numerical-answers

3.14.14 Trees: TIFR2012-B-15

https://gateoverflow.in/25212

Let T be a tree of n nodes. Consider the following algorithm, that constructs a sequence of leaves $u_1, u_2 \ldots$. Let u_1 be some leaf of tree. Let u_2 be a leaf that is farthest from u_1. Let u_3 be the leaf that is farthest from u_2, and, in general, let u_{i+1} be a leaf of T that is farthest from u_i (if there are many choices for u_{i+1}, pick one arbitrarily). The algorithm stops when some u_i is visited again. What can u say about the distance between u_i and u_{i+1}, as $i = 1, 2, \ldots ?$

A. For some trees, the distance strictly reduces in each step.
B. For some trees, the distance increases initially and then decreases.
C. For all trees, the path connecting u_2 and u_3 is a longest path in the tree.
D. For some trees, the distance reduces initially, but then stays constant.
E. For the same tree, the distance between the last two vertices visited can be different, based on the choice of the first leaf u_1.

tifr2012 data-structure trees

© Copyright GATE Overflow. All rights reserved.

4 Programming and DS: Programming (118)

Programming in C. Recursion.

Mark Distribution in Previous GATE

Year	2019	2018	2017-1	2017-2	2016-1	2016-2	Minimum	Average	Maximum
1 Mark Count	2	2	1	2	2	1	1	1.7	2
2 Marks Count	3	3	4	4	2	2	2	3	4
Total Marks	8	8	9	10	6	5	**5**	**8**	**10**

4.1 Aliasing (1)

4.1.1 Aliasing: GATE2000-1.16

https://gateoverflow.in/639

Aliasing in the context of programming languages refers to

A. multiple variables having the same memory location
B. multiple variables having the same value
C. multiple variables having the same identifier
D. multiple uses of the same variable

gate2000 programming easy aliasing

4.2 Goto (2)

4.2.1 Goto: GATE1989-3-i

https://gateoverflow.in/87095

An unrestricted use of the "go to" statement is harmful because of which of the following reason (s):

A. It makes it more difficult to verify programs.
B. It makes programs more inefficient.
C. It makes it more difficult to modify existing programs.
D. It results in the compiler generating longer machine code.

gate1989 normal programming goto

4.2.2 Goto: GATE1994-1.5

https://gateoverflow.in/2442

An unrestricted use of the "*goto*" statement is harmful because

A. it makes it more difficult to verify programs
B. it increases the running time of the programs
C. it increases the memory required for the programs
D. it results in the compiler generating longer machine code

gate1994 programming easy goto

4.3 Identify Function (4)

4.3.1 Identify Function: GATE1995-3

https://gateoverflow.in/2639

Consider the following high level programming segment. Give the contents of the memory locations for variables W, X, Y and Z after the execution of the program segment. The values of the variables A and B are $5CH$ and $92H$, respectively. Also indicate error conditions if any.

```
var
    A, B, W, X, Y   :unsigned byte;
    Z               :unsigned integer, (each integer is represented by two bytes)
begin
    X               :=A+B
    Y               :=abs(A-B);
    W               :=A-B
    Z               :=A*B
end;
```

gate1995 programming identify-function descriptive

© Copyright GATE Overflow. All rights reserved.

4.3.2 Identify Function: GATE1998-2.13

https://gateoverflow.in/1685

What is the result of the following program?

```
    program side-effect (input, output);
    var x, result: integer;
    function f (var x:integer:integer;
    begin
        x:x+1;f:=x;
    end
begin
    x:=5;
    result:=f(x)*f(x);
    writeln(result);
    end
```

A. 5 B. 25 C. 36 D. 42

gate1998 programming normal identify-function

4.3.3 Identify Function: GATE2004-IT-15

https://gateoverflow.in/3656

Let x be an integer which can take a value of 0 or 1. The statement

```
if (x == 0) x = 1; else x = 0;
```

is equivalent to which one of the following ?

A. $x = 1 + x;$ B. $x = 1 - x;$ C. $x = x - 1;$ D. $x = 1\%x;$

gate2004-it programming easy identify-function

4.3.4 Identify Function: GATE2017-2-43

https://gateoverflow.in/118388

Consider the following snippet of a C program. Assume that swap $(\&x, \&y)$ exchanges the content of x and y:

```
int main () {
    int array[] = {3, 5, 1, 4, 6, 2};
    int done =0;
    int i;
    while (done==0) {
        done =1;
        for (i=0; i<=4; i++) {
            if (array[i] < array[i+1]) {
                swap(&array[i], &array[i+1]);
                done=0;
            }
        }
        for (i=5; i>=1; i--) {
            if (array[i] > array[i-1]) {
                swap(&array[i], &array[i-1]);
                done =0;
            }
        }
    }
    printf("%d", array[3]);
}
```

The output of the program is _____

gate2017-2 programming algorithms numerical-answers identify-function

4.4 Loop Invariants (12)

4.4.1 Loop Invariants: GATE1987-7a

https://gateoverflow.in/82425

List the invariant assertions at points A, B, C, D and E in program given below:

```
Program division (input, output)
Const
    dividend = 81;
    divisor = 9;
Var  remainder, quotient:interger
begin
    (*(dividend >= 0) AND (divisor > 0)*)
    remainder := dividend;
    quotient := 9;
```

© Copyright GATE Overflow. All rights reserved.

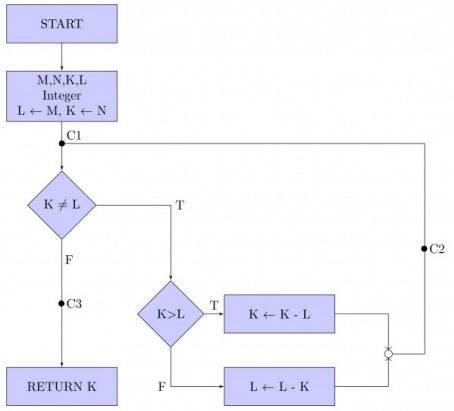
```
    (*A*)
While (remainder >= 0) do
begin (*B*)
    quotient := quotient + 1;
    remainder := remainder - divisor;
    (*C*)
end;
    (*D*)
    quotient := quotient - 1;
    remainder := remainder + divisor;
    (*E*)
end
```

gate1987 programming loop-invariants

4.4.2 Loop Invariants: GATE1988-6ii

https://gateoverflow.in/94364

Below figure is the flow-chart corresponding to a program to calculate the gcd of two integers, M and N respectively, $(M, N > 0)$. Use assertions at the cut point C_1, C_2 and C_3 to prove that the flow-chart is correct.



gate1988 normal descriptive loop-invariants

4.4.3 Loop Invariants: GATE1988-8ii

https://gateoverflow.in/94379

Consider the two program segments below:

a.
```
for
    i:=1 to f(x) by 1 do
    S
end
```

b.
```
i:=1;
While   i<=f(x) do
        S
        i:=i+1
end
```

Under what conditions are these two programs equivalent? Treat S as any sequence of statement and f as a function.

gate1988 programming descriptive loop-invariants

4.4.4 Loop Invariants: GATE1991-1,vi

https://gateoverflow.in/504

Consider the following PASCAL program segment:

```
if i mod 2 = 0 then
    while i >= 0 do
    begin
        i := i div 2;
        if i mod 2 < > 0  then i := i - 1;
        else i := i - 2;
    end;
```

An appropriate loop-invariant for the while-loop is _____

gate1991 programming loop-invariants normal

© Copyright GATE Overflow. All rights reserved.

4.4.5 Loop Invariants: GATE2004-32

https://gateoverflow.in/1029

Consider the following program fragment for reversing the digits in a given integer to obtain a new integer.
Let $n = d_1\, d_2 \ldots d_m$.

```
int n, rev;
rev = 0;
while(n > 0) {
    rev = rev * 10 + n%10;
    n = n/10;
}
```

The loop invariant condition at the end of the i^{th} iteration is:

A. $n = d_1\, d_2 \ldots d_{m-i}$ **and** $\text{rev} = d_m\, d_{m-1} \ldots d_{m-i+1}$
B. $n = d_{m-i+1} \ldots d_{m-1}\, d_m$ **or** $\text{rev} = d_{m-i} \ldots d_2\, d_1$
C. $n \neq \text{rev}$
D. $n = d_1\, d_2 \ldots d_m$ **or** $\text{rev} = d_m \ldots d_2\, d_1$

gate2004 programming loop-invariants normal

4.4.6 Loop Invariants: GATE2015-1-33

https://gateoverflow.in/8276

Consider the following pseudo code, where x and y are positive integers.

```
begin
    q := 0
    r := x
   while r ≥ y do
      begin
      r := r - y
      q := q + 1
    end
end
```

The post condition that needs to be satisfied after the program terminates is

A. $\{r = qx + y \wedge r < y\}$
B. $\{x = qy + r \wedge r < y\}$
C. $\{y = qx + r \wedge 0 < r < y\}$
D. $\{q + 1 < r - y \wedge y > 0\}$

gate2015-1 programming loop-invariants normal

4.4.7 Loop Invariants: GATE2016-2-35

https://gateoverflow.in/39578

The following function computes X^Y for positive integers X and Y.

```
int exp (int X, int Y) {
     int res =1, a = X, b = Y;

     while (b != 0) {
         if (b % 2 == 0) {a = a * a; b = b/2; }
         else            {res = res * a; b = b - 1; }
     }
     return res;
}
```

Which one of the following conditions is TRUE before every iteration of the loop?

