

GATE Overflow

2017 Vol. 2



www.gateoverflow.in

www.facebook.com/groups/gateoverflow

www.gatecse.in

This book was created programmatically by GATE Overflow on Jul 5, 2017. If you feel any doubt regarding the answer, click on the question link and give a comment on the site. Studying all these questions might get you 60 marks in GATE but that might not be enough for an IIT. So, read standard books, solve exercise questions and use these questions for cementing the concepts and aim 85+. At least if you are not getting the solution to a given problem first refer standard book. If any error is found on any question it shall be updated at <http://gateoverflow.in/corrections>.

PDFs for the remaining subjects will be made available at <http://classroom.gateoverflow.in> and you can enroll in [this course](#) to get notification for the same. Enrollment is free and account details are of GATE Overflow with a new password which have been sent to all registered emails on GATE Overflow. New users will receive this email within a few minutes of confirming their email address.

You can now join our [Facebook group](#) for GATE CSE discussions.

You can visit <http://playlists.gatecse.in> for high quality videos for GATE CSE and how to use GO site/ebook.

This book consists of only previous year GATE, TIFR, ISI and CMI questions (CS from 1987 and all 5 years of IT) all of which are relevant for GATE. Out of syllabus subjects as of GATE 2017 are removed from this book except in rare cases.

Since **GATE Overflow** started in August 2014, a lot of people have dedicated their time and effort in bringing this book now. Initiated by **Omesh Pandita** and **Arjun Suresh** as a Q/A platform for CSE students, **Kathleen Bankson** was instrumental in getting all previous year GATE questions here. Then experts like **Pravne Saini**, **Happy Mittal**, **Sankaranarayanan P.N.**, **Suraj Kumar** etc. have contributed a lot to the answers here. **Pragy Agarwal** even after topping GATE has continuously contributed here with his knowledge as well as in making the contents beautiful with fine latex skills. We also have to thank the work by **Jothee, Misbah, Ishrat** and **Nataliyah** who are continuously adding and keeping the contents here neat and clean. There are also many toppers of GATE 2015, 2016, 2017 and probably 2018 who are contributing a lot here. The list of all the contributors can be found [here](#) but even that does not include the contributions of some like Arif Ali Anapparakkal in helping design this book, **Arvind Devaraj** and others who have provided guidance and help etc. Last but not the least, we thank all the users of GATE Overflow.

We thank the contributions of **Silpa V.S.**, **Rahul Kumar Yadav** and others for getting the **GATECSE Lastrank** page maintained. **Bikram Ballav** is behind most of the exams on GO (<http://mockgate.com>) and **Arindam Sarkar** made the interface for it. **Pragy Agarwal** is also behind the rank and score predictor tool, (<http://mymarks.gatecse.in>) used by GO which has 99-100% accuracy over the last 2 years.

Special thanks to **Sachin Mittal** for making the **How to Use GO vidoes**, **Silpa V.S.** for classifying the questions topicwise for the book, **Pooja Palod** for making the **GATE 2018 schedule** and **Debashish Deka** for GO classroom contributions.

Also thanks to all toppers who took time to write a review for GO.

Table of Contents

1. Algorithms	(304)
1. Algorithm Design	(5)
2. Algorithm Design Techniques	(4)
3. Asymptotic Notations	(21)
4. Dynamic Programming	(11)
5. Graph Algorithms	(48)
6. Greedy Algorithm	(8)
7. Identify Function	(35)
8. Minimum Maximum	(4)
9. Np Completeness	(15)
10. Numerical Computation	(2)
11. Recurrence	(29)
12. Searching	(7)
13. Sorting	(52)
14. Spanning Tree	(31)
15. Time Complexity	(32)
2. Compiler Design	(157)
1. Abstract Syntax Tree	(1)
2. Assembler	(5)
3. Code Optimization	(4)
4. Compilation Phases	(7)
5. Expression Evaluation	(1)
6. Expression Tree	(1)
7. Grammar	(39)
8. Infix Postfix	(2)
9. Intermediate Code	(7)
10. Left Recursion	(1)
11. Lexical Analysis	(4)
12. Linking	(3)
13. Live Variable	(1)
14. Macros	(3)
15. Parameter Passing	(8)
16. Parsing	(42)
17. Programming In C	(1)
18. Recursion	(1)
19. Register Allocation	(4)
20. Runtime Environments	(9)
21. Static Single Assignment	(2)
22. Syntax Directed Translation	(6)
23. Target Code Generation	(2)
24. Variable Scope	(2)
25. Viable Prefix	(1)
3. Programming & DS: DS	(198)
1. Abstract Data Type	(1)
2. Arrays	(14)
3. Binary Search Tree	(29)
4. Binary Tree	(46)
5. Graphs	(6)
6. Hashing	(17)
7. Heap	(24)
8. Linked Lists	(19)
9. Queues	(11)
10. Stack	(17)
11. Trees	(14)
4. Programming & DS: Programming	(107)
1. Aliasing	(1)
2. Goto	(1)
3. Identify Function	(4)
4. Loop Invariants	(8)
5. Programming Constructs	(1)
6. Programming In C	(60)
7. Programming Paradigms	(3)
8. Pseudo Code	(1)
9. Recursion	(18)
10. Runtime Environments	(9)
11. Type Checking	(1)
5. Theory of Computation	(268)
1. Closure Property	(9)

2. Context Free Language	(32)
3. Countable Set	(1)
4. Decidability	(22)
5. Finite Automata	(63)
6. Grammar	(1)
7. Identify Class Language	(30)
8. Non Determinism	(7)
9. Pumping Lemma	(1)
10. Pushdown Automata	(13)
11. Recursive And Recursively Enumerable Languages	(13)
12. Regular Expressions	(28)
13. Regular Grammar	(3)
14. Regular Languages	(36)
15. Turing Machine	(9)

1

Algorithms (304) [top](#)

1.1

Algorithm Design(5) [top](#)

1.1.1 Algorithm Design: CMI2012-B-03b [top](#)

<http://gateoverflow.in/47088>

Let A be array of n integers, sorted so that $A[1] \leq A[2] \leq \dots \leq A[n]$. Suppose you are given a number x and you wish to find out if there exist indices k, l such that $A[k] + A[l] = x$.

- a. Design an $O(n)$ algorithm for this problem.

[descriptive](#) [cmi2012](#) [algorithms](#) [algorithm-design](#)

Answer

1.1.2 Algorithm Design: GATE1992-8 [top](#)

<http://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](#)

Answer

1.1.3 Algorithm Design: GATE2006-17 [top](#)

<http://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](#)

Answer

1.1.4 Algorithm Design: GATE2006-54 [top](#)

<http://gateoverflow.in/1832>

Given two arrays of numbers a_1, \dots, a_n and b_1, \dots, b_n where each number is 0 or 1, the fastest algorithm to find the largest span (i, j) such that $a_i + a_{i+1} + \dots + a_j = b_i + b_{i+1} + \dots + b_j$ 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](#)

Answer

1.1.5 Algorithm Design: TIFR2011-B-29 [top](#)

<http://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](#)

[Answer](#)

Answers: Algorithm Design

1.1.1 Algorithm Design: CMI2012-B-03b [top](#)

<http://gateoverflow.in/47088>



Selected Answer

Here is the algorithm, Which return **True** if there is a number x present such that $(A[k] + A[l] == x)$ else return **false**.

```
// Consider this algorithm is called as
// AlgorithmCheck(A, l, size, x);
// Where A is the sorted array

Bool AlgorithmCheck(A, L, K, x) {
    while(L < K) {
        if(A[L] + A[K] == x)
            return true;
        else if(A[L] + A[K] < x)
            L++;
        else
            K--;
    }
    return false;
}
```

This will take only **O(n)** time to do its work.

4 votes

-- Muktinath Vishwakarma (33.9k points)

1.1.2 Algorithm Design: GATE1992-8 [top](#)

<http://gateoverflow.in/58>

Union-Find Algorithm can be used to find the cycle.

Ref: <http://www.geeksforgeeks.org/union-find/>

3 votes

-- Rajarshi Sarkar (34.4k points)

1.1.3 Algorithm Design: GATE2006-17 [top](#)

<http://gateoverflow.in/978>



Selected Answer

Ans B should be correct.

We can move from right keeping a note of the maximum element(suppose current_max). At the start the right most element will always be a leader. If an element is greater than our current_max, it will a leader. Add this element to leaders. Set current_max to this element and carry on leftward. Time Complexity would be O(n)

12 votes

-- Madhur Rawat (2.6k points)

1.1.4 Algorithm Design: GATE2006-54 [top](#)

<http://gateoverflow.in/1832>



Selected Answer

Answer is (C). Following algorithm would do.

Since array is binary, the max sum will go until n and so the sum difference of the two arrays can vary between $-n$ and n . We use array start to keep the starting index of each possible sum (hence of size $2n + 1$) and array end to keep the ending index (these two arrays work like hash tables and since we have only $2n + 1$ possible keys, we can do a perfect hashing). So, our required solution will be $\max(\text{end}[i] - \text{start}[i])$ provided both are assigned values.

The algorithm works as follows:

1. Initialize diff array to contain the difference of sum of elements of array a and b. i.e., $\text{diff}[i] = \sum_{i=0}^n a[i] - b[i]$.
2. Now $\text{diff}[i]$ can have values from $-n$ to n which gives $2n + 1$ possible values and the first occurrence of a diff value marks the beginning of a span and the last occurrence marks the end. We use start and end array for storing these two positions for the $2n + 1$ possible values.
3. Now, the largest value of $\text{end}[i] - \text{start}[i]$ for any i , will be the largest span and the start of it will be $\text{start}[i] + 1$, and end will be $\text{end}[i]$. If the span is starting from first position itself (arrays a and b have same first elements), then it will start from $\text{start}[i]$ itself.

```
#include <stdio.h>

#define size 100 //assume n is less than 100
int main()
{
    int n, a[size], b[size];
    int start[2*size+1], end[2*size+1];
    int sum1 = 0, sum2 = 0, i;
    int diff[size];
    printf("Enter n: ");
    scanf("%d", &n);
    for(i = 0; i < n; i++)
    {
        printf("Enter a[%d]: ", i);
        scanf("%d", &a[i]);
    }
    for(i = 0; i < n; i++)
    {
        printf("Enter b[%d]: ", i);
        scanf("%d", &b[i]);
    }

    for(i = 0; i < n; i++)
    {
        if(a[i]) sum1++;
        if(b[i]) sum2++;
        diff[i] = sum1 - sum2;
    }
    for(i = 0; i < 2*n; i++)
        start[i] = -1, end[i] = -1;
    start[n] = end[n] = 0;
    //initially sum is 0 at the beginning of array and
    //the first n-1 elements of start and end are used
    //if sum of A till ith element is less than sum of B till ith element
    for(i=0; i < n; i++)
    {
        if(start[diff[i] + n] == -1)//interested only in the first occurrence of diff[i]
            start[diff[i] + n] = i;
        end[diff[i] + n] = i;//interested in the last occurrence of diff[i]
    }
    int max = -1;
    int savei = -1; //savei is for storing the sum having the largest span

    for(i = 0; i < 2*n; i++)
    {
        if(start[i] > -1 && (end[i] - start[i] > max))
        {
            max = end[i] - start[i];
            savei = i;
        }
    }
    if(savei >= 0)
    {
        printf("The largest span is from %d to %d\n", start[savei]+(savei != n), end[savei]);
        //when sum zero is having the largest span, span starts from first element itself.
        //Else, the span starts from the next element from which the span does not change
    }
    else
    {
        printf("No span\n");
    }
}
```

5 votes

-- Arjun Suresh (287k points)

1.1.5 Algorithm Design: TIFR2011-B-29 [top](#)

<http://gateoverflow.in/20576>



I think its Tower of Hanoi problem.