A. $X^Y = a^b$
B. $(res * a)^Y = (res * X)^b$
C. $X^Y = res * a^b$
D. $X^Y = (res * a)^b$

gate2016-2 programming loop-invariants normal

4.4.8 Loop Invariants: GATE2017-2-37

https://gateoverflow.in/118381

Consider the C program fragment below which is meant to divide x by y using repeated subtractions. The variables x, y, q and r are all unsigned int.

```
while (r >= y) {
    r=r-y;
```

© Copyright GATE Overflow. All rights reserved.

```
    q=q+1;
}
```

Which of the following conditions on the variables x, y, q and r before the execution of the fragment will ensure that the loop terminated in a state satisfying the condition $x == (y * q + r)$?

A. $(q == r)\ \&\&\ (r == 0)$
B. $(x > 0)\ \&\&\ (r == x)\ \&\&\ (y > 0)$
C. $(q == 0)\ \&\&\ (r == x)\ \&\&\ (y > 0)$
D. $(q == 0)\ \&\&\ (y > 0)$

gate2017-2 programming loop-invariants

4.4.9 Loop Invariants: TIFR2010-B-30

https://gateoverflow.in/19042

Consider the following program for summing the entries of the array b: array $[0..N-1]$ of integers, where N is a positive integer. (The symbol '<>' denotes 'not equal to').

```
var
    i, s: integer;
Program
    i:= 0;
    s:= 0;
[*] while i <> N do
        s := s + b[i];
        i := i + 1;
    od
```

Which of the following gives the invariant that holds at the beginning of each loop, that is, each time the program arrives at point $[*]$?

A. $s = \sum_{j=0}^{N} b[j]\ \&\ 0 \leq i \leq N$
B. $s = \sum_{j=0}^{i-1} b[j]\ \&\ 0 \leq i < N$
C. $s = \sum_{j=0}^{i} b[j]\ \&\ 0 < i \leq N$
D. $s = \sum_{j=1}^{N} b[j]\ \&\ 0 \leq i < N$
E. $s = \sum_{j=0}^{i-1} b[j]\ \&\ 0 \leq i \leq N$

tifr2010 programming loop-invariants

4.4.10 Loop Invariants: TIFR2010-B-37

https://gateoverflow.in/19251

Consider the program where a, b are integers with $b > 0$.

```
x:=a; y:=b; z:=0;
while y > 0 do
    if odd (x) then
        z:= z + x;
        y:= y - 1;
    else y:= y % 2;
        x:= 2 * x;
    fi
```

Invariant of the loop is a condition which is true before and after every iteration of the loop. In the above program the loop invariant is given by

$$0 \leq y \text{ and } z + x * y = a * b$$

Which of the following is true of the program?

A. The program will not terminate for some values of a, b.
B. The program will terminate with $z = 2^b$
C. The program will terminate with $z = a * b$.
D. The program will not terminate for some values of a, b but when it does terminate, the condition $z = a * b$ will hold.

© Copyright GATE Overflow. All rights reserved.

E. The program will terminate with $z = a^b$

tifr2010 programming loop-invariants

4.4.11 Loop Invariants: TIFR2017-B-5

https://gateoverflow.in/95683

Consider the following psuedocode fragment, where y is an integer that has been initialized.

```
int i=1
int j=1
while (i<10):
    j=j*i
    i=i+1
    if (i==y):
        break
    end if
end while
```

Consider the following statements:

i. $(i == 10)$ or $(i == y)$
ii. If $y > 10$, then $i == 10$
iii. If $j = 6$, then $y == 4$

Which of the above statements is/are TRUE at the end of the while loop? Choose from the following options.

A. i only B. iii only C. ii and iii only D. i, ii, and iii E. None of the above

tifr2017 programming loop-invariants

4.4.12 Loop Invariants: TIFR2019-B-9

https://gateoverflow.in/280486

Consider the following program fragment:

```
var x, y: integer;
x := 1; y := 0;
while y < x do
begin
   x := 2*x;
   y := y+1
end;
```

For the above fragment , which of the following is a loop invariant ?

A. $x = y + 1$
B. $x = (y + 1)^2$
C. $x = (y + 1)2^y$
D. $x = 2^y$
E. None of the above, since the loop does not terminate

tifr2019 programming loop-invariants

4.5 Parameter Passing (7)

4.5.1 Parameter Passing: GATE1999-15

https://gateoverflow.in/1514

What will be the output of the following program assuming that parameter passing is

i. call by value
ii. call by reference
iii. call by copy restore

```
procedure P{x, y, z};
begin
y:y+1;
z: x+x
end;
begin
a:= b:= 3;
P(a+b, a, a);
Print(a)
end.
```

gate1999 programming parameter-passing normal

© Copyright GATE Overflow. All rights reserved.

4.5.2 Parameter Passing: GATE2001-2.17 | UGCNET-AUG2016-III-21

https://gateoverflow.in/735

What is printed by the print statements in the program $P1$ assuming call by reference parameter passing?

```
Program P1()
{
    x = 10;
    y = 3;
    func1(y,x,x);
    print x;
    print y;
}

func1(x,y,z)
{
    y = y + 4;
    z = x + y + z
}
```

A. $10, 3$ B. $31, 3$ C. $27, 7$ D. None of the above

gate2001 programming compiler-design parameter-passing normal runtime-environments ugcnetaug2016iii

4.5.3 Parameter Passing: GATE2003-73

https://gateoverflow.in/960

The following program fragment is written in a programming language that allows global variables and does not allow nested declarations of functions.

```
global int i=100, j=5;
void P(x) {
 int i=10;
    print(x+10);
    i=200;
    j=20;
    print (x);
}
main() {P(i+j);}
```

If the programming language uses static scoping and call by need parameter passing mechanism, the values printed by the above program are:

A. $115, 220$ B. $25, 220$ C. $25, 15$ D. $115, 105$

gate2003 compiler-design normal runtime-environments parameter-passing

4.5.4 Parameter Passing: GATE2013-42

https://gateoverflow.in/60

What is the return value of $f(p,p)$, if the value of p is initialized to 5 before the call? Note that the first parameter is passed by reference, whereas the second parameter is passed by value.

```
int f (int &x, int c) {
       c = c - 1;
       if (c==0) return 1;
       x = x + 1;
       return f(x,c) * x;
}
```

gate2013 compiler-design normal marks-to-all numerical-answers parameter-passing runtime-environments

4.5.5 Parameter Passing: GATE2018-29

https://gateoverflow.in/204103

```
#include<stdio.h>
void fun1(char* s1, char* s2){
    char* temp;
    temp = s1;
    s1 = s2;
    s2 = temp;
}
void fun2(char** s1, char** s2){
    char* temp;
    temp = *s1;
    *s1 = *s2;
    *s2 = temp;
}
int main(){
```

© Copyright GATE Overflow. All rights reserved.

```
    char *str1="Hi", *str2 = "Bye";
    fun1(str1, str2); printf("%s %s", str1, str2);
    fun2(&str1, &str2); printf("%s %s", str1, str2);
    return 0;
}
```

The output of the program above is:

A. Hi Bye Bye Hi
B. Hi Bye Hi Bye
C. Bye Hi Hi Bye
D. Bye Hi Bye Hi

gate2018 programming-in-c pointers parameter-passing normal programming

4.5.6 Parameter Passing: TIFR2011-B-32

https://gateoverflow.in/20619

Various parameter passing mechanisms have been in used in different programming languages. Which of the following statements is true?

a. Call by value result is used in language Ada.
b. Call by value result is the same as call by name.
c. Call by value is the most robust.
d. Call by reference is the same as call by name.
e. Call by name is the most efficient.

tifr2011 programming parameter-passing

4.5.7 Parameter Passing: TIFR2019-B-8

https://gateoverflow.in/280487

Consider the following program fragment:

```
var a,b : integer;
procedure G(c,d: integer);
begin
    c:=c-d;
    d:=c+d;
    c:=d-c
end;
a:=2;
b:=3;
G(a,b);
```

If both parameters to G are passed by reference, what are the values of a and b at the end of the above program fragment ?

A. $a = 0$ and $b = 2$ B. $a = 3$ and $b = 2$ C. $a = 2$ and $b = 3$ D. $a = 1$ and $b = 5$ E. None of the above

tifr2019 programming parameter-passing

4.6 Programming Constructs (1)

4.6.1 Programming Constructs: GATE1999-2.5

https://gateoverflow.in/1483

Given the programming constructs

i. assignment
ii. for loops where the loop parameter cannot be changed within the loop
iii. if-then-else
iv. forward go to
v. arbitrary go to
vi. non-recursive procedure call
vii. recursive procedure/function call
viii. repeat loop,

which constructs will you not include in a programming language such that it should be possible to program the terminates (i.e., halting) function in the same programming language

A. (ii), (iii), (iv)
B. (v), (vii), (viii)
C. (vi), (vii), (viii)
D. (iii), (vii), (viii)

gate1999 programming normal programming-constructs

4.7 Programming In C (69)

© Copyright GATE Overflow. All rights reserved.

4.7.1 Programming In C: GATE2000-1.11

https://gateoverflow.in/634

The following C declarations:

```
struct node {
    int i:
    float j;
 };
 struct node *s[10];
```

define s to be:

A. An array, each element of which is a pointer to a structure of type node
B. A structure of 2 fields, each field being a pointer to an array of 10 elements
C. A structure of 3 fields: an integer, a float, and an array of 10 elements
D. An array, each element of which is a structure of type node

gate2000 programming programming-in-c easy

4.7.2 Programming In C: GATE2000-1.12

https://gateoverflow.in/635

The most appropriate matching for the following pairs

$X:$	$\text{m} = \text{malloc}(5); \text{m} = \text{NULL};$	$1:$	using dangling pointers
$Y:$	$\text{free}(\text{n}); \text{n} \text{->} \text{value} = 5;$	$2:$	using uninitialized pointers
$Z:$	$\text{char} \text{*p}, \text{*p} = \text{'a'};$	$3:$	lost memory

is:

A. $X-1\ Y-3\ Z-2$
B. $X-2\ Y-1\ Z-3$
C. $X-3\ Y-2\ Z-1$
D. $X-3\ Y-1\ Z-2$

gate2000 programming programming-in-c normal

4.7.3 Programming In C: GATE2000-1.17, ISRO2015-79

https://gateoverflow.in/640

Consider the following C declaration:

```
struct (
    short x[5];
    union {
        float y;
        long z;
    } u;
 )t;
```

Assume that the objects of the type short, float and long occupy 2 bytes, 4 bytes and 8 bytes, respectively. The memory requirement for variable t, ignoring alignment consideration, is:

A. 22 bytes
B. 14 bytes
C. 18 bytes
D. 10 bytes

gate2000 programming programming-in-c easy isro2015

4.7.4 Programming In C: GATE2000-2.20

https://gateoverflow.in/667

The value of j at the end of the execution of the following C program:

```
int incr (int i)
{
    static int count = 0;
    count = count + i;
    return (count);
}
main () {
    int i, j;
    for (i = 0; i <= 4; i++)
       j = incr (i);
}
```

is:

A. 10
B. 4
C. 6
D. 7

© Copyright GATE Overflow. All rights reserved.

gate2000 programming programming-in-c easy

4.7.5 Programming In C: GATE2001-2.18

https://gateoverflow.in/736

Consider the following three C functions:

$[P1]$

```
int *g(void)
{
    int x = 10;
    return (&x);
}
```

$[P2]$

```
int *g(void)
{
    int *px;
    *px = 10;
    return px;
}
```

$[P3]$

```
int *g(void)
{
    int *px;
    px = (int*) malloc (sizeof(int));
    *px = 10;
    return px;
}
```

Which of the above three functions are likely to cause problems with pointers?

A. Only $P3$ B. Only $P1$ and $P3$ C. Only $P1$ and $P2$ D. $P1, P2$ and $P3$

gate2001 programming programming-in-c normal

4.7.6 Programming In C: GATE2002-1.17

https://gateoverflow.in/822

In the C language:

A. At most one activation record exists between the current activation record and the activation record for the main
B. The number of activation records between the current activation record and the activation records from the main depends on the actual function calling sequence.
C. The visibility of global variables depends on the actual function calling sequence
D. Recursion requires the activation record for the recursive function to be saved in a different stack before the recursive function can be called.

gate2002 programming programming-in-c easy descriptive

4.7.7 Programming In C: GATE2002-2.18

https://gateoverflow.in/848

The C language is:

A. A context free language
B. A context sensitive language
C. A regular language
D. Parsable fully only by a Turing machine

gate2002 programming programming-in-c normal

4.7.8 Programming In C: GATE2002-2.8

https://gateoverflow.in/838

Consider the following declaration of a two-dimensional array in C:

char $a[100][100]$;

Assuming that the main memory is byte-addressable and that the array is stored starting from memory address 0, the address of $a[40][50]$ is:

A. 4040 B. 4050 C. 5040 D. 5050

gate2002 programming-in-c programming easy

© Copyright GATE Overflow. All rights reserved.

4.7.9 Programming In C: GATE2003-2

https://gateoverflow.in/893

Assume the following C variable declaration:

```
int *A[10], B[10][10];
```

Of the following expressions:

I. $A[2]$
II. $A[2][3]$
III. $B[1]$
IV. $B[2][3]$

which will not give compile-time errors if used as left hand sides of assignment statements in a C program?

A. I, II, and IV only B. II, III, and IV only C. II and IV only D. IV only

gate2003 programming programming-in-c easy

4.7.10 Programming In C: GATE2003-89

https://gateoverflow.in/972

Consider the C program shown below:

```
#include<stdio.h>
#define print(x) printf("%d", x)

int x;
void Q(int z)
{
        z+=x;
        print(z);
}

void P(int *y)
{
        int x = *y + 2;
        Q(x);
        *y = x - 1;
        print(x);
}
main(void) {
        x = 5;
        P(&x);
        print(x);
}
```

The output of this program is:

A. 12 7 6 B. 22 12 11 C. 14 6 6 D. 7 6 6

gate2003 programming programming-in-c normal

4.7.11 Programming In C: GATE2004-33

https://gateoverflow.in/1030

Consider the following C program segment:

```
char p[20]; int i;
char* s = "string";
int length = strlen(s);
for(i = 0; i < length; i++)
    p[i] = s[length-i];
printf("%s", p);
```

The output of the program is:

A. gnirts B. string C. gnirt D. no output is printed

gate2004 programming programming-in-c easy

4.7.12 Programming In C: GATE2004-IT-58

https://gateoverflow.in/3701

Consider the following C program which is supposed to compute the transpose of a given 4×4 matrix M. Note that, there is an X in the program which indicates some missing statements. Choose the correct option to replace X in the program.

© Copyright GATE Overflow. All rights reserved.

```
#include<stdio.h>
#define ROW 4
#define COL 4
int M[ROW][COL] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16};
main()
{
    int i, j, t;
    for (i = 0; i < 4; ++i)
    {
        X
    }
    for (1 = 0; i < 4; ++i)
        for (j = 0; j < 4; ++j)
            printf ("%d", M[i][j]);
}
```

A.
```
for(j = 0; j < 4; ++j){
     t = M[i][j];
     M[i][j] = M[j][i];
     M[j][i] = t;
}
```

B.
```
for(j = 0; j < 4; ++j){
     M[i][j] = t;
     t = M[j][i];
     M[j][i] = M[i][j];
}
```

C.
```
for(j = i; j < 4; ++j){
     t = M[i][j];
     M[i][j] = M[j][i];
     M[j][i] = t;
}
```

D.
```
for(j = i; j < 4; ++j){
     M[i][j] = t;
     t = M[j][i];
     M[j][i] = M[i][j];
}
```

gate2004-it programming easy programming-in-c

4.7.13 Programming In C: GATE2004-IT-59

https://gateoverflow.in/3702

What is the output of the following program?

```
#include<stdio.h>
int funcf (int x);
int funcg (int y);
main ()
{
    int x = 5, y = 10, count;
    for (count = 1; count <= 2; ++count) {
        y += funcf(x) + funcg(x);
        printf ("%d", y);
    }
}
funcf (int x) {
    int y;
    y = funcg(x);
    return (y);
}
funcg (int x) {
    static int y = 10;
    y += 1;
    return (y + x);
}
```

A. 43 80 B. 42 74 C. 33 37 D. 32 32

gate2004-it programming programming-in-c normal

4.7.14 Programming In C: GATE2004-IT-60

https://gateoverflow.in/3703

Choose the correct option to fill the ?1 and ?2 so that the program prints an input string in reverse order. Assume that the input string is terminated by a new line character.

```
#include <stdio.h>
void wrt_it (void);
int main (void)
{
    printf("Enter Text");
    printf ("\n");
    wrt_it();
```

© Copyright GATE Overflow. All rights reserved.

```
    printf ("\n");
    return 0;
}
void wrt_it (void)
{
    int c;
    if (?1)
        wrt_it();
    ?2
}
```

A. ?1 is $getchar() != \text{'\textbackslash n'}$
   ?2 is $getchar(c);$
B. ?1 is $(c = getchar()); != \text{'\textbackslash n'}$
   ?2 is $getchar(c);$
C. ?1 is $c != \text{'\textbackslash n'}$
   ?2 is $putchar(c);$
D. ?1 is $(c = getchar()) != \text{'\textbackslash n'}$
   ?2 is $putchar(c);$

gate2004-it programming programming-in-c normal

4.7.15 Programming In C: GATE2004-IT-61

https://gateoverflow.in/3704

Consider the following C program:

```
#include <stdio.h>
typedef struct {
    char *a;
    char *b;
    } t;
void f1 (t s);
void f2 (t *p);
main()
{
    static t s = {"A", "B"};
    printf ("%s %s\n", s.a, s.b);
    f1(s);
    printf ("%s %s\n", s.a, s.b);
    f2(&s);
}
void f1 (t s)
{
    s.a = "U";
    s.b = "V";
    printf ("%s %s\n", s.a, s.b);
    return;
}
void f2(t *p)
{
    p -> a  = "V";
    p -> b = "W";
    printf("%s %s\n", p -> a, p -> b);
    return;
}
```

What is the output generated by the program ?