Therefore Total number of function call $2^n - 1 = 1023$ option B

4 votes

-- Umang Raman (14.5k points)

1.2

Algorithm Design Techniques(4) [top](#)

1.2.1 Algorithm Design Techniques: GATE1997-1.5 [top](#)

<http://gateoverflow.in/2221>

The correct matching for the following pairs is

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

[Answer](#)

1.2.2 Algorithm Design Techniques: GATE2015-1-6 [top](#)

<http://gateoverflow.in/8088>

Match the following:

- | | | |
|-------------------------------------|--------------------------|--------------------------|
| P. Prim's algorithm for i. | minimum spanning tree | Backtracking |
| Q. Floyd-Warshall algorithm for ii. | all pairs shortest paths | Greedy method |
| R. Mergesort | | 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](#)

[Answer](#)

1.2.3 Algorithm Design Techniques: GATE2015-2-36 [top](#)

<http://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

Answer

1.2.4 Algorithm Design Techniques: GATE2017-1-05 [top](#)

<http://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

Answer

Answers: Algorithm Design Techniques

1.2.1 Algorithm Design Techniques: GATE1997-1.5 [top](#)

<http://gateoverflow.in/2221>



Selected Answer

answer B

7 votes

-- ankitrokdeonsns (9.1k points)

1.2.2 Algorithm Design Techniques: GATE2015-1-6 [top](#)

<http://gateoverflow.in/8088>



Selected Answer

option c is correct ..

10 votes

-- Anoop Sonkar (4.8k points)

1.2.3 Algorithm Design Techniques: GATE2015-2-36 [top](#)

<http://gateoverflow.in/8161>

Answer: C

12 votes

-- Rajarshi Sarkar (34.4k points)

1.2.4 Algorithm Design Techniques: GATE2017-1-05 [top](#)<http://gateoverflow.in/118707>

Selected Answer

in KRUSKAL in every iteration, an edge of the MOST MINIMUM WEIGHT (GREEDIEST) possible is selected and added to MST construction. Hence, GREEDY.

in QUICKSORT we partition the problem in to subproblems, solve them and the combine. Hence it is DIVIDE and CONQUER

Floyd-Warshal uses Dynamic programming.

Hence correct answer is

OPTION (C)

8 votes

-- sriv_shubham (2.4k points)

1.3**Asymptotic Notations(21)** [top](#)<http://gateoverflow.in/2466>**1.3.1 Asymptotic Notations: GATE1994-1.23** [top](#)

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](#)

Answer

1.3.2 Asymptotic Notations: GATE1996-1.11 [top](#)<http://gateoverflow.in/2715>

Which of the following is false?

- A. $100n \log n = O\left(\frac{n \log n}{100}\right)$
- 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](#)

Answer

1.3.3 Asymptotic Notations: GATE1999-2.21 [top](#)<http://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) $T = O(n^2)$ |
| (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](#)

[Answer](#)

1.3.4 Asymptotic Notations: GATE2000-2.17 [top](#)

<http://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](#)

[Answer](#)

1.3.5 Asymptotic Notations: GATE2001-1.16 [top](#)

<http://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 $g(n) = O(f(n))$

[gate2001](#) [algorithms](#) [asymptotic-notations](#) [time-complexity](#) [normal](#)

[Answer](#)

1.3.6 Asymptotic Notations: GATE2003-20 [top](#)

<http://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

[Answer](#)

1.3.7 Asymptotic Notations: GATE2004-29 [top](#)

<http://gateoverflow.in/1028>

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

[Answer](#)

1.3.8 Asymptotic Notations: GATE2004-IT-55 [top](#)

<http://gateoverflow.in/3698>

Let $f(n)$, $g(n)$ and $h(n)$ be functions defined for positive inter 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

[Answer](#)

1.3.9 Asymptotic Notations: GATE2005-37 [top](#)

<http://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

[Answer](#)

1.3.10 Asymptotic Notations: GATE2008-39 [top](#)

<http://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

Answer

1.3.11 Asymptotic Notations: GATE2008-IT-10 [top](#)

<http://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

Answer

1.3.12 Asymptotic Notations: GATE2011-37 [top](#)

<http://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

Answer

1.3.13 Asymptotic Notations: GATE2012-18 [top](#)

<http://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. $A(n) = O(W(n))$
- D. $A(n) = o(W(n))$

gate2012 algorithms easy asymptotic-notations

Answer

1.3.14 Asymptotic Notations: GATE2015-3-4 [top](#)

<http://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

Answer

1.3.15 Asymptotic Notations: GATE2015-3-42 [top](#)

<http://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

Answer

1.3.16 Asymptotic Notations: GATE2017-1-04 [top](#)

<http://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

Answer

1.3.17 Asymptotic Notations: TIFR2011-B-27 [top](#)

<http://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

Answer**1.3.18 Asymptotic Notations: TIFR2012-B-6** [top](#)<http://gateoverflow.in/25108>

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. $\frac{n}{\log n} < 2\sqrt{2 \log n} < n^{1/3}$

[tifr2012](#) | [algorithms](#) | [asymptotic-notations](#)**Answer****1.3.19 Asymptotic Notations: TIFR2014-B-8** [top](#)<http://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](#)**Answer****1.3.20 Asymptotic Notations: TIFR2016-B-7** [top](#)<http://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 $m = O((\log^2 n))$
- E. $m = \Theta(\log^{1.5} n)$

[tifr2016](#) | [asymptotic-notations](#)**Answer****1.3.21 Asymptotic Notations: TIFR2017-A-4** [top](#)<http://gateoverflow.in/84943>

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](#)**Answer****Answers: Asymptotic Notations****1.3.1 Asymptotic Notations: GATE1994-1.23** [top](#)<http://gateoverflow.in/2466>



Selected Answer

For asymptotic complexity, we assume sufficiently large n . So, $g_1(n) = n^2$ and $g_2(n) = n^3$. Growth rate of g_1 is less than that of g_2 . i.e., $g_1(n) = O(g_2(n))$.

Options A and B are true here.

10 votes

-- Arjun Suresh (287k points)

1.3.2 Asymptotic Notations: GATE1996-1.11 top

<http://gateoverflow.in/2715>



Selected Answer

- $100n \log n = (\frac{n \log n}{100})$: cant do comment about it not given properly in paper.
- $\sqrt{\log n} = O(\log \log n)$: false take any long value like 256 LHS results 16 But RHS results 4 only . gerenraly we take log left side but that is wrong.
- $0 < x < y$ then $n^x = O(n^y)$: true since y is always greater than x so RHS is always greater than LHS.
- $2^n \neq O(nk)$: true since k is constant .so for large value of n LHS is very higher than RHS (exponential function always greater than linear)

Only B is false

9 votes

-- Prashant Singh (47.6k points)

1.3.3 Asymptotic Notations: GATE1999-2.21 top

<http://gateoverflow.in/1498>



Selected Answer

$$(M) T(n) = \text{Sum of first } n \text{ natural numbers} = \frac{n(n+1)}{2} = O(n^2)$$

(N) $T(n) = \Theta(n) = O(n)$, third case of Master theorem

$(f(n) = n = \Omega(n^{\log_b a + \epsilon})) = \Omega(n^{\log_2 1 + \epsilon}) = \Omega(n^{0+\epsilon})$, satisfied for any positive $\epsilon \leq 1$. Also, $af(\frac{n}{b}) < cf(n) \implies f(\frac{n}{2}) < cf(n) \implies \frac{n}{2} < cn$, satisfied for any c between 0 and 0.5)

(O) $T(n) = \Theta(n \log n) = O(n \log n)$, third case of Master theorem

$(f(n) = n \log n = \Omega(n^{\log_b a + \epsilon})) = \Omega(n^{\log_2 1 + \epsilon}) = \Omega(n^{0.5+\epsilon})$, satisfied for positive $\epsilon = 0.5$. Also, $af(\frac{n}{b}) < cf(n) \implies f(\frac{n}{2}) < cf(n) \implies \frac{n}{2} \log \frac{n}{2} < cn \log n$, satisfied for $c = 0.5$)

(P) Like in (M), here we are adding the log of the first n natural numbers. So,

$$T_n = \log 1 + \log 2 + \log 3 + \dots + \log n$$

$$= \log(1 \times 2 \times \dots \times n)$$

$$= \log(n!)$$

= $\Theta(n \log n)$ (Stirling's Approximation)

11 votes

-- Arjun Suresh (287k points)

1.3.4 Asymptotic Notations: GATE2000-2.17 top

<http://gateoverflow.in/654>



Selected Answer

	$n = 256$	$n = 65536$

$f(n) = 3n^{\sqrt{n}}$	3×256^{16} $= 3 \times 2^{128}$	3×65536^{256} $= 3 \times 2^{16 \times 256}$ $= 3 \times 2^{4096}$
$g(n) = 2^{\sqrt{n} \log_2 n}$	$2^{16 \times 8}$ $= 2^{128}$	$2^{256 \times 16}$ $= 2^{4096}$
$h(n) = n!$	$256!$ $= O((2^8)^{256})$ $= O(2^{2048})$	$65536!$ $= O((2^{16})^{65536})$ $= O(2^{1M})$

Case of $h(n)$ is given only by an upper bound but factorial has higher growth rate than exponential.

<http://math.stackexchange.com/questions/351815/do-factorials-really-grow-faster-than-exponential-functions>

$f(n)$ and $g(n)$ are having same order of growth as $f(n)$ is simply 3 $g(n)$ (we can prove this by taking log also). So, (d) is correct and all other choices are false.

16 votes

-- Arjun Suresh (287k points)

1.3.5 Asymptotic Notations: GATE2001-1.16 [top](#)

<http://gateoverflow.in/709>



Selected Answer

	f(n)	g(n)
n = 2¹⁰	$10 * 2^{10} * 2^{10}$	$2^{10} * 10^{10}$
n = 2²⁵⁶	$256 * 2^{256} * 2^{256}$	$2^{256} * 256^{10}$

So, as n is going larger, $f(n)$ is overtaking $g(n)$ and the growth rate of f is faster than that of g . So, $g(n) = O(f(n))$ and $f(n) \neq O(g(n))$.

B choice.

17 votes

-- Arjun Suresh (287k points)

1.3.6 Asymptotic Notations: GATE2003-20 [top](#)

<http://gateoverflow.in/910>



Selected Answer

1) Clearly rate of growth of $(n + k)^m = n^m$ as k and m are constants

so TRUE

2) $2^{n+1} = 2 * (2^n) = \Theta(2^n)$ as 2 is a constant here

As 2^{n+1} is both upper and lower bounded by 2^n we can say $2^{n+1} = O(2^n)$

so TRUE

3) 2^{2n+1} has same rate of growth as 2^{2n}

$2^{2n} = 2^n 2^n$

2^n is upper bounded by $(2^n)^2$, not the other way round

so FALSE

 11 votes

-- Danish (3.5k points)

1.3.7 Asymptotic Notations: GATE2004-29 [top](#)

<http://gateoverflow.in/1028>

Selected Answer

For comparison-based sorting the asymptotically tight bound for worst case is given by $\Theta(n \log n)$, which means it is the tightest upper bound (big O) as well as the tightest lower bound (big omega). So, answer is $n \log n$.