A. $A\ B$
   $U\ V$
   $V\ W$
   $V\ W$
B. $A\ B$
   $U\ V$
   $A\ B$
   $V\ W$
C. $A\ B$
   $U\ V$
   $U\ V$
   $V\ W$
D. $A\ B$
   $U\ V$
   $V\ W$
   $U\ V$

gate2004-it programming programming-in-c normal

4.7.16 Programming In C: GATE2005-1, ISRO2017-55

https://gateoverflow.in/1343

What does the following C-statement declare?

```
int (*f) (int * );
```

© Copyright GATE Overflow. All rights reserved.

A. A function that takes an integer pointer as argument and returns an integer
B. A function that takes an integer as argument and returns an integer pointer
C. A pointer to a function that takes an integer pointer as argument and returns an integer
D. A function that takes an integer pointer as argument and returns a function pointer

gate2005 programming programming-in-c easy isro2017

4.7.17 Programming In C: GATE2005-32

https://gateoverflow.in/1368

Consider the following C program:

```
double foo (double);  /* Line 1 */
int main() {
    double da, db;
    //input da
    db = foo(da);
}
double foo (double a) {
    return a;
}
```

The above code compiled without any error or warning. If Line 1 is deleted, the above code will show:

A. no compile warning or error
B. some compiler-warnings not leading to unintended results
C. some compiler-warnings due to type-mismatch eventually leading to unintended results
D. compiler errors

gate2005 programming programming-in-c compiler-design easy

4.7.18 Programming In C: GATE2005-IT-53

https://gateoverflow.in/3814

The following$C$ function takes two ASCII strings and determines whether one is an anagram of the other. An anagram of a string s is a string obtained by permuting the letters in s.

```
int anagram (char *a, char *b) {
    int count [128], j;
    for (j = 0;  j < 128; j++) count[j] = 0;
    j = 0;
    while (a[j] && b[j]) {
        A;
        B;
    }
    for (j = 0; j < 128; j++) if (count [j]) return 0;
    return 1;
}
```

Choose the correct alternative for statements A and B.

A. $A : count[a[j]]++ and\ B : count[b[j]]--$
B. $A : count[a[j]]++ and\ B : count[b[j]]++$
C. $A : count[a[j++]]++ and\ B : count[b[j]]--$
D. $A : count[a[j]]++ and\ B : count[b[j++]]--$

gate2005-it programming normal programming-in-c

4.7.19 Programming In C: GATE2005-IT-58

https://gateoverflow.in/3819

Let a be an array containing n integers in increasing order. The following algorithm determines whether there are two distinct numbers in the array whose difference is a specified number $S > 0$.

```
i = 0; j = 1;
while (j < n ){
        if (E) j++;
        else if (a[j] - a[i] == S) break;
        else i++;
}
if (j < n) printf("yes") else printf ("no");
```

© Copyright GATE Overflow. All rights reserved.

Choose the correct expression for E.

A. $a[j]-a[i]>S$
B. $a[j]-a[i]<S$
C. $a[i]-a[j]<S$
D. $a[i]-a[j]>S$

gate2005-it programming normal programming-in-c

4.7.20 Programming In C: GATE2006-57

https://gateoverflow.in/1835

Consider this C code to swap two integers and these five statements: the code

```
void swap (int *px, int *py)
   {
      *px = *px - *py;
      *py = *px + *py;
       *px = *py - *px;
   }
```

S1: will generate a compilation error

S2: may generate a segmentation fault at runtime depending on the arguments passed

S3: correctly implements the swap procedure for all input pointers referring to integers stored in memory locations accessible to the process

S4: implements the swap procedure correctly for some but not all valid input pointers

S5: may add or subtract integers and pointers

A. S1 B. S2 and S3 C. S2 and S4 D. S2 and S5

gate2006 programming programming-in-c normal

4.7.21 Programming In C: GATE2006-IT-49

https://gateoverflow.in/3592

Which one of the choices given below would be printed when the following program is executed ?

```
#include <stdio.h>
struct test {
              int i;
              char *c;
}st[] = {5, "become", 4, "better", 6, "jungle", 8, "ancestor", 7, "brother"};
main ()
{
    struct test *p = st;
    p += 1;
    ++p -> c;
    printf("%s,", p++ -> c);
    printf("%c,", *++p -> c);
    printf("%d,", p[0].i);
    printf("%s \n", p -> c);
}
```

A. jungle, n, 8, nclastor
B. etter, u, 6, ungle
C. cetter, k, 6, jungle
D. etter, u, 8, ncestor

gate2006-it programming programming-in-c normal

4.7.22 Programming In C: GATE2006-IT-50

https://gateoverflow.in/3593

Which one of the choices given below would be printed when the following program is executed?

```
#include <stdio.h>
void swap (int *x, int *y)
{
    static int *temp;
    temp = x;
    x = y;
    y = temp;
}
void printab ()
{
    static int i, a = -3, b = -6;
    i = 0;
    while (i <= 4)
    {
        if ((i++)%2 == 1) continue;
        a = a + i;
        b = b + i;
    }
    swap (&a, &b);
```

© Copyright GATE Overflow. All rights reserved.

```
    printf("a =  %d, b = %d\n", a, b);
}
main()
{
    printab();
    printab();
}
```

A. $a=0, b=3$
   $a=0, b=3$

B. $a=3, b=0$
   $a=12, b=9$

C. $a=3, b=6$
   $a=3, b=6$

D. $a=6, b=3$
   $a=15, b=12$

gate2006-it programming programming-in-c normal

4.7.23 Programming In C: GATE2006-IT-51

https://gateoverflow.in/3594

Which one of the choices given below would be printed when the following program is executed?

```
#include <stdio.h>
int a1[] = {6, 7, 8, 18, 34, 67};
int a2[] = {23, 56, 28, 29};
int a3[] = {-12, 27, -31};
int *x[] = {a1, a2, a3};
void print(int *a[])
{
            printf("%d,", a[0][2]);
            printf("%d,", *a[2]);
            printf("%d,", *++a[0]);
            printf("%d,", *(++a)[0]);
            printf("%d\n", a[-1][+1]);
}
main()
{
             print(x);
}
```

A. $8, -12, 7, 23, 8$

B. $8, 8, 7, 23, 7$

C. $-12, -12, 27, -31, 23$

D. $-12, -12, 27, -31, 56$

gate2006-it programming programming-in-c normal

4.7.24 Programming In C: GATE2007-IT-31

https://gateoverflow.in/3464

Consider the C program given below :

```
#include <stdio.h>
int main ()    {
    int sum = 0, maxsum = 0,  i,  n = 6;
    int a [] = {2, -2, -1, 3, 4, 2};
    for (i = 0; i < n; i++)    {
            if (i == 0 || a [i]  < 0  || a [i] < a [i - 1])  {
                    if (sum > maxsum) maxsum = sum;
                    sum = (a [i] > 0) ? a [i] : 0;
            }
            else sum += a [i];
    }
    if (sum > maxsum) maxsum = sum ;
    printf ("%d\n", maxsum);

}
```

What is the value printed out when this program is executed?

A. 9 B. 8 C. 7 D. 6

gate2007-it programming programming-in-c normal

4.7.25 Programming In C: GATE2008-18

https://gateoverflow.in/416

Which combination of the integer variables x, y and z makes the variable a get the value 4 in the following expression?

$$a = (x > y)?((x > z)?x : z) : ((y > z)?y : z)$$

A. $x=3, y=4, z=2$

B. $x=6, y=5, z=3$

C. $x=6, y=3, z=5$

D. $x=5, y=4, z=5$

gate2008 programming programming-in-c easy

© Copyright GATE Overflow. All rights reserved.

4.7.26 Programming In C: GATE2008-60

https://gateoverflow.in/483

What is printed by the following C program?

```
int f(int x, int *py, int **ppz)
{
    int y, z;
    **ppz += 1; z = **ppz;   // corrected z = *ppz; to z = **ppz;
    *py += 2; y = *py;
    x += 3;
    return x+y+z;
}

void main()
{
    int c, *b, **a;
    c = 4; b = &c; a = &b;
    printf("%d", f(c, b, a));

}
```

A. 18 B. 19 C. 21 D. 22

gate2008 programming programming-in-c normal

4.7.27 Programming In C: GATE2008-61

https://gateoverflow.in/484

Choose the correct option to fill ?1 and ?2 so that the program below prints an input string in reverse order. Assume that the input string is terminated by a new line character.

```
void reverse(void)
{
    int c;
    if(?1) reverse();
    ?2
}
main()
{
    printf("Enter text");
    printf("\n");
    reverse();
    printf("\n");
}
```

A. ?1 is $(getchar() != '\n')$
?2 is $getchar(c);$

B. ?1 is $((c = getchar()) != '\n')$
?2 is $getchar(c);$

C. ?1 is $(c != '\n')$
?2 is $putchar(c);$

D. ?1 is $((c = getchar()) != '\n')$
?2 is $putchar(c);$

gate2008 programming normal programming-in-c

4.7.28 Programming In C: GATE2008-IT-49

https://gateoverflow.in/3359

What is the output printed by the following C code?

```
# include <stdio.h>
int main ()
{
    char a [6] = "world";
    int i, j;
    for (i = 0, j = 5; i < j; a [i++] = a [j--]);
    printf ("%s\n", a);
}
```

A. dlrow B. Null string C. dlrld D. worow

gate2008-it programming programming-in-c normal

© Copyright GATE Overflow. All rights reserved.

4.7.29 Programming In C: GATE2008-IT-50

https://gateoverflow.in/3360

Consider the C program below. What does it print?

```
# include <stdio.h>
# define swapl (a, b) tmp = a; a = b; b = tmp
void swap2 ( int a, int b)
{
        int tmp;
        tmp = a; a = b; b = tmp;
 }
void swap3 (int*a, int*b)
{
        int tmp;
        tmp = *a; *a = *b; *b = tmp;
}
int main ()
{
        int num1 = 5, num2 = 4, tmp;
        if (num1 < num2) {swap1 (num1, num2);}
        if (num1 < num2) {swap2 (num1 + 1, num2);}
        if (num1 > = num2) {swap3 (&num1, &num2);}
        printf ("%d, %d", num1, num2);
}
```

A. 5,5 B. 5,4 C. 4,5 D. 4,4

gate2008-it programming programming-in-c normal

4.7.30 Programming In C: GATE2008-IT-51

https://gateoverflow.in/3361

Consider the C program given below. What does it print?

```
#include <stdio.h>
int main ()
{
        int i, j;
        int a [8] = {1, 2, 3, 4, 5, 6, 7, 8};
        for(i = 0; i < 3; i++) {
             a[i] = a[i] + 1;
             i++;
        }
        i--;
        for (j = 7; j > 4; j--) {
              int i = j/2;
              a[i] = a[i] - 1;
        }
        printf ("%d, %d", i, a[i]);
}
```

A. 2,3 B. 2,4 C. 3,2 D. 3,3

gate2008-it programming programming-in-c normal

4.7.31 Programming In C: GATE2008-IT-52

https://gateoverflow.in/3362

C program is given below:

```
# include <stdio.h>
int main ()
{
        int i, j;
        char a [2] [3] = {{'a', 'b', 'c'}, {'d', 'e', 'f'}};
        char b [3] [2];
        char *p = *b;
        for (i = 0; i < 2; i++) {
              for (j = 0; j < 3; j++) {
              *(p + 2*j + i) = a [i] [j];
              }
        }
}
```

What should be the contents of the array b at the end of the program?

A. a b
c d
e f

© Copyright GATE Overflow. All rights reserved.

B. a d
b e
c f

C. a c
e b
d f

D. a e
d c
b f

gate2008-it programming programming-in-c normal

4.7.32 Programming In C: GATE2010-11

https://gateoverflow.in/2184

What does the following program print?

```
#include<stdio.h>

void f(int *p, int *q) {
    p=q;
    *p=2;
}

int i=0, j=1;

int main() {
    f(&i, &j);
    printf("%d %d\n", i,j);
    return 0;
}
```

A. 2 2 B. 2 1 C. 0 1 D. 0 2

gate2010 programming programming-in-c easy

4.7.33 Programming In C: GATE2011-22

https://gateoverflow.in/2124

What does the following fragment of C program print?

```
    char c[] = "GATE2011";
    char *p = c;
    printf("%s", p + p[3] - p[1]);
```

A. *GATE*2011 B. *E*2011 C. 2011 D. 011

gate2011 programming programming-in-c normal

4.7.34 Programming In C: GATE2012-3

https://gateoverflow.in/35

What will be the output of the following C program segment?

```
char inChar = 'A';
   switch ( inChar ) {
      case 'A' : printf ("Choice A \n");
      case 'B' :
      case 'C' : printf ("Choice B");
      case 'D' :
      case 'E' :
      default : printf ("No Choice");
   }
```

A. No Choice
C. Choice A
Choice B No Choice

B. Choice A
D. Program gives no output as it is erroneous

gate2012 programming easy programming-in-c

4.7.35 Programming In C: GATE2012-48

https://gateoverflow.in/2176

Consider the following C code segment.

```
int a, b, c = 0;
```

© Copyright GATE Overflow. All rights reserved.

```
void prtFun(void);
main()
{
    static int a = 1;       /* Line 1 */
    prtFun();
    a += 1;
    prtFun();
    printf(" \n %d %d ", a, b);
}

void prtFun(void)
{
    static int a = 2;       /* Line 2 */
    int b = 1;
    a += ++b;
    printf(" \n %d %d ", a, b);
}
```

What output will be generated by the given code segment?

A. 3 1
4 1
4 2

B. 4 2
6 1
6 1

C. 4 2
6 2
2 0

D. 3 1
5 2
5 2

gate2012 programming programming-in-c normal

4.7.36 Programming In C: GATE2012-49

https://gateoverflow.in/43314

Consider the following C code segment.

```
int a, b, c = 0;
void prtFun(void);
main()
{
    static int a = 1;       /* Line 1 */
    prtFun();
    a += 1;
    prtFun();
    printf(" \n %d %d ", a, b);
}

void prtFun(void)
{
    static int a = 2;       /* Line 2 */
    int b = 1;
    a += ++b;
    printf(" \n %d %d ", a, b);
}
```

What output will be generated by the given code segment if:

Line 1 is replaced by **auto int** $a = 1;$

Line 2 is replaced by **register int** $a = 2;$

A. 3 1
4 1
4 2

B. 4 2
6 1
6 1

C. 4 2
6 2
2 0

D. 4 2
4 2
2 0

normal gate2012 programming-in-c programming

4.7.37 Programming In C: GATE2014-1-10

https://gateoverflow.in/1770

Consider the following program in C language:

```
#include <stdio.h>

main()
{
   int i;
   int*pi = &i;

   scanf("%d",pi);
   printf("%d\n", i+5);
}
```

Which one of the following statements is **TRUE?**

A. Compilation fails.

© Copyright GATE Overflow. All rights reserved.

B. Execution results in a run-time error.
C. On execution, the value printed is 5 more than the address of variable i.
D. On execution, the value printed is 5 more than the integer value entered.

gate2014-1 programming programming-in-c easy

4.7.38 Programming In C: GATE2014-2-11

https://gateoverflow.in/1965

Suppose n and p are unsigned int variables in a C program. We wish to set p to ${}^nC_3$. If n is large, which one of the following statements is most likely to set p correctly?

A. $p = n * (n-1) * (n-2)/6;$
B. $p = n * (n-1)/2 * (n-2)/3;$
C. $p = n * (n-1)/3 * (n-2)/2;$
D. $p = n * (n-1) * (n-2)/6.0;$

gate2014-2 programming programming-in-c normal

4.7.39 Programming In C: GATE2014-2-42

https://gateoverflow.in/2008

Consider the C function given below.

```
int f(int j)
{
    static int i = 50;
    int k;
    if (i == j)
    {
        printf("something");
        k = f(i);
        return 0;
    }
    else return 0;
}
```

Which one of the following is **TRUE**?

A. The function returns 0 for all values of j.
B. The function prints the string **something** for all values of j.
C. The function returns 0 when $j = 50$.
D. The function will exhaust the runtime stack or run into an infinite loop when $j = 50$.

gate2014-2 programming programming-in-c

4.7.40 Programming In C: GATE2015-1-11

https://gateoverflow.in/8185

The output of the following C program is_____.

```
void f1 ( int a, int b)  {
                int c;
                c = a; a = b;
                 b = c;
}
void f2 ( int * a, int * b) {
               int c;
               c = * a; *a = *b; *b = c;
}
int main () {
        int a = 4, b = 5, c = 6;
        f1 ( a, b);
        f2 (&b, &c);
        printf ("%d", c - a - b);
 }
```

gate2015-1 programming programming-in-c easy numerical-answers

4.7.41 Programming In C: GATE2015-1-35

https://gateoverflow.in/8283

What is the output of the following C code? Assume that the address of x is 2000 (in decimal) and an integer requires four bytes of memory.

```
int main () {
         unsigned int x [4] [3] =
         {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12}};
         printf ("%u, %u, %u", x + 3, *(x + 3), *(x + 2) + 3);
```

© Copyright GATE Overflow. All rights reserved.

```
}
```

A. $2036, 2036, 2036$
B. $2012, 4, 2204$
C. $2036, 10, 10$
D. $2012, 4, 6$

gate2015-1 programming programming-in-c normal

4.7.42 Programming In C: GATE2015-2-15

https://gateoverflow.in/8086

Consider the following function written in the C programming langauge :

```
void foo(char *a)
 {
    if (*a && *a != ' ')
      {
       foo(a+1);
       putchar(*a);
      }
}
```

The output of the above function on input "$ABCD\ EFGH$" is

A. $ABCD\ EFGH$ B. $ABCD$ C. $HGFE\ DCBA$ D. $DCBA$

gate2015-2 programming programming-in-c normal

4.7.43 Programming In C: GATE2015-3-26

https://gateoverflow.in/8478

Consider the following C program

```
#include<stdio.h>
int main()  {
    static int a[] = {10, 20, 30, 40, 50};
    static int *p[] = {a, a+3, a+4, a+1, a+2};
    int **ptr = p;
    ptr++;
    printf("%d%d", ptr-p, **ptr);

}
```

The output of the program is _____.

gate2015-3 programming programming-in-c normal numerical-answers

4.7.44 Programming In C: GATE2015-3-30

https://gateoverflow.in/8486

Consider the following two C code segments. Y and X are one and two dimensional arrays of size n and $n \times n$ respectively, where $2 \leq n \leq 10$. Assume that in both code segments, elements of Y are initialized to 0 and each element $X[i][j]$ of array X is initialized to $i + j$. Further assume that when stored in main memory all elements of X are in same main memory page frame.