Tightest lower bound of sorting (say $S(n)$) is $n \log n$ means there is no function f which has an order of growth larger than $n \log n$ and $f(n) = \Omega(S(n))$ holds.

A usual mistake is to think worst case changes with lower and upper bounds, but that is not the case. Worst case is defined for the algorithm and it is always the input which causes the algorithm the maximum complexity.

 15 votes

-- Arjun Suresh (287k points)

1.3.8 Asymptotic Notations: GATE2004-IT-55 [top](#)

<http://gateoverflow.in/3698>

Answer is (D)

We can verify as : $f \leq g$ BUT $g \neq f$. Therefore $f < g$

Also $g = h$ as $g = O(h)$ and $h = O(g)$

 6 votes

-- Sandeep_Uniyal (7.3k points)

1.3.9 Asymptotic Notations: GATE2005-37 [top](#)

<http://gateoverflow.in/1373>

Selected Answer

Applying Masters theorem $T(n) = \Theta(n \log n)$ So, it can't be $\Omega(n^2)$

Hence answer is C.

 11 votes

-- shreya ghosh (3.4k points)

1.3.10 Asymptotic Notations: GATE2008-39 [top](#)

<http://gateoverflow.in/450>

Selected Answer

$g(n) = n!$ on expanding factorial we get $g(n) = \mathcal{O}(n^n)$:

$$\begin{aligned} n^n &> n^{\log n} \\ n^n &> 2^n \end{aligned}$$

this condition is violated by option A, B and C by first statements of each Hence, they cannot be said true.

second statement of option D says that $g(n)$ is asymptotically biggest of all.

answer = **option D**

 5 votes

-- Amar Vashishth (27.9k points)

1.3.11 Asymptotic Notations: GATE2008-IT-10 [top](#)

<http://gateoverflow.in/3270>

Selected Answer

A < C and A < D

E < B

and

C, D < E as E is exponential function.

Now, we just need to see if C or D is larger.

In C we have a term $n^{3/4}$ and correspondingly in D we have $\log^9 n$ (after taking n out).

$n^{3/4}$ is asymptotically larger than $\log^9 n$ as when $n = 10^{100}$, $\log^9 n$ gives 100^9 , while $n^{3/4}$ gives $10^{75} > 100^{37}$ a much higher value and this is true for all higher values of n . So, D < C.

Thus A is correct.

13 votes

-- Arjun Suresh (287k points)

1.3.12 Asymptotic Notations: GATE2011-37 [top](#)

<http://gateoverflow.in/2139>



Selected Answer

[EDIT]

answer A

$n \log_2 n < n^{3/2}$ is quite straightforward

also $n^{3/2} < n^{\log_2 n}$ and $n^{3/2} < 2^n$

now only $n^{\log_2 n}$ and 2^n need to be compared

taking log of both $(\log_2 n)^2$ and n

$n > (\log_2 n)^2$

hence $2^n > n^{\log_2 n}$

12 votes

-- ankitrokdeonsns (9.1k points)

1.3.13 Asymptotic Notations: GATE2012-18 [top](#)

<http://gateoverflow.in/50>



Selected Answer

Worst case complexity can never be lower than the average case complexity, but it can be higher. So, (C) is the answer.

$$A(n) = O(W(n))$$

14 votes

-- Arjun Suresh (287k points)

1.3.14 Asymptotic Notations: GATE2015-3-4 [top](#)

<http://gateoverflow.in/8398>



Selected Answer

Sum of the cubes of the first n natural numbers is given by $(n(n+1)/2)^2$ which is $\Theta(n^4)$. So, I, III and IV are correct. II is wrong. C choice.

21 votes

-- Arjun Suresh (287k points)

1.3.15 Asymptotic Notations: GATE2015-3-42 [top](#)

<http://gateoverflow.in/8501>



The answer is option D.

Since the value of $\sin(n)$ will always range from -1 to +1, hence $g(n)$ can take values 1, n, n^2 .

Hence, if $g(n) = 1$, Statement I is incorrect.

And, if $g(n) = n^2$, then Statement II is incorrect.

22 votes

-- saurabhrk (1.5k points)

1.3.16 Asymptotic Notations: GATE2017-1-04 [top](#)

<http://gateoverflow.in/118703>

→ Comparing \sqrt{n} vs $\log_2 n$.
 Square it → n vs $\log_2 n * \log_2 n$.
 put $n = 2^{10}$ → $2^{10} * 2^{10}$ vs $\log_2 2^{10} * \log_2 2^{10}$.
 $\Rightarrow 2^{10} * 2^{10}$ vs $10 * 10$.
 So here \sqrt{n} wins.

→ $\frac{100}{n}$ vs $\log_2 n$. | put $n = 2^{10}$ | $\frac{100}{2^{10}}$ vs $\underline{\underline{20}}$
 $\log_2 n$ wins

→ 10 vs $\frac{100}{n}$. put $n = 2^{10}$ | 10 vs $\frac{100}{2^{10}}$
 So here 10 wins

So ↑↑ order of asymptotic comp →

Option B → $\frac{100}{n}, 10, \log_2 n, \sqrt{n}, n$

Here I'm using the term win but that I don't think is right.. I have been using this for my own understanding purposes..

3 votes

-- Joker (1.4k points)

1.3.17 Asymptotic Notations: TIFR2011-B-27 [top](#)

<http://gateoverflow.in/20573>



Let $n = 2^x$. Then, $\log_2 n = x$

$$f(n) = n^{1/\sqrt{\log_2 n}} = (2^x)^{1/\sqrt{x}} = 2^{x/\sqrt{x}} = 2^{\sqrt{x}}$$

$$g(n) = \sqrt{\log_2 n} = \sqrt{\log_2(2^x)} = \sqrt{x}$$

$$h(n) = n^{1/100} = (2^x)^{1/100} = 2^{x/100}$$

Since exponentials grow faster than polynomials, $h(n) > g(n)$ for large n .

Since linear functions grow faster than square roots, $\frac{x}{100} > \sqrt{x}$ for large x . Thus, $h(n) > f(n)$ for large n .

Since exponentials grow faster than polynomials, $2^{\sqrt{x}} > \sqrt{x}$ for large \sqrt{x} . Thus, $f(n) > g(n)$ for large n .

Hence, the relation is,

$$g(n) < f(n) < h(n)$$

Thus, option D is correct.

6 votes

-- Pragy Agarwal (19.3k points)

1.3.18 Asymptotic Notations: TIFR2012-B-6 [top](#)

<http://gateoverflow.in/25108>



Selected Answer

Ans will be C

Take $n = 2^{1024}$

Now, $2^{\sqrt{2\log n}} \approx 2^{45}$

$$n^{\frac{1}{3}} \approx 2^{341}$$

$$n/\log n = 2^{1024}/1024 \approx 2^{1014}$$

Just one value is not enough to confirm growth rate. So, take $n = 1024$.

Now, $2^{\sqrt{2\log n}} \approx 2^4$

$$n^{\frac{1}{3}} \approx 2^3$$

$$n/\log n = 2^{10}/10 \approx 2^7$$

So, as n increases the gap between second and third function increases and also the second function overtakes the first. So, $f1 < f2 < f3$.

7 votes

-- srestha (55.2k points)

1.3.19 Asymptotic Notations: TIFR2014-B-8 [top](#)

<http://gateoverflow.in/27192>



Selected Answer

Assuming that the base of the log in the question is e .

Let us try to rewrite each of these functions in the form $e^{\text{something}}$, to make the comparison easier.

$$\begin{array}{lll} a. & e^n / n & = \frac{e^n}{e^{\ln n}} \\ & & = e^{(n - \ln n)} \\ b. & e^{n-0.9\ln n} & = e^{(n - 0.9\ln n)} \\ c. & 2^n & = (e^{\ln 2})^n \\ & & = e^{(n \ln 2)} \\ d. & (\ln n)^{n-1} & = (e^{\ln \ln n})^{n-1} \\ & & = e^{(n \ln \ln n - \ln \ln n)} \end{array}$$

Now, if we just compare the exponents of all, we can clearly see that $(n \ln \ln n - \ln \ln n)$ grows faster than the rest. Note that in option c. the multiplicative $\ln 2$ is a constant, and hence grows slower than the multiplicative $\ln \ln n$ from option d.

This implies that $e^{(n \ln \ln n - \ln \ln n)}$ grows the fastest, and hence, $(\ln n)^{n-1}$ grows the fastest.

Thus, option d. is the correct answer.

9 votes

-- Pragy Agarwal (19.3k points)

1.3.20 Asymptotic Notations: TIFR2016-B-7 top

<http://gateoverflow.in/30720>

m	n=m!	log n / log log n
4	24	$4/2 = 2$
6	720	$9/3 = 3$
8	$720 * 56$	$15/3 = 5$
10	$720 * 56 * 90$	$21/4 = 5$

If we see, m is growing at same rate as log n / log log n.

2 votes

-- Arjun Suresh (287k points)

1.3.21 Asymptotic Notations: TIFR2017-A-4 top

<http://gateoverflow.in/9494>



Selected Answer

Let $N = 2^{256}$

- $(\log \log N)! = 8!$
- $(\log \log N)^{\log N} = (8)^{256} = 2^{768}$
- $(\log \log N)^{\log \log \log N} = 8^3 = 512$
- $(\log N)^{\log \log N} = (256)^8 = 2^{64}$
- $2^{\sqrt{\log \log N}} = 2^{2\sqrt{2}}$

Let $N = 2^{16}$

- $(\log \log N)! = 4!$
- $(\log \log N)^{\log N} = (4)^{16} = 2^{32}$
- $(\log \log N)^{\log \log \log N} = 4^2 = 16$
- $(\log N)^{\log \log N} = (16)^4 = 2^{16}$
- $2^{\sqrt{\log \log N}} = 2^2 = 4$

Taking ratio for both N values,

- $(\log \log N)! \rightarrow 8!/4!$
- $(\log \log N)^{\log N} \rightarrow 2^{736}$
- $(\log \log N)^{\log \log \log N} \rightarrow 32$
- $(\log N)^{\log \log N} \rightarrow 2^{48}$
- $2^{\sqrt{\log \log N}} \rightarrow \approx 2$

Option B = $(\log \log N)^{\log N}$ asymptotically grows the fastest as N goes to infinity as for the same change in N, the value increased the most (growth) for option B.

9 votes

-- Kapil Phulwani (47.5k points)

1.4

Dynamic Programming(11) top

1.4.1 Dynamic Programming: GATE 2016-2-14 [top](#)

<http://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.

[gate2016-2](#) | [algorithms](#) | [dynamic-programming](#) | [easy](#)

[Answer](#)

1.4.2 Dynamic Programming: GATE 2016-2-38 [top](#)

<http://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 $A_1 A_2 A_3 A_4$ using the basic matrix multiplication method is _____.

[gate2016-2](#) | [dynamic-programming](#) | [normal](#) | [numerical-answers](#)

[Answer](#)

1.4.3 Dynamic Programming: GATE2008-80 [top](#)

<http://gateoverflow.in/498>

The subset-sum problem is defined as follows. Given a set of n positive integers, $S = \{a_1, a_2, a_3, \dots, a_n\}$, 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 $\{a_1, a_2, \dots, a_i\}$ 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. $X[i, j] = X[i - 1, j] \vee X[i, j - a_i]$
- 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](#)

[Answer](#)

1.4.4 Dynamic Programming: GATE2008-81 [top](#)

<http://gateoverflow.in/43464>

The subset-sum problem is defined as follows. Given a set of n positive integers, $S = \{a_1, a_2, a_3, \dots, a_n\}$, 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 $\{a_1, a_2, \dots, a_i\}$ 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](#)

[Answer](#)

1.4.5 Dynamic Programming: GATE2009-53 [top](#)

<http://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:

```
 $I(i,j) = 0, \text{ if either } i = 0 \text{ or } j = 0$ 
 $= \text{expr1, if } i,j > 0 \text{ and } X[i-1] = Y[j-1]$ 
 $= \text{expr2, if } i,j > 0 \text{ and } X[i-1] \neq Y[j-1]$ 
```

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

Answer

1.4.6 Dynamic Programming: GATE2009-54 [top](#)

<http://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:

```
 $I(i,j) = 0, \text{ if either } i = 0 \text{ or } j = 0$ 
 $= \text{expr1, if } i,j > 0 \text{ and } X[i-1] = Y[j-1]$ 
 $= \text{expr2, if } i,j > 0 \text{ and } X[i-1] \neq Y[j-1]$ 
```

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

Answer

1.4.7 Dynamic Programming: GATE2010-34 [top](#)

<http://gateoverflow.in/2208>

The weight of a sequence a_0, a_1, \dots, a_{n-1} of real numbers is defined as $a_0 + a_1/2 + \dots + a_{n-1}/2^{n-1}$. 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_0, a_1, \dots, a_{n-1} and Y the maximum possible weight of a subsequence of a_1, a_2, \dots, a_{n-1} . Then X is equal to

- 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

Answer

1.4.8 Dynamic Programming: GATE2011-25 [top](#)

<http://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 $\text{Max}(L_0, L_1, \dots, L_{n-1})$.

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](#)

[Answer](#)

1.4.9 Dynamic Programming: GATE2011_38 [top](#)

<http://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](#)

[Answer](#)

1.4.10 Dynamic Programming: GATE2014-2-37 [top](#)

<http://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 $x + 10y = \underline{\hspace{2cm}}$.

[gate2014-2](#) [algorithms](#) [normal](#) [numerical-answers](#) [dynamic-programming](#)

[Answer](#)

1.4.11 Dynamic Programming: GATE2014-3-37 [top](#)

<http://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 $\underline{\hspace{2cm}}$.

[gate2014-3](#) [algorithms](#) [normal](#) [numerical-answers](#) [dynamic-programming](#)

[Answer](#)

Answers: Dynamic Programming

1.4.1 Dynamic Programming: GATE 2016-2-14 [top](#)

<http://gateoverflow.in/39570>



Selected Answer

In floyd warshalls we calculate all possibilities and select best one so its neither dac nor greedy but based on Dynamic Programming Paradigm.

16 votes

-- Anurag Semwal (7.6k points)

1.4.2 Dynamic Programming: GATE 2016-2-38 [top](#)



Selected Answer

Answer is 1500 !

Matrix Paranthesizing => A1 ((A2 A3) A4)

Check my solution below, using dynamic programming (There was little mistake while writing in parentheses in this image, ignore it Check parenthesis above) =>

The handwritten notes show the following steps:

- Dimensions: A₁ 10x5, A₂ 5x20, A₃ 20x10, A₄ 10x5.
- Calculations for individual matrix products:
 - A₁₂ = 10x5x20 = 1000
 - A₂₃ = 5x20x10 = 1000
 - A₃₄ = 20x10x5 = 1000
- Calculation for A₁₃ using dynamic programming:

$$A_{13} = \min \begin{cases} A_{12} + A_{23} + 10 \times 20 \times 10 = 2000 \\ A_{11} + A_{22} + 10 \times 5 \times 10 = 1500 \\ 0 \quad 1000 \end{cases}$$
- Calculation for A₂₄ using dynamic programming:

$$A_{24} = \min \begin{cases} A_{23} + A_{34} + 5 \times 10 \times 5 = 1250 \\ A_{22} + A_{34} + 5 \times 20 \times 5 = 1500 \\ 0 \quad 1000 \end{cases}$$
- Calculation for A₁₄ using dynamic programming:

$$A_{14} = \min \begin{cases} A_{11} + A_{24} + 10 \times 5 \times 5 = 1500 \\ A_{12} + A_{34} + 10 \times 20 \times 5 > 2000 \\ A_{13} + A_{24} + 10 \times 10 \times 5 = 2000 \\ 0 \quad 1500 \end{cases}$$
- Final answer: Ans = ~~A₁₁ + A₂₄~~ 1500
- Final expression: A₁₄
(A₁(A₂ A₃) A₄)

14 votes

-- Akash (41.5k points)

1.4.3 Dynamic Programming: GATE2008-80 [top](#)

I think answers are

80. (B)

$$X[i][j] = X[i-1][j] \cup X[i-1][j-a_i]$$

We calculate the value of $X[i][j]$ as if we include the value

a_i in the subset $X[i-1][j-a_i]$ or we do not include the value in the subset $X[i-1][j]$.

81. (C)

```

// Returns true if there is a subset of set[] with sum equal to given sum
bool isSubsetSum(int set[], int n, int sum)
{
    // The value of subset[i][j] will be true if there is a subset of set[0..j-1]
    // with sum equal to i
    bool subset[sum+1][n+1];

    // If sum is 0, then answer is true
    for (int i = 0; i <= n; i++)
        subset[0][i] = true;

    // If sum is not 0 and set is empty, then answer is false
    for (int i = 1; i <= sum; i++)
        subset[i][0] = false;

    // Fill the subset table in bottom up manner
    for (int i = 1; i <= sum; i++)
    {
        for (int j = 1; j <= n; j++)
        {
            subset[i][j] = subset[i][j-1];
            if (i >= set[j-1])
                subset[i][j] = subset[i][j] || subset[i - set[j-1]][j-1];
        }
    }

    /* // uncomment this code to print table
    for (int i = 0; i <= sum; i++)
    {
        for (int j = 0; j <= n; j++)
            printf ("%4d", subset[i][j]);
        printf ("\n");
    } */

    return subset[sum][n];
}

```

4 votes

-- Sona Praneeth Akula (4k points)

1.4.4 Dynamic Programming: GATE2008-81 [top](#)

<http://gateoverflow.in/43484>

Selected Answer

ANS) C ,If LAST ROW and LAST COLUMN entry is 1, then there exists subset whose elements sum to W

4 votes

-- Shivam Bhardwaj (167 points)

1.4.5 Dynamic Programming: GATE2009-53 [top](#)

<http://gateoverflow.in/1338>

Selected Answer

Answer is C. When the currently compared elements doesn't match, we have two possibilities for the LCS, one including X[i] but not Y[j] and other including Y[j] but not X[i].

15 votes

-- Akash (41.5k points)

Answer is C. When the currently compared elements doesn't match, we have two possibilities for the LCS, one including X[i] but not Y[j] and other including Y[j] but not X[i].

```

/* Returns length of LCS for X[0..m-1], Y[0..n-1] */
int lcs( char *X, char *Y, int m, int n )
{
    if (m == 0 || n == 0)
        return 0;
    if (X[m-1] == Y[n-1])
        return 1 + lcs(X, Y, m-1, n-1);
    else
        return max(lcs(X, Y, m, n-1), lcs(X, Y, m-1, n));
}

```

10 votes

-- Sona Praneeth Akula (4k points)

1.4.6 Dynamic Programming: GATE2009-54 [top](#)

<http://gateoverflow.in/43476>



Selected Answer

$$\text{expr2} = \max(l(i-1, j), l(i, j-1))$$

When the currently compared elements doesn't match, we have two possibilities for the LCS, one including $X[i]$ but not $Y[j]$ and other including $Y[j]$ but not $X[i]$.

```
/* Returns length of LCS for X[0..m-1], Y[0..n-1] */
int lcs( char *X, char *Y, int m, int n )
{
    if (m == 0 || n == 0)
        return 0;
    if (X[m-1] == Y[n-1])
        return 1 + lcs(X, Y, m-1, n-1);
    else
        return max(lcs(X, Y, m, n-1), lcs(X, Y, m-1, n));
}
```

54. Answer is **B**. Dynamic programming is used to save the previously found LCS. So, for any index $[p,q]$ all smaller ones should have been computed earlier. Option D is not correct as the condition given requires even $L[3,2]$ to be computed before $L[2,4]$ which is not a necessity if we follow row-major order.

```
int lcs( char *X, char *Y, int m, int n )
{
    int L[m+1][n+1];
    int i, j;

    /* Following steps build L[m+1][n+1] in bottom up fashion. Note
       that L[i][j] contains length of LCS of X[0..i-1] and Y[0..j-1] */
    for (i=0; i<=m; i++)
    {
        for (j=0; j<=n; j++)
        {
            if (i == 0 || j == 0)
                L[i][j] = 0;

            else if (X[i-1] == Y[j-1])
                L[i][j] = L[i-1][j-1] + 1;

            else
                L[i][j] = max(L[i-1][j], L[i][j-1]);
        }
    }

    /* L[m][n] contains length of LCS for X[0..n-1] and Y[0..m-1] */
    return L[m][n];
}
```

8 votes

-- Arjun Suresh (287k points)

$$\text{expr2} = \max(l(i-1, j), l(i, j-1))$$

When the currently compared elements doesn't match, we have two possibilities for the LCS, one including $X[i]$ but not $Y[j]$ and other including $Y[j]$ but not $X[i]$.

Answer is **B**. We can either use Row Major or column major order.

Issue of option D -> Read option D carefully.

$L[p,q]$ needs to be computed before $L[r,s]$ if either $p < q$ or $r < s$

Assuming that we want to compute $L(3,3)$. We need not compute $L(4,2)$ if we are using Row Major Order ! Here $L(4,2) = L[p,q] \& L(3,3) = L[r,s]$. Then $q < s$ still we need not compute it ! so **D IS FALSE**

12 votes

-- Akash (41.5k points)

1.4.7 Dynamic Programming: GATE2010-34 [top](#)

<http://gateoverflow.in/2208>



Selected Answer

$$\begin{aligned} S &= \langle a_0, S_1 \rangle \\ S_1 &= \langle a_1, a_2, a_3 \dots a_{n-1} \rangle \end{aligned}$$

Two possible cases arise:

1. a_0 is included in the max weight subsequence of S :

In this case, $X = \text{weight}(\langle a_0, S_1 \rangle) = a_0 + \frac{Y}{2}$

2. a_0 is not included in the max weight subsequence of S :

In this case, $X = \text{weight}(S_1) = Y$

Since the value of a_0 can be anything (negative or $< \frac{Y}{2}$ in general) $\{\because a_i \in \mathbb{R}\}$, it is possible that $Y > a_0 + \frac{Y}{2}$.

The maximum possible weight of a subsequence of S is given by:

$$X = \max \left(Y, a_0 + \frac{Y}{2} \right)$$

Thus, option B is correct.

28 votes

-- Pragy Agarwal (19.3k points)

1.4.8 Dynamic Programming: GATE2011-25 [top](#)

<http://gateoverflow.in/2127>



Selected Answer

(A) is the answer.