Code segment 1:

```
// initialize elements of Y to 0
// initialize elements of X[i][j] of X to i+j
for (i=0; i<n; i++)
    Y[i] += X[0][i];
```

Code segment 2:

```
// initialize elements of Y to 0
// initialize elements of X[i][j] of X to i+j
for (i=0; i<n; i++)
    Y[i] += X[i][0];
```

Which of the following statements is/are correct?

S1: Final contents of array Y will be same in both code segments

S2: Elements of array X accessed inside the for loop shown in code segment 1 are contiguous in main memory

S3: Elements of array X accessed inside the for loop shown in code segment 2 are contiguous in main memory

A. Only S2 is correct
B. Only S3 is correct
C. Only S1 and S2 are correct
D. Only S1 and S3 are correct

gate2015-3 programming-in-c normal

© Copyright GATE Overflow. All rights reserved.

4.7.45 Programming In C: GATE2015-3-48

https://gateoverflow.in/8557

Consider the following C program:

```
#include<stdio.h>
int main()
{
    int i, j, k = 0;
    j=2 * 3 / 4 + 2.0 / 5 + 8 / 5;
    k-=--j;
    for (i=0; i<5; i++)
    {
        switch(i+k)
        {
            case 1:
            case 2: printf("\n%d", i+k);
            case 3: printf("\n%d", i+k);
            default: printf("\n%d", i+k);
        }
    }
    return 0;
}
```

The number of times printf statement is executed is _____.

gate2015-3 programming programming-in-c normal numerical-answers

4.7.46 Programming In C: GATE2015-3-54

https://gateoverflow.in/8563

Consider the following C program:

```
#include<stdio.h>
int f1(void);
int f2(void);
int f3(void);
int x=10;
int main()
{
    int x=1;
    x += f1() + f2 () + f3() + f2();
    printf("%d", x);
    return 0;
}
int f1() { int x = 25; x++; return x;}
int f2() { static int x = 50; x++; return x;}
int f3() { x *= 10; return x;}
```

The output of the program is _____.

gate2015-3 programming programming-in-c normal numerical-answers

4.7.47 Programming In C: GATE2015-3-7

https://gateoverflow.in/8401

Consider the following C program segment.

```
# include <stdio.h>
int main()
{
    char s1[7] = "1234", *p;
    p = s1 + 2;
    *p = '0';
    printf("%s", s1);
}
```

What will be printed by the program?

A. 12 B. 120400 C. 1204 D. 1034

gate2015-3 programming programming-in-c normal

4.7.48 Programming In C: GATE2016-1-12

https://gateoverflow.in/39638

Consider the following "C" program.

```
void f(int, short);
void main()
{
    int i = 100;
```

© Copyright GATE Overflow. All rights reserved.

```
    short s = 12;
    short *p = &s;
    _____________;  // call to f()
}
```

Which one of the following expressions , when placed in the blank above, will NOT result in a type checking error?

A. $f(s, *s)$ B. $i = f(i, s)$ C. $f(i, *s)$ D. $f(i, *p)$

gate2016-1 programming-in-c easy

4.7.49 Programming In C: GATE2016-1-15 https://gateoverflow.in/39642

Consider the following C program.

```
# include <stdio.h>
void mystery (int *ptra, int *ptrb) {
     int *temp;
     temp = ptrb;
     ptrb =ptra;
     ptra = temp;
}
int main () {
    int a = 2016, b=0, c= 4, d = 42;
    mystery (&a, &b);
    if (a < c)
          mystery (&c, &a);
    mystery (&a, &d);
    print f("%d\n", a);
}
```

The output of the program is _____.

gate2016-1 programming-in-c easy numerical-answers

4.7.50 Programming In C: GATE2016-1-34 https://gateoverflow.in/39704

The following function computes the maximum value contained in an integer array $P[\,]$ of size n $(n >= 1)$.

```
int max (int *p,int n) {
    int a = 0, b=n-1;

    while (__________) {
        if (p[a]<= p[b]) {a = a+1;}
        else             {b = b-1;}
    }
    return p[a];
}
```

The missing loop condition is:

A. $a! = n$ B. $b! = 0$
C. $b > (a+1)$ D. $b! = a$

gate2016-1 programming-in-c normal

4.7.51 Programming In C: GATE2016-2-12 https://gateoverflow.in/39565

The value printed by the following program is _____.

```
      void f (int * p, int m) {
           m = m + 5;
          *p = *p + m;
           return;
      }
      void main () {
       int i=5, j=10;

       f (&i, j);
       print f ("%d", i+j);
     }
```

gate2016-2 programming-in-c normal numerical-answers

© Copyright GATE Overflow. All rights reserved.

4.7.52 Programming In C: GATE2016-2-37

https://gateoverflow.in/39602

Consider the following program:

```
int f (int * p, int n)
{
    if (n <= 1) return 0;
    else return max (f (p+1, n-1), p[0] - p[1]);
}
 int main ()
{
    int a[] = {3, 5, 2, 6, 4};
    print f(" %d", f(a, 5));
}
```

Note: $max(x,y)$ returns the maximum of x and y.

The value printed by this program is _____.

gate2016-2 programming-in-c normal numerical-answers

4.7.53 Programming In C: GATE2017-1-13

https://gateoverflow.in/118293

Consider the following C code:

```
#include<stdio.h>
int *assignval (int *x, int val) {
    *x = val;
    return x;
}

void main () {
    int *x = malloc(sizeof(int));
    if (NULL == x) return;
    x = assignval (x,0);
    if (x) {
        x = (int *)malloc(sizeof(int));
        if (NULL == x) return;
        x = assignval (x,10);
    }
    printf("%d\n", *x);
    free(x);
}
```

The code suffers from which one of the following problems:

A. compiler error as the return of $malloc$ is not typecast appropriately.
B. compiler error because the comparison should be made as $x == NULL$ and not as shown.
C. compiles successfully but execution may result in dangling pointer.
D. compiles successfully but execution may result in memory leak.

gate2017-1 programming-in-c programming

4.7.54 Programming In C: GATE2017-1-36

https://gateoverflow.in/118319

Consider the C functions foo and bar given below:

```
int foo(int val) {
    int x=0;
    while(val > 0) {
        x = x + foo(val--);
    }
    return val;
}
```

```
int bar(int val) {
    int x = 0;
    while(val > 0) {
        x= x + bar(val-1);
    }
    return val;
}
```

Invocations of $foo(3)$ and $bar(3)$ will result in:

A. Return of 6 and 6 respectively.
B. Infinite loop and abnormal termination respectively.

© Copyright GATE Overflow. All rights reserved.

C. Abnormal termination and infinite loop respectively.

D. Both terminating abnormally.

gate2017-1 programming-in-c programming normal

4.7.55 Programming In C: GATE2017-1-53

https://gateoverflow.in/118473

Consider the following C program.

```
#include<stdio.h>
#include<string.h>

void printlength(char *s, char *t) {
    unsigned int c=0;
    int len = ((strlen(s) - strlen(t)) > c) ? strlen(s) : strlen(t);
    printf("%d\n", len);
}

void main() {
    char *x = "abc";
    char *y = "defgh";
    printlength(x,y);
}
```

Recall that *strlen* is defined in *string.h* as returning a value of type *size_t*, which is an unsigned int. The output of the program is _____ .

gate2017-1 programming programming-in-c normal numerical-answers

4.7.56 Programming In C: GATE2017-1-55

https://gateoverflow.in/118442

The output of executing the following C program is _____ .

```
#include<stdio.h>

int total(int v) {
    static int count = 0;
    while(v) {
        count += v&1;
        v >>= 1;
    }
    return count;
}

void main() {
    static int x=0;
    int i=5;
    for(; i>0; i--) {
        x = x + total(i);
    }
    printf("%d\n", x);
}
```

gate2017-1 programming programming-in-c normal numerical-answers

4.7.57 Programming In C: GATE2017-2-14

https://gateoverflow.in/118245

Consider the following function implemented in C:

```
void printxy(int x, int y) {
    int *ptr;
    x=0;
    ptr=&x;
    y=*ptr;
    *ptr=1;
    printf("%d, %d", x, y);
}
```

The output of invoking *printxy*$(1,1)$ is:

A. $0,0$ B. $0,1$ C. $1,0$ D. $1,1$

gate2017-2 programming-in-c programming

© Copyright GATE Overflow. All rights reserved.