The algorithm is storing the optimal solutions to subproblems at each point (for each i), and then using it to derive the optimal solution of a bigger problem. And that is dynamic programming approach. And the program has linear time complexity.

<http://stackoverflow.com/questions/1065433/what-is-dynamic-programming>

Now, branch and bound comes when we explore all possible solutions (branch) and backtracks as soon as we find we won't get a solution (in classical backtracking we will retreat only when we won't find the solution). So, backtracking gives all possible solutions while branch and bound will give only the optimal one. <http://www.cs.cornell.edu/~wdtseng/icpc/notes/bt2.pdf>

The given algorithm here is neither backtracking nor branch and bound. Because we are not branching anywhere in the solution space.

And the algorithm is not divide and conquer as we are not dividing the problem and then merging the solution as in the case of merge sort (where merge is the conquer step).

https://en.wikipedia.org/wiki/Divide_and_conquer_algorithms

20 votes

-- Arjun Suresh (287k points)

1.4.9 Dynamic Programming: GATE2011_38 [top](#)

<http://gateoverflow.in/2140>



Selected Answer

Answer is C.

Ordering: First Multiply $M_2 \times M_3$. This requires $100*20*5$ multiplications.

Then Multiply $M_1 \times (M_2 \times M_3)$. This requires $10*100*5$ multiplications.

Then Multiply $(M_1 \times (M_2 \times M_3)) \times M_4$. This requires $10*5*8$ multiplications.

Total 19000 Multiplications.

Brute Force approach - anyone can do.

No. of possible ordering for 4 matrices is C_3 where C_3 is the 3rd Catalan number and given by $n = 3$ in $\frac{1}{n+1} {}^{2n}C_n = 5$.

So, here we have

1. $(M_1 \times M_2) \times (M_3 \times M_4)$
2. $(M_1 \times (M_2 \times M_3)) \times M_4$
3. $((M_1 \times M_2) \times M_3) \times M_4$
4. $M_1 \times (M_2 \times (M_3 \times M_4))$
5. $M_1 \times ((M_2 \times M_3) \times M_4)$

Each of these would give no. of multiplications required as

1. $pqr + rst + prt$
2. $qrs + pqs + pst$
3. $pqr + prs + pst$
4. $rst + qrt + pqt$
5. $qrs + qst + pst$

The last 2 are having qt terms which are the highest terms by far and hence we can avoid them from consideration. $qt = 8000$ multiplied by one other term would be larger than any value in choice. So, just find the value of first 3 terms.

1. $pqr + rst + prt = 20000 + 8000 + 16000 = 44000$
2. $qrs + pqs + pst = 10000 + 5000 + 4000 = 19000$ - smallest value in choice, we can stop here.
3. $pqr + prs + pst$

Dynamic Programming Solution (should know [Matrix Chain Ordering algorithm](#))

Here we have a chain of length 4.

Dynamic programming solution of Matrix chain ordering has the solution

$$m[i, j] = \begin{cases} 0 & \text{if } i = j \\ \min_{i \leq k < j} m[i][k] + m[k+1][j] + p_{i-1}p_jp_k & \text{if } i < j \end{cases}$$

So, we can fill the following table starting with the diagonals and moving upward diagonally. Here $k < j$ but $\geq i$.

	j=1	j=2	j=3	j=4
i=1	$p_0 p_1 p_2 = 20000$	$\min(10000 + p_0 p_1 p_3, 20000 + p_0 + p_2 p_3) = 15000$	$\min(18000 + p_0 p_1 p_4, 20000 + 8000 + p_0 + p_2 + p_4, 15000 + p_0 p_3 p_4) = 19000$	
i=2	0	$p_1 p_2 p_3 = 10000$	$\min(10000 + p_2 p_3 p_4, p_1 p_3 p_4) = 18000$	
i=3		0	$p_2 p_3 p_4 = 8000$	
i=4			0	

Our required answer is given by $m[1, 4] = 19000$.

10 votes

-- Sona Praneeth Akula (4k points)

1.4.10 Dynamic Programming: GATE2014-2-37 [top](#)

<http://gateoverflow.in/1996>

Pqrr

qpqr

In first string

If we want to get 4 as max len den lcs should end with either rr or qr

Only 4 combinations possible for lcs with len 4

qpqr

qqrr

pqrr

qrrr

Now check for matching sequences in second string, except for qqrr all possible

7 votes

-- Anurag Semwal (7.6k points)

1.4.11 Dynamic Programming: GATE2014-3-37 [top](#)

<http://gateoverflow.in/2071>



Selected Answer

$$T(9) = \text{Distance from 9 to 100}$$

$$T(9) = 1 + \min(T(y), T(z)) = 1 + \min(\text{Distance from } y \text{ to 100}, \text{Distance from } z \text{ to 100})$$

There are only two such values where we can reach from 9, one is simple step to right on number line, i.e 10 and another is 15 (given shortcut)

$$\text{Hence, } y=10, z=15 \\ yz = 10 \times 15 = 150$$

16 votes

-- Srinath Jayachandran (3.7k points)

1.5

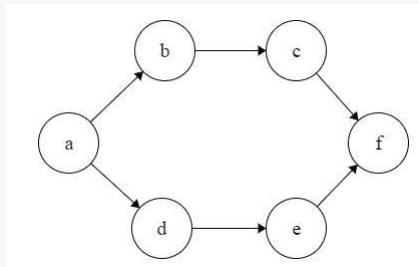
Graph Algorithms(48) [top](#)

<http://gateoverflow.in/39669>

1.5.1 Graph Algorithms: GATE 2016-1-11 [top](#)

<http://gateoverflow.in/39669>

Consider the following directed graph:



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

[gate2016-1](#) | [algorithms](#) | [graph-algorithms](#) | [normal](#) | [numerical-answers](#)

Answer

1.5.2 Graph Algorithms: GATE 2016-2-11 [top](#)

<http://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

Answer

1.5.3 Graph Algorithms: GATE1994-1.22 [top](#)

<http://gateoverflow.in/2465>

Which of the following statements is false?

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

gate1994 | algorithms | normal | graph-algorithms

Answer

1.5.4 Graph Algorithms: GATE1994_24 [top](#)

<http://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 := ∅
while V ≠ ∅ do
begin
    select a vertex u ∈ V such that
    _____;
    V := V - {u};
    if u is such that
    _____ then I := I ∪ {u}
end;
Output(I);
```

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

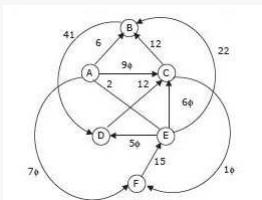
gate1994 | algorithms | graph-algorithms | normal

Answer

1.5.5 Graph Algorithms: GATE1996_17 [top](#)

<http://gateoverflow.in/2769>

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



We are interested in the shortest paths from 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
- Write down sequence of vertices in the shortest path from A to E
- What is the cost of the shortest path from A to E ?

gate1996 | algorithms | graph-algorithms | normal

Answer**1.5.6 Graph Algorithms: GATE1998-1.21, ISRO2008-16** [top](#)<http://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](#)
Answer**1.5.7 Graph Algorithms: GATE2000-1.13** [top](#)<http://gateoverflow.in/638>

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](#)
Answer**1.5.8 Graph Algorithms: GATE2001-2.14** [top](#)<http://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 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](#)
Answer**1.5.9 Graph Algorithms: GATE2002-12** [top](#)<http://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 $n \times n$ 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{_____,_____}; }
        }
    }
}

```

- Copy the complete line containing the blanks in the Initialization step and fill in the blanks.
- Copy the complete line containing the blanks in the Algorithm step and fill in the blanks.
- Fill in the blank: The running time of the Algorithm is $O(\underline{\hspace{2cm}})$.

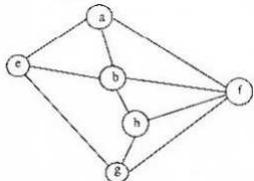
gate2002 algorithms graph-algorithms time-complexity normal descriptive

[Answer](#)

1.5.10 Graph Algorithms: GATE2003-21 [top](#)

<http://gateoverflow.in/911>

Consider the following graph



Among the following sequences

- I. abeghaf
- II. abfehg
- III. abfhge
- IV. afghbe

which are 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

[Answer](#)

1.5.11 Graph Algorithms: GATE2003-67 [top](#)

<http://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

[Answer](#)

1.5.12 Graph Algorithms: GATE2003-70 [top](#)

<http://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 $(v_i, v_{i+1}, \dots, v_j)$ 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.

```
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

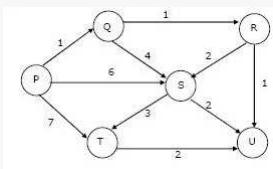
[gate2003](#) [algorithms](#) [graph-algorithms](#) [normal](#)

Answer

1.5.13 Graph Algorithms: GATE2004-44 [top](#)

<http://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](#)

Answer

1.5.14 Graph Algorithms: GATE2004-81 [top](#)

<http://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

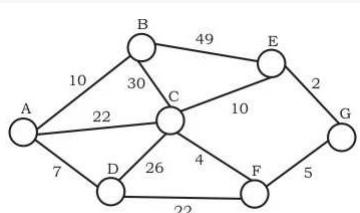
[gate2004](#) [algorithms](#) [graph-algorithms](#) [normal](#)

Answer

1.5.15 Graph Algorithms: GATE2004-IT-56 [top](#)

<http://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](#)

[Answer](#)

1.5.16 Graph Algorithms: GATE2005-38 [top](#)

<http://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. $O(|V|^2)$
- B. $O(|E| + |V|\log|V|)$
- C. $O(|V|\log|V|)$
- D. $O((|E| + |V|)\log|V|)$

[gate2005](#) [algorithms](#) [graph-algorithms](#) [normal](#)

[Answer](#)

1.5.17 Graph Algorithms: GATE2005-82a [top](#)

<http://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](#)

[Answer](#)

1.5.18 Graph Algorithms: GATE2005-82b [top](#)

<http://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](#)

[Answer](#)

1.5.19 Graph Algorithms: GATE2005-IT-14 [top](#)

<http://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](#)

Answer

1.5.20 Graph Algorithms: GATE2005-IT-15 [top](#)

<http://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.

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

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

[gate2005-it](#) [algorithms](#) [graph-algorithms](#) [normal](#)

Answer

1.5.21 Graph Algorithms: GATE2005-IT-84a [top](#)

<http://gateoverflow.in/3855>

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 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](#)
Answer

1.5.22 Graph Algorithms: GATE2005-IT-84b [top](#)

<http://gateoverflow.in/385>

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 E3

- A. $(A[i][j] \&\& !A[j][i])$
- B. $(!A[i][j] \&\& A[j][i])$
- C. $(!A[i][j] \mid\mid A[j][i])$
- D. $(A[i][j] \mid\mid !A[j][i])$

[gate2005-it](#) [algorithms](#) [graph-algorithms](#) [normal](#)
Answer

1.5.23 Graph Algorithms: GATE2006-12 [top](#)

<http://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](#)
Answer

1.5.24 Graph Algorithms: GATE2006-48 [top](#)

<http://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?

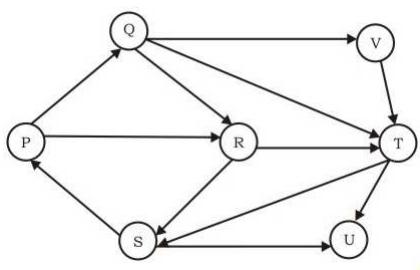
- A. There must exist a vertex w adjacent to both u and v in G
- B. There must exist a vertex w whose removal disconnects u and v in G
- C. There must exist a cycle in G containing u and v
- D. There must exist a cycle in G containing u and all its neighbours in G

[gate2006](#) [algorithms](#) [graph-algorithms](#) [normal](#)
Answer

1.5.25 Graph Algorithms: GATE2006-IT-46 [top](#)

<http://gateoverflow.in/3589>

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



- 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

[Answer](#)

1.5.26 Graph Algorithms: GATE2006-IT-47 [top](#)

<http://gateoverflow.in/359>

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

$$\begin{array}{ll} d(P) = 5 & f(P) = 12 \\ \text{units} & \text{units} \\ d(Q) = 6 & f(Q) = 10 \\ \text{units} & \text{units} \\ d(R) = 14 & f(R) = 18 \\ \text{unit} & \text{units} \end{array}$$

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

[Answer](#)

1.5.27 Graph Algorithms: GATE2007-41 [top](#)

<http://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.

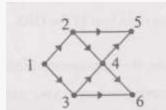
gate2007 algorithms graph-algorithms easy

[Answer](#)

1.5.28 Graph Algorithms: GATE2007-5 [top](#)

<http://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](#)

[Answer](#)

1.5.29 Graph Algorithms: GATE2007-IT-24 [top](#)

<http://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](#)

[Answer](#)

1.5.30 Graph Algorithms: GATE2007-IT-3, UGCNET-June2012-III-34 [top](#)

<http://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 $(u, v) > 12$

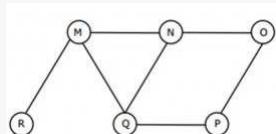
[gate2007-it](#) [algorithms](#) [graph-algorithms](#) [normal](#) [ugcnetjune2012iii](#)

[Answer](#)

1.5.31 Graph Algorithms: GATE2008-19 [top](#)

<http://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



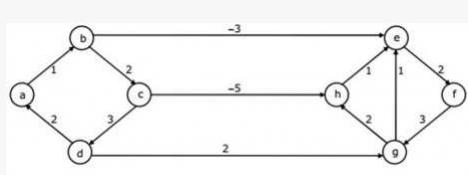
- A. MNOPQR
- B. NQMPOR
- C. QMNPRO
- D. QMNPOR

[gate2008](#) [normal](#) [algorithms](#) [graph-algorithms](#)

[Answer](#)

1.5.32 Graph Algorithms: GATE2008-45 [top](#)

<http://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](#)

[Answer](#)

1.5.33 Graph Algorithms: GATE2008-7 [top](#)

<http://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](#)

[Answer](#)

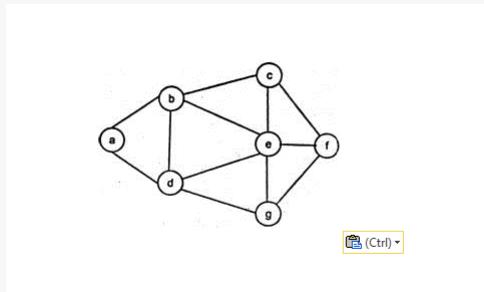
1.5.34 Graph Algorithms: GATE2008-IT-47 [top](#)

<http://gateoverflow.in/335>

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

1. a b e f d g c
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 all 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](#)

[Answer](#)

1.5.35 Graph Algorithms: GATE2009-13 [top](#)

<http://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.
- P only
 - Q only
 - Both P and Q
 - Neither P nor Q

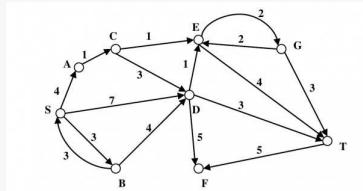
gate2009 | algorithms | graph-algorithms | normal

Answer

1.5.36 Graph Algorithms: GATE2012_40 [top](#)

<http://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.



- SDT
- SBDT
- SACDT
- SACET

gate2012 | algorithms | graph-algorithms | normal

Answer

1.5.37 Graph Algorithms: GATE2013_19 [top](#)

<http://gateoverflow.in/1441>

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

- $\theta(n^2)$
- $\theta(n^2 \log n)$
- $\theta(n^3)$
- $\theta(n^3 \log n)$

gate2013 | algorithms | graph-algorithms | normal

Answer

1.5.38 Graph Algorithms: GATE2014-1-11 [top](#)

<http://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?

- $\Theta(n)$
- $\Theta(n + m)$
- $\Theta(n^2)$
- $\Theta(m^2)$

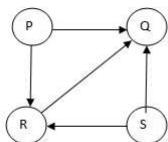
gate2014-1 | algorithms | graph-algorithms | normal

Answer

1.5.39 Graph Algorithms: GATE2014-1-13 [top](#)

<http://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](#)

[Answer](#)

1.5.40 Graph Algorithms: GATE2014-2-14 [top](#)

<http://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.

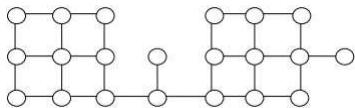
[gate2014-2](#) [algorithms](#) [graph-algorithms](#) [normal](#)

[Answer](#)

1.5.41 Graph Algorithms: GATE2014-3-13 [top](#)

<http://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](#)

[Answer](#)

1.5.42 Graph Algorithms: GATE2015-1_45 [top](#)

<http://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](#)

[Answer](#)

1.5.43 Graph Algorithms: GATE2016-2-41 [top](#)

<http://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

[Answer](#)

1.5.44 Graph Algorithms: GATE2017-1-26 [top](#)

<http://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)

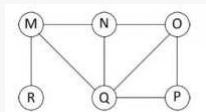
gate2017-1 | algorithms | graph-algorithms | normal

[Answer](#)

1.5.45 Graph Algorithms: GATE2017-2-15 [top](#)

<http://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

[Answer](#)

1.5.46 Graph Algorithms: Gate2000-2.19 [top](#)

<http://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

- 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

Answer

1.5.47 Graph Algorithms: TIFR2013-B-5 [top](#)

<http://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. $O(n \cdot \log(n))$ 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 $O(n^3)$

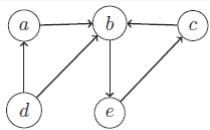
tifr2013 algorithms graph-algorithms

Answer

1.5.48 Graph Algorithms: TIFR2014-B-3 [top](#)

<http://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

Answer

Answers: Graph Algorithms

1.5.1 Graph Algorithms: GATE 2016-1-11 [top](#)

<http://gateoverflow.in/39669>



Selected Answer

Here Start with a and End with f.

a _ _ _ _ f

Blanks spaces are to be filled with b, c, d, e such that b comes before c, and d comes before e..

Number of ways to arrange b,c,d,e such that b comes before c and d comes before e. will be = $4!/(2!*2!) = 6$

28 votes

-- Abhilash Panicker (8.7k points)

1.5.2 Graph Algorithms: GATE 2016-2-11 [top](#)

<http://gateoverflow.in/39563>



Selected Answer

No of nodes at level 0(root) of tree =>1
 No of nodes at level 1 of tree =>2
 No of nodes at level 2 of tree =>4
 No of nodes at level 3 of tree =>8
 No of nodes at level 4 of tree =>16
 Last node in level 4th is the node we are looking for => $1+2+4+8+16 = 31$

13 votes

-- Akash (41.5k points)

1.5.3 Graph Algorithms: GATE1994-1.22 [top](#)

<http://gateoverflow.in/2465>

Answer: B

- (a) True.
- (b) False.
- (c) True.
- (d) True.

6 votes

-- Rajarshi Sarkar (34.4k points)

1.5.4 Graph Algorithms: GATE1994_24 [top](#)

<http://gateoverflow.in/2520>

- (a) While adding vertex u to I it should not have an edge with any node in I.
- (b) The algorithm runs till V is empty (in $O(n)$ time) and is checking u with each vertex v in set I (in $O(n)$ time). So, overall complexity $O(n^2)$.

3 votes

-- Rajarshi Sarkar (34.4k points)

1.5.5 Graph Algorithms: GATE1996_17 [top](#)

<http://gateoverflow.in/2769>

Part a: A B D C F E

Part b: A B D C F E

Part c : 84

4 votes

-- Himanshu Agarwal (15.9k points)

1.5.6 Graph Algorithms: GATE1998-1.21, ISRO2008-16 [top](#)

<http://gateoverflow.in/1658>

Selected Answer

Answer is A because floyd warshall algorithm used to find all shortest path which is a dynamic programming approach.

8 votes

-- shashi shekhar (505 points)

1.5.7 Graph Algorithms: GATE2000-1.13 [top](#)

<http://gateoverflow.in/636>

Selected Answer

X - 3 DFS uses stack implicitly

Y-2 BFS uses queue explicitly in Algo
Z-1 Heap-Heapsort

Answer C

6 votes

-- Akash (41.5k points)

1.5.8 Graph Algorithms: GATE2001-2.14 [top](#)

<http://gateoverflow.in/732>



Ans :->

C.

BFS is used to count shortest path from source (If all path costs are 1 !)

now if u is visited before v it means 2 things.

1. Either u is closer to v.
2. if u & v are same distance from r, then our BFS algo chose to visit u before v.

7 votes

-- Akash (41.5k points)

1.5.9 Graph Algorithms: GATE2002-12 [top](#)

<http://gateoverflow.in/865>



Selected Answer