4.7.58 Programming In C: GATE2017-2-2

https://gateoverflow.in/118171

Match the following:

P.	static char var ;	i.	Sequence of memory locations to store addresses
Q.	m = malloc(10); m =NULL ;	ii.	A variable located in data section of memory
R.	char *ptr[10] ;	iii.	Request to allocate a CPU register to store data
S.	register int varl;	iv.	A lost memory which cannot be freed

A. P-ii; Q-iv; R-i; S-iii
B. P-ii; Q-i; R-iv; S-iii
C. P-ii; Q-iv; R-iii; S-i
D. P-iii; Q-iv; R-i; S-ii

gate2017-2 programming programming-in-c

4.7.59 Programming In C: GATE2017-2-54

https://gateoverflow.in/118272

Consider the following C program.

```
#include<stdio.h>
int main () {
    int m=10;
    int n, n1;
    n=++m;
    n1=m++;
    n--;
    --n1;
    n-=n1;
    printf("%d", n);
    return 0;
}
```

The output of the program is _____

gate2017-2 programming-in-c numerical-answers

4.7.60 Programming In C: GATE2017-2-55

https://gateoverflow.in/118335

Consider the following C program.

```
#include<stdio.h>
#include<string.h>
int main() {
    char* c="GATECSIT2017";
    char* p=c;
    printf("%d", (int)strlen(c+2[p]-6[p]-1));
    return 0;
}
```

The output of the program is _____

gate2017-2 programming-in-c numerical-answers

4.7.61 Programming In C: GATE2018-32

https://gateoverflow.in/204106

Consider the following C code. Assume that unsigned long int type length is 64 bits.

```
unsigned long int fun(unsigned long int n) {
        unsigned long int i, j=0, sum = 0;
        for( i=n; i>1; i=i/2) j++;
        for( ; j>1; j=j/2) sum++;
        return sum;
}
```

The value returned when we call fun with the input 2^{40} is:

A. 4 B. 5 C. 6 D. 40

gate2018 programming-in-c normal programming

4.7.62 Programming In C: GATE2018-45

https://gateoverflow.in/204120

Consider the following program written in pseudo-code. Assume that x and y are integers.

```
Count (x, y) {
```

© Copyright GATE Overflow. All rights reserved.

```
    if (y !=1 ) {
        if (x !=1) {
            print("*");
            Count (x/2, y);
        }
        else {
            y=y-1;
            Count (1024, y);
        }
    }
}
```

The number of times that the $print$ statement is executed by the call $Count(1024,1024)$ is _____

gate2018 programming-in-c numerical-answers

4.7.63 Programming In C: GATE2019-18

https://gateoverflow.in/302830

Consider the following C program :

```
#include<stdio.h>
int jumble(int x, int y){
    x = 2*x+y;
    return x;
}
int main(){
    int x=2, y=5;
    y=jumble(y,x);
    x=jumble(y,x);
    printf("%d \n",x);
    return 0;
}
```

The value printed by the program is _____.

gate2019 numerical-answers programming-in-c programming

4.7.64 Programming In C: GATE2019-24

https://gateoverflow.in/302824

Consider the following C program:

```
#include <stdio.h>
int main() {
        int arr[]={1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 5}, *ip=arr+4;
        printf("%d\n", ip[1]);
        return 0;
}
```

The number that will be displayed on execution of the program is _____

gate2019 numerical-answers programming-in-c programming

4.7.65 Programming In C: GATE2019-27

https://gateoverflow.in/302821

Consider the following C program:

```
#include <stdio.h>
int r() {
       static int num=7;
       return num--;
}
int main() {
       for (r();r();r())
              printf("%d",r());
       return 0;
}
```

Which one of the following values will be displayed on execution of the programs?

A. 41 B. 52 C. 63 D. 630

gate2019 programming-in-c programming

4.7.66 Programming In C: GATE2019-52

https://gateoverflow.in/302796

Consider the following C program:

© Copyright GATE Overflow. All rights reserved.

```c
#include <stdio.h>
int main() {
    float sum = 0.0, j=1.0, i=2.0;
    while (i/j > 0.0625) {
        j=j+j;
        sum=sum+i/j;
        printf("%f\n", sum);
    }
    return 0;
}
```

The number of times the variable sum will be printed, when the above program is executed, is _____

gate2019 numerical-answers programming-in-c programming

4.7.67 Programming In C: GATE2019-53

https://gateoverflow.in/302795

Consider the following C program:

```c
#include <stdio.h>
int main()
{
    int a[] = {2, 4, 6, 8, 10};
    int i, sum=0, *b=a+4;
    for (i=0; i<5; i++)
        sum=sum+(*b-i)-*(b-i);
    printf("%d\n", sum);
    return 0;
}
```

The output of the above C program is _____

gate2019 numerical-answers programming-in-c programming

4.7.68 Programming In C: TIFR2018-A-7

https://gateoverflow.in/179276

```c
Consider the following function definition.
void greet(int n)
{
    if(n>0)
    {
        printf("hello");
        greet(n-1);
    }
    printf("world");
}
```

If you run greet(n) for some non-negative integer n, what would it print?

A. n times "hello", followed by n+1 times "world"
B. n times "hello", followed by n times "world"
C. n times "helloworld"
D. n+1 times "helloworld"
E. n times "helloworld", followed by "world"

tifr2018 programming-in-c

4.7.69 Programming In C: TIFR2019-B-6

https://gateoverflow.in/280489

Given the following pseudocode for function printx() below, how many times is x printed if we execute printx(5)?

```c
void printx(int n)  {
    if(n==0){
        printf("x");
}
for(int i=0;i<=n-1;++i){
    printx(n-1);
}
}
```

A. 625 B. 256 C. 120 D. 24 E. 5

tifr2019 programming programming-in-c

4.8 Programming Paradigms (2)

© Copyright GATE Overflow. All rights reserved.

4.8.1 Programming Paradigms: GATE2004-1

https://gateoverflow.in/998

The goal of structured programming is to:

A. have well indented programs
B. be able to infer the flow of control from the compiled code
C. be able to infer the flow of control from the program text
D. avoid the use of GOTO statements

gate2004 programming easy programming-paradigms

4.8.2 Programming Paradigms: GATE2004-90

https://gateoverflow.in/1084

Choose the best matching between the programming styles in Group 1 and their characteristics in Group 2.

Group 1		Group 2	
$P.$	Functional	1.	Common-based, procedural
$Q.$	Logic	2.	Imperative, abstract data types
$R.$	Object-oriented	3.	Side-effect free, declarative, expression evaluations
$S.$	Imperative	4.	Declarative, clausal representation, theorem proving

A. $P-2 \quad Q-3 \quad R-4 \quad S-1$
B. $P-4 \quad Q-3 \quad R-2 \quad S-1$
C. $P-3 \quad Q-4 \quad R-1 \quad S-2$
D. $P-3 \quad Q-4 \quad R-2 \quad S-1$

gate2004 programming normal programming-paradigms

4.9 Recursion (17)

4.9.1 Recursion: GATE1991-01,x

https://gateoverflow.in/507

Consider the following recursive definition of fib:

```
fib(n) :=  if n = 0 then 1
           else if n = 1 then 1
           else fib(n-1) + fib(n-2)
```

The number of times fib is called (including the first call) for evaluation of $fib(7)$ is_____.

gate1991 programming recursion normal

4.9.2 Recursion: GATE1994-21

https://gateoverflow.in/2517

Consider the following recursive function:

```
function fib (n:integer);integer;
begin
if (n=0) or (n=1) then fib := 1
else fib := fib(n-1) + fib(n-2)
end;
```

The above function is run on a computer with a stack of 64 bytes. Assuming that only return address and parameter are passed on the stack, and that an integer value and an address takes 2 bytes each, estimate the maximum value of n for which the stack will not overflow. Give reasons for your answer.

gate1994 programming recursion normal

4.9.3 Recursion: GATE1995-2.9

https://gateoverflow.in/2621

A language with string manipulation facilities uses the following operations

```
head(s): first character of a string
tail(s): all but exclude the first character of a string
```

```
concat(s1, s2): s1s2
```

For the string "$acbc$" what will be the output of

```
concat(head(s), head(tail(tail(s))))
```

© Copyright GATE Overflow. All rights reserved.