```
INITIALIZATION: For i = 1 ... n
    {For j = 1 ... n
        { if a[i,j] = 0 then P[i,j] = infinite // i.e. if there is no direct path then put infinite
            else P[i,j] = a[i,j];
        }
    }
ALGORITHM:
For i = 1 ... n
{For j = 1 ... n
    {For k = 1 ... n
        {
            P[i, j] = min( p[i,j] , p[i,k] + p[k,j])
        };
    }
}
```

time complexity $O(n^3)$

this algorithm is 4 weighted graph but it will work 4 unweighted graph 2 because if $p[i,j]=1$, $p[i,k]=1$ and $p[k,j]=1$ then according to the algo $p[i,j] = \min(p[i,j], p[i,k] + p[k,j]) = \min(1,2) = 1$

And all the other case is also satisfied.(like as if $p[i,j]$ was 0 in last iteration nd there exist a path via k)

3 votes

-- Saurav Kumar Gupta (2.1k points)

1.5.10 Graph Algorithms: GATE2003-21 [top](#)

<http://gateoverflow.in/911>



Selected Answer

Answer will be (D)

DFS goes upto how much depth possible and then backtrack and go to the next link.

Here only 'abfehg' not possible because e and h consecutively is not possible by any backtracking of DFS traversal

2 votes

-- srestha (55.2k points)

1.5.11 Graph Algorithms: GATE2003-67 [top](#)



Selected Answer

After applying the shortest path algorithm, check cost of vertex from source to every vertex in G_1 . If G_1 is connected all these costs must be 0 as edge weights of subgraph G_1 is 0 and that should be the shortest path. If cost is not 0, to at least one vertex in G_1 (not necessarily G), then G_1 is disconnected.

Ans is b

17 votes

-- Anurag Semwal (7.6k points)

1.5.12 Graph Algorithms: GATE2003-70 [top](#)



Selected Answer

D is correct.

Consider a graph with 2 nodes and one edge from V_1 to V_2 ,

Running the above algorithm will result in A being

A	1	2
1	1	2
2	1	2

Clearly options B and C are wrong. Since

1. $A[1][1]$ and $A[2][2] > n-1$ and there exists no Hamiltonian cycle. Hence invalid
2. The longest path between V_1 and V_2 is 1, but $A[1][2]$ is 2, which is invalid. And no path between V_2 and V_1 yet $A[2][1] = 1$ // **it should be max cost path between j and k not path length.**

Hence A or D could be valid.

Now consider a graph with 2 nodes and two edges, one from V_1 and V_2 and other form V_2 and V_1 . Running the above algorithm will result in A being

A	1	2
1	2	3
2	3	4

Hence option A is invalid, as $A[i][j]$ can be $>n$

D is correct

12 votes

-- ryan sequeira (2.9k points)

1.5.13 Graph Algorithms: GATE2004-44 [top](#)



Selected Answer

Ans is (B). In Dijkstra's algorithm at each point we choose the smallest weight edge which starts from any one of the vertices in the shortest path found so far and add it to the shortest path.

7 votes

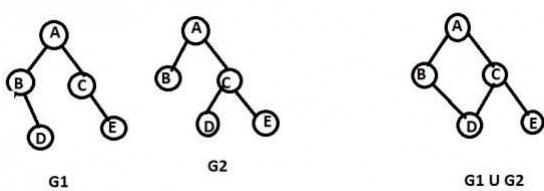
-- gate_asp (655 points)

1.5.14 Graph Algorithms: GATE2004-81 [top](#)



Selected Answer

Take a tree for example



- (A) False. Every vertex of tree(other than leaves) is a cut vertex
 (B) True
 (C) False. Without E in G1 and G2, G1 U G2 has no bridge.
 (D) False. G1 U G2, G1, G2 three graphs have same chromatic number of 2.

12 votes

-- srestha (55.2k points)

1.5.15 Graph Algorithms: GATE2004-IT-56 [top](#)



Answer is (D)

- (A) and (B) produce disconnected components with the GIVEN order in options which is NEVER allowed by prim's algorithm.
 (C) produces connected component every instant a new edge is added BUT when first vertex is chosen(first vertex is chosen randomly) first edge must be the minimum weight edge that is chosen . Therefore (A,D) MUST be chosen BEFORE (A,B). Therefore (C) is FALSE

7 votes

-- Sandeep_Uniyal (7.3k points)

1.5.16 Graph Algorithms: GATE2005-38 [top](#)



- D.- Binary heap. $|E|$ decrease key operations and each taking $O(\log |V|)$ time plus $|V|$ extract-min operations each taking $O(\log |V|)$.
 B- Fibonacci heap. $|E|$ decrease key operations and each taking $O(1)$ time plus $|V|$ extract-min operations each taking $O(\log |V|)$.
 A- Array. Finding min-vertex in each iteration takes $O(V)$ and this needs to be done $|V|$ times.
 Binomial Heap- same as Binary heap here as the critical operations are decrease key and extract-min.

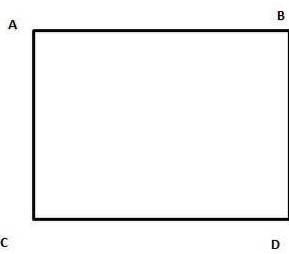
14 votes

-- Gate Keeda (18.8k points)

1.5.17 Graph Algorithms: GATE2005-82a [top](#)



For 82a The answer should be Option A because edge 'e' is the lightest safe edge connecting X and Y so the minimum spanning tree of G must contain 'e' (Greedy and optimal choice).
 While B might seem correct but it is not always true. One such case is when G is not connected therefore there might not be any path between 's' and 't'.
 Since the question is about definitely true B is incorrect and A is the only correct option



Lets say $AC = 1$ $CD = 2$ $BD = 3$ and $AB = 4$

Then if $s = A$ and $t = B$ then AC is the lightest edge crossing X and Y where $X = \{A\}$ and $Y = \{C, B, D\}$
But clearly AC is not on the shortest path from A to B . The shortest path is $AB = 4$.

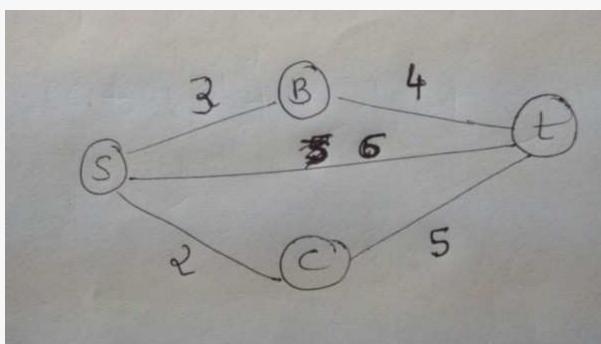
4 votes

-- chandan1223 (167 points)

1.5.18 Graph Algorithms: GATE2005-82b [top](#)

<http://gateoverflow.in/82129>

Here ans should be A.



Here shortest path will give 6 .

Spanning tree contains edges of weights 2,3,4 so congestion in this case is $\max(2,3,4)$ that is 4. for path s to t overall congestion is $\max(3,4) = 4$ but total weight is 7.

option C and D are i think not related to this question.

3 votes

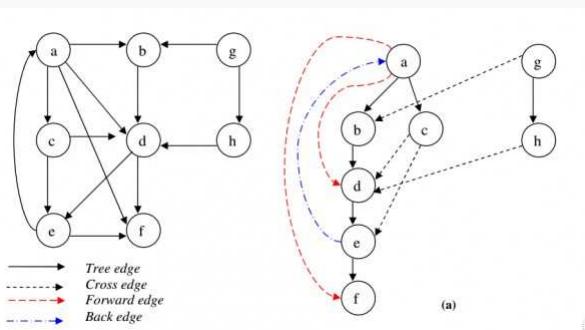
-- papesh (22.8k points)

1.5.19 Graph Algorithms: GATE2005-IT-14 [top](#)

<http://gateoverflow.in/3759>



Selected Answer



Tree edges are those edges which appear in the final DFS forest. For example in case of connected graph (i.e. resulting dfs forest containing only one tree), if we run DFS over it, edges belonging to the resulting DFS tree are called tree edges.

Let us assume the graph has x number of connected (or strongly connected in case of a directed graph) components. And assume 1st component has K_1 tree edges, 2nd component has K_2 tree edges and x th component has K_x tree edges.

Such that $K_1 + K_2 + K_3 + \dots + K_x = K$ ($=$ total)

Or in other way we can imagine like, the final DFS forest has x trees and those trees are having $K_1, K_2, K_3, \dots, K_x$ edges respectively.

Now we know that a tree having K_x edges contains $K_x + 1$ nodes. Similarly a tree having K_1 edges contains $K_1 + 1$ nodes, etc. and so on.

So, Summation of nodes in each tree = n

$$(K_1 + 1) + (K_2 + 1) + (K_3 + 1) + \dots + (K_x + 1) = n \Rightarrow (K_1 + K_2 + K_3 + \dots + K_x) + x = n \Rightarrow$$

13 votes

-- Debashish Deka (49.1k points)

Answer D => $n-k$

Why ?

If Graph is connected , while doing DFS we will visit some spanning Tree of Graph. So no of edges will be $n-1$ &

No of components => $n - (n-1) \Rightarrow 1$

If Graph is not connected in that case when we do the DFS on those disconnected graph,

For every disconnected component with say x vertices, we will get $x-1$ Tree edge. When we sum up all vertices we will get total no of vertices. When we sum up all edges in spanning tree of each component, we will get => Total no of vertices - Total No of connected component (Due to for each connected component we are getting tree of no of vertices in that connected component - 1)

So Total connected component => D) $n-k$

11 votes

-- Akash (41.5k points)

1.5.20 Graph Algorithms: GATE2005-IT-15 [top](#)

<http://gateoverflow.in/3760>



Selected Answer

1. Bellman-Ford algorithm =>: $O(nm)$ Assuming n as edges , m as vertices, for every vertex we relax all edges. $m*n$, $O(mn)$
2. Kruskal's algorithm => Remaining Option ,A : $O(m \log n)$
3. Floyd-Warshall algorithm => Dynamic Programming Algo, $O(N^3)$
4. Topological sorting => Boils Down to DFS, $O(V+E)$ D

Answer A)

7 votes

-- Akash (41.5k points)

1.5.21 Graph Algorithms: GATE2005-IT-84a [top](#)

<http://gateoverflow.in/3856>



Selected Answer

If there is a sink in the graph, the adjacency matrix will contain all 1's(except diagonal) in one column and all 0's (except diagonal) in the corresponding row of that vertex. The given algorithm is a smart way of doing this as it finds the sink in $O(n)$ time complexity.

The first part of the code, is finding if there is any vertex which doesn't have any outgoing edge to any vertex coming after it in adjacency matrix. The smart part of the code is E2, which makes rows skip when there is no edge from i to it, making it impossible them to form a sink. This is done through

E1: $\text{!A}[i][j]$

and
E2: $i = j$;

E1 makes sure that there is no edge from i to j and i is a potential sink till $A[i][j]$ becomes 1. If $A[i][j]$ becomes 1, i can no longer be a sink, similarly all previous j can also not be a sink (as there was no edge from i to them and a sink requires an edge from all other vertices). Now, the next potential candidate for sink is j . So, in E2, we must make $i = j$.

So, answer is (C)

For E3,
http://gateoverflow.in/3857/gate2005-it_84b

9 votes

-- Arjun Suresh (287k points)

1.5.22 Graph Algorithms: GATE2005-IT-84b [top](#)

<http://gateoverflow.in/3857>



Selected Answer

If there is a sink in the graph, the adjacency matrix will contain all 1s (except diagonal) in one column and all 0s (except diagonal) in the corresponding row of that vertex. The given algorithm is a smart way of doing this as it finds the sink in $O(n)$ time complexity.

The first part of the code, is finding if there is any vertex which does not have any outgoing edge to any vertex coming after it in adjacency matrix. The smart part of the code is E2, which makes rows skip when there is no edge from i to i , making it impossible them to form a sink. This is done through