A. ac B. bc C. ab D. cc

gate1995 algorithms normal recursion

4.9.4 Recursion: GATE2000-16

https://gateoverflow.in/687

A recursive program to compute Fibonacci numbers is shown below. Assume you are also given an array $f[0 \ldots m]$ with all elements initialized to 0.

```
fib(n) {
    if (n > M) error ();
    if (n == 0) return 1;
    if (n == 1)return 1;
    if (▭)_____________________(1)
        return ▭_____________(2)
    t = fib(n - 1) + fib(n - 2);
    ▭_____________________(3)
    return t;
}
```

A. Fill in the boxes with expressions/statement to make $fib()$ store and reuse computed Fibonacci values. Write the box number and the corresponding contents in your answer book.
B. What is the time complexity of the resulting program when computing $fib(n)$?

gate2000 algorithms normal descriptive recursion

4.9.5 Recursion: GATE2001-13

https://gateoverflow.in/754

Consider the following C program:

```
void abc(char*s)
{
    if(s[0]=='\0')return;
    abc(s+1);
    abc(s+1);
    printf("%c",s[0]);
}

main()
{
    abc("123");
}
```

A. What will be the output of the program?
B. If $abc(s)$ is called with a null-terminated string s of length n characters (not counting the null ('\0') character), how many characters will be printed by $abc(s)$?

gate2001 programming recursion normal descriptive

4.9.6 Recursion: GATE2002-11

https://gateoverflow.in/864

The following recursive function in C is a solution to the Towers of Hanoi problem.

```
void move(int n, char A, char B, char C)  {
    if (.....................) {
        move (..........................);
        printf("Move disk %d from pole %c to pole %c\n", n, A,C);
        move (.....................);
    }
}
```

Fill in the dotted parts of the solution.

gate2002 programming recursion normal descriptive

4.9.7 Recursion: GATE2004-31, ISRO2008-40

https://gateoverflow.in/1028

Consider the following C function:

```
int f(int n)
{
    static int i = 1;
    if(n >= 5) return n;
```

© Copyright GATE Overflow. All rights reserved.

```
    n = n+i;
    i++;
    return f(n);
}
```

The value returned by $f(1)$ is:

A. 5 B. 6 C. 7 D. 8

gate2004 programming programming-in-c recursion easy isro2008

4.9.8 Recursion: GATE2005-81a

https://gateoverflow.in/1403

```
double foo(int n)
{
    int i;
    double sum;
    if(n == 0)
    {
        return 1.0;
    }
    else
    {
        sum = 0.0;
        for(i = 0; i < n; i++)
        {
            sum += foo(i);
        }
        return sum;
    }

}
```

The space complexity of the above code is?

A. $O(1)$ B. $O(n)$ C. $O(n!)$ D. n^n

gate2005 programming recursion normal

4.9.9 Recursion: GATE2005-81b

https://gateoverflow.in/82146

```
double foo(int n)
{
    int i;
    double sum;
    if(n == 0)
    {
        return 1.0;
    }
    else
    {
        sum = 0.0;
        for(i = 0; i < n; i++)
        {
            sum += foo(i);
        }
        return sum;
    }

}
```

Suppose we modify the above function $foo()$ and stores the value of $foo(i)\ 0 \leq i < n$, as and when they are computed. With this modification the time complexity for function $foo()$ is significantly reduced. The space complexity of the modified function would be:

A. $O(1)$ B. $O(n)$ C. $O(n^2)$ D. $n!$

gate2005 programming recursion normal

4.9.10 Recursion: GATE2007-42

https://gateoverflow.in/1240

Consider the following C function:

```
int f(int n)
{
    static int r = 0;
```

© Copyright GATE Overflow. All rights reserved.

```
    if (n <= 0) return 1;
    if (n > 3)
    {    r = n;
         return f(n-2) + 2;
    }
    return f(n-1) + r;
}
```

What is the value of $f(5)$?

A. 5 B. 7 C. 9 D. 18

gate2007 programming recursion normal

4.9.11 Recursion: GATE2007-IT-27

https://gateoverflow.in/3460

The function f is defined as follows:

```
int f (int n) {
    if (n <= 1) return 1;
    else if (n % 2  ==  0) return f(n/2);
    else return f(3n - 1);
}
```

Assuming that arbitrarily large integers can be passed as a parameter to the function, consider the following statements.

i. The function f terminates for finitely many different values of $n \geq 1$.
ii. The function f terminates for infinitely many different values of $n \geq 1$.
iii. The function f does not terminate for finitely many different values of $n \geq 1$.
iv. The function f does not terminate for infinitely many different values of $n \geq 1$.

Which one of the following options is true of the above?

A. i and iii B. i and iv C. ii and iii D. ii and iv

gate2007-it programming recursion normal

4.9.12 Recursion: GATE2014-2-40

https://gateoverflow.in/2000

Consider the following function.

```
double f(double x){
    if( abs(x*x - 3) < 0.01)
        return x;
    else
        return f(x/2 + 1.5/x);
}
```

Give a value q (to 2 decimals) such that $f(q)$ will return q:_____.

gate2014-2 programming recursion numerical-answers normal

4.9.13 Recursion: GATE2016-1-35

https://gateoverflow.in/39730

What will be the output of the following C program?

```
void count (int n) {
    static int d=1;

    printf ("%d",n);
    printf ("%d",d);
    d++;
    if (n>1) count (n-1);
    printf ("%d",d);


}

void main(){
    count (3);
}
```

A. $3\,1\,2\,2\,1\,3\,4\,4\,4$
B. $3\,1\,2\,1\,1\,1\,2\,2\,2$
C. $3\,1\,2\,2\,1\,3\,4$

© Copyright GATE Overflow. All rights reserved.

D. 3121112

gate2016-1 programming-in-c recursion normal

4.9.14 Recursion: GATE2017-1-35

https://gateoverflow.in/118317

Consider the following two functions.

```
void fun1(int n) {
    if(n == 0) return;
    printf("%d", n);
    fun2(n - 2);
    printf("%d", n);
}
void fun2(int n) {
    if(n == 0) return;
    printf("%d", n);
    fun1(++n);
    printf("%d", n);
}
```

The output printed when fun1(5) is called is

A. 53423122233445
B. 53423120112233
C. 53423122132435
D. 53423120213243

gate2017-1 programming normal tricky recursion

4.9.15 Recursion: GATE2018-21

https://gateoverflow.in/204095

Consider the following C program:

```
#include<stdio.h>

int counter=0;

int calc (int a, int b) {
        int c;
        counter++;
        if(b==3) return (a*a*a);
        else {
                c = calc(a, b/3);
                return (c*c*c);
        }
}

int main() {
        calc(4, 81);
        printf("%d", counter);
}
```

The output of this program is _____.

gate2018 programming-in-c numerical-answers recursion programming

4.9.16 Recursion: TIFR2010-B-31

https://gateoverflow.in/26484

Consider the following computation rules. **Parallel-outermost rule**: Replace all the outermost occurrences of F (i.e., all occurrences of F which do not occur as arguments of other F's) simultaneously. **Parallel - innermost rule**: Replace all the innermost occurrences of F (i.e.,all occurrences of F with all arguments free of F's) simultaneously. Now consider the evaluations of the recursive program over the integers.

```
F(x, y) <== if x = 0 then 0 else
            [ F(x + 1, F(x, y)) * F(x - 1, F(x, y))]
```

where the multiplication functions * is extended as follows:

```
0 * w & w * 0 are 0
a * w & w * a are w (for any non-zero integer a)
w * w is w
```

We say that $F(x,y) =$ w when the evaluation of $F(x,y)$ does not terminate. Computing $F(1,0)$ using the parallel -

© Copyright GATE Overflow. All rights reserved.

innermost and parallel - outermost rule yields

A. w and 0 respectively
B. 0 and 0 respectively
C. w and w respectively
D. w and 1 respectively
E. none of the above

tifr2010 programming recursion

4.9.17 Recursion: TIFR2011-B-38

https://gateoverflow.in/20923

Consider the class of recursive and iterative programs. Which of the following is false?

A. Recursive programs are more powerful than iterative programs.
B. For every iterative program there is an equivalent recursive program.
C. Recursive programs require dynamic memory management.
D. Recursive programs do not terminate sometimes.
E. Iterative programs and recursive programs are equally expressive.

tifr2011 recursion programming

4.10 Structures (1)

4.10.1 Structures: GATE2018-2

https://gateoverflow.in/204076

Consider the following C program:

```
#include<stdio.h>
struct Ournode{
    char x, y, z;
};
int main() {
    struct Ournode p={'1', '0', 'a'+2};
    struct Ournode *q=&p;
    printf("%c, %c", *((char*)q+1), *((char*)q+2));
    return 0;
}
```

The output of this program is:

A. 0, c B. 0, a+2 C. '0', 'a+2' D. '0', 'c'

gate2018 programming-in-c programming structures pointers normal

4.11 Type Checking (1)

4.11.1 Type Checking: GATE2003-24

https://gateoverflow.in/914

Which of the following statements is FALSE?

A. In statically typed languages, each variable in a program has a fixed type
B. In un-typed languages, values do not have any types
C. In dynamically typed languages, variables have no types
D. In all statically typed languages, each variable in a program is associated with values of only a single type during the execution of the program

gate2003 programming normal type-checking

4.12 Variable Binding (1)

4.12.1 Variable Binding: GATE2007-IT-34, UGCNET-Dec2012-III-52

https://gateoverflow.in/3467

Consider the program below in a hypothetical programming language which allows global variables and a choice of static or dynamic scoping.

```
int i ;
program main ()
{
    i = 10;
    call f();
}
```

© Copyright GATE Overflow. All rights reserved.

```
procedure f()
{
    int i = 20;
    call g ();
}
procedure g ()
{
    print i;
}
```

Let x be the value printed under static scoping and y be the value printed under dynamic scoping. Then, x and y are:

A. $x = 10, y = 20$ B. $x = 20, y = 10$ C. $x = 10, y = 10$ D. $x = 20, y = 20$

gate2007-it programming variable-binding normal ugcnetdec2012iii

© Copyright GATE Overflow. All rights reserved.