E1: $\neg A[i][j]$
and
E2: $i = j$;

E1 makes sure that there is no edge from i to j and i is a potential sink till $A[i][j]$ becomes 1. If $A[i][j]$ becomes 1, i can no longer be a sink, similarly all previous j can also not be a sink (as there was no edge from i to them and a sink requires an edge from all other vertices). Now, the next potential candidate for sink is j . So, in E2, we must make $i = j$.

Now, the loop breaks when we found a potential sink- that is a vertex which does not have any outgoing edge to any coming after it in adjacency matrix. So, if the column in which this vertex comes is all 1s and the row is all 0s (except diagonal), this is the sink. Otherwise there is no sink in the graph. So, E3 is checking this condition.

But in the code flag is used for storing the state that sink is present or not. And as per the usage of flag in code, by default sink is considered present. So, the condition in E3 must make flag = 0, if the found i is not a sink. So, the condition should be:

$A[i][j] \mid\mid \neg A[j][i]$

So, (D) is the answer.

12 votes

-- Arjun Suresh (287k points)

1.5.23 Graph Algorithms: GATE2006-12 [top](#)

<http://gateoverflow.in/891>



Answer is A) Queue

we can find single source shortest path in unweighted graph by using Breadth first search (BFS) algorithm which using "Queue" data structure , which time $O(m+n)$ (i.e. linear with respect to the number of vertices and edges.)

21 votes

-- Mithlesh Upadhyay (5.3k points)

1.5.24 Graph Algorithms: GATE2006-48 [top](#)

<http://gateoverflow.in/1824>



Consider following graph



Dfs is .

so D is answer.

5 votes

-- Nikunj Vinchhi (61 points)

1.5.25 Graph Algorithms: GATE2006-IT-46 [top](#)

<http://gateoverflow.in/3589>



Here the answer is B.

A graph is said to be **strongly connected** if every vertex is reachable from every other vertex.

The strongly connected component is always maximal that is if x is strongly connected component there should not exist another strongly connected component which contains x.

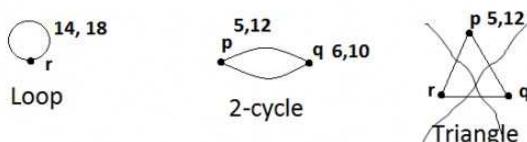
If we take R as a strongly connected component but which is part of PQRS and PQRS is part of PQRSVT.

8 votes

-- papesh (22.8k points)

1.5.26 Graph Algorithms: GATE2006-IT-47 [top](#)

<http://gateoverflow.in/3590>



As seen in question after 10 we have to go for p again and since p is finish and then r is start so r must be disconnected because if there are edges from q to r then r must be visited before of q and p ending.

D is answer

6 votes

-- Prashant Singh (47.6k points)

1.5.27 Graph Algorithms: GATE2007-41 [top](#)

<http://gateoverflow.in/1239>



Dijkstra and warshall 's algo only used for weighted graph.

Both DFS and BFS can be used for finding path between 2 vertices in undirected and unweighted graph but BFS can only give the shortest path as concern in given question so BFS is Ans.

Note :- finding only path(DFS) and finding shortest path(BFS) ..Matters a lot

Must Read:-

<https://www.quora.com/What-are-the-advantages-of-using-BFS-over-DFS-or-using-DFS-over-BFS-What-are-the-applications-and-downsides-of-each>

1 votes

-- Rajesh Pradhan (16.9k points)

1.5.28 Graph Algorithms: GATE2007-5 [top](#)

<http://gateoverflow.in/31821>



Selected Answer

Go with Vertex with indegree 0 remove that vertex with all edges going from it . Follow that procedure.

We See 3 cannot come at first B/c indegree is not 0. **so D is answer here .**

ALL other options are Topologocal order.

Only 1 and 4 order matter for this quetion.

5 votes

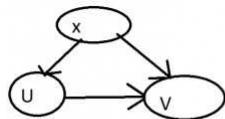
-- Prashant Singh (47.6k points)

1.5.29 Graph Algorithms: GATE2007-IT-24 [top](#)

<http://gateoverflow.in/3457>



Selected Answer



I'm gonna disprove all wrong options here.

A) $d[u] < d[v]$, Counter Example => Well if we directly start DFS on V first. Then I call DFS on X which visits U.

B) $d[u] < f[v]$ Counter example => Same as A)

C) $f[u] < f[v]$ Counter example => Same as A) again 😊

So answer is D)

8 votes

-- Akash (41.5k points)

1.5.30 Graph Algorithms: GATE2007-IT-3, UGCNET-June2012-III-34 [top](#)

<http://gateoverflow.in/3434>



Selected Answer

C. $\text{weight}(u,v) \geq 12$

If weight $(u, v) < 12$, then the min. weight of $(s, v) = \text{weight of } (s, u) + \text{weight of } (u, v) = 53 + (<12)$ will be less than 65.

16 votes

-- Arjun Suresh (287k points)

1.5.31 Graph Algorithms: GATE2008-19 [top](#)

<http://gateoverflow.in/417>

Selected Answer

- A) MNOPQR -> If you try to run BFS, after M, you must traverse NQR (In some order) Here P is traversed before Q, which is wrong.
- B) NQMPOR -> This is also not BFS. P is traversed before O !
- C) QMNPOR -> Correct
- D) QMNPOR -> Incorrect. Because R need to be traversed before O.(Because M is ahead of N in queue).

Answer :- > C

4 votes

-- Akash (41.5k points)

1.5.32 Graph Algorithms: GATE2008-45 [top](#)

<http://gateoverflow.in/457>

Selected Answer

- (d) all the vertices. Just simulate the Dijkstra's algorithm on it. Dijkstra's algorithm is not meant for graphs with negative-edge-weight-cycle, but here it does give the correct shortest path.

14 votes

-- Arjun Suresh (287k points)

1.5.33 Graph Algorithms: GATE2008-7 [top](#)

<http://gateoverflow.in/405>

Selected Answer

Run DFS to find connected components. Its time complexity is $\Theta(m + n)$, hence (C) is the answer.

11 votes

-- Happy Mittal (10.7k points)

1.5.34 Graph Algorithms: GATE2008-IT-47 [top](#)

<http://gateoverflow.in/3357>

Selected Answer

Answer: B

1. After f is visited, c or g should be visited next. So, the traversal is incorrect.
4. After c is visited, e or f should be visited next. So, the traversal is incorrect.
- 2 and 3 are correct.

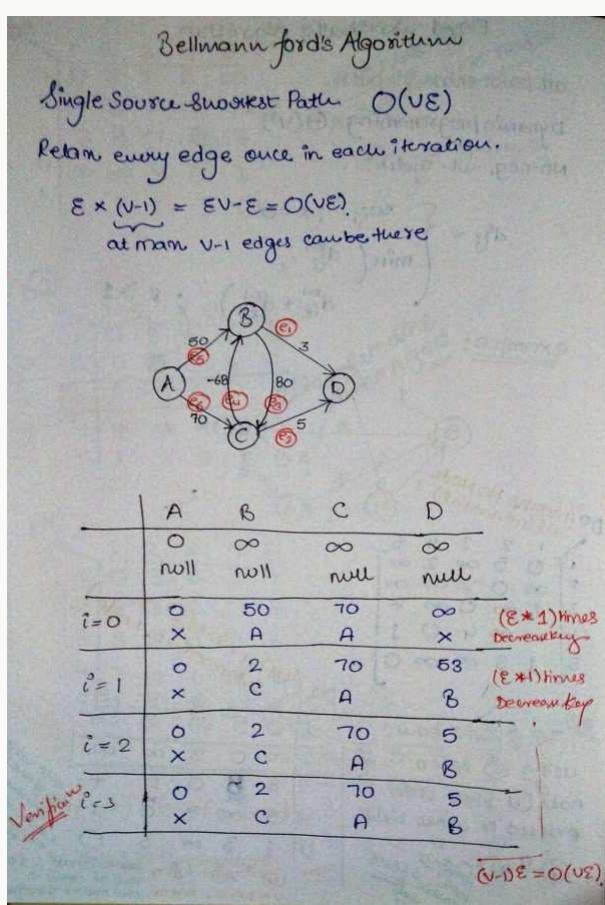
6 votes

-- Rajarshi Sarkar (34.4k points)

1.5.35 Graph Algorithms: GATE2009-13 [top](#)

<http://gateoverflow.in/1305>

Selected Answer



as we can see that last step is the verification step. In that step values remained unchanged. If there was a negative edge weight cycle reachable from source then at verification step also those values will be different from the values above.

In case the cycle is not reachable from source then we can see that they will be at ∞ distance(or cost) from the source from the beginning till the last step. As take anything away from the ∞ it will still be infinite.

But it can also be the case that there are some points which are not forming a cycle and are still unreachable from source those also will be at ∞ distance from the source from the beginning till end.

Hence, we won't be able to make a distinction among the cycle and such vertices. Thus, we say that this algorithm can detect negative edge weight cycles only if they are reachable from the source.

answer = **option B**

11 votes

-- Amar Vashishth (27.9k points)

1.5.36 Graph Algorithms: GATE2012_40 top

<http://gateoverflow.in/1765>



Selected Answer

Relaxation at every vertex is as follows

Note the next vertex is taken out here is in **RED** colour

A	B	C	D	E	F	G	T
S 4	3(by s)	∞	7	∞	∞	∞	∞
B 4(by s)		∞	7	$\because (4+3)$ also=7)(S->d)	∞	∞	∞
A		5(S->B-7)		∞	∞	∞	∞

C	>A)	7	6(S->B->C)	∞	∞	∞
E		7(S->D)		∞	8(S->A->C->E)	10(S->A->C->E)
D			12(S->B->D)	8	10(same so no change)	
E			12		10	
T			12			

Now We see for S to T its (S->A->C->E->.T)

which is Option D

15 votes

-- Kalpish Singhal (2k points)

1.5.37 Graph Algorithms: GATE2013_19 [top](#)

<http://gateoverflow.in/1441>



Selected Answer

Time complexity of Bellman-Ford algorithm is $\Theta(|V||E|)$ where $|V|$ is number of vertices and $|E|$ is number of edges. If the graph is complete, the value of $|E|$ becomes $\Theta(|V|^2)$. So overall time complexity becomes $\Theta(|V|^3)$. And given here is n vertices. So answers ends up to $\Theta(n^3)$.

21 votes

-- Gate Keeda (18.8k points)

1.5.38 Graph Algorithms: GATE2014-1-11 [top](#)

<http://gateoverflow.in/1771>



Selected Answer

Ans (C)

<http://web.eecs.utk.edu/~huangj/CS302S04/notes/graph-searching.html>

12 votes

-- Keith Kr (6.2k points)

1.5.39 Graph Algorithms: GATE2014-1-13 [top](#)

<http://gateoverflow.in/1779>



Selected Answer

The C option has been copied wrongly
 C) Both PSRQ and SPRQ are topological orderings
 i) Apply DFS by choosing P or S as starting vertices
 ii) As the vertex gets a finishing time assign it to the head of a linked list
 iii) The linked list is your required topological ordering

11 votes

-- Akshay Jindal (471 points)

choose vertex in the graph which has 0 indegree . now see graph has such two vertices ie P,S so we can start topological sort either from vertex P or from Q .

lets first start from vertex P now remove this vertex from graph ,we left with three vertices named as Q,S,R from these vertices see which vertex has INDEGREE 0,S vertex HAS indegree 0 therefore sequence IS P,S,R,Q

repeat the above step from wuth vertex S we get sequence as S,P,R,Q

12 votes

-- kunal (20.1k points)

1.5.40 Graph Algorithms: GATE2014-2-14 [top](#)

<http://gateoverflow.in/1969>



Selected Answer

BFS always having starting node. It does not calculate shortest path between every pair but it computes shortest path between W and any other vertex ..

14 votes

-- Digvijay (46k points)

1.5.41 Graph Algorithms: GATE2014-3-13 [top](#)

<http://gateoverflow.in/2047>

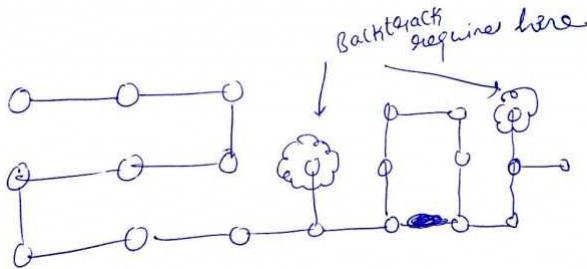


Selected Answer

Total 21 nodes are there out of 2 nodes require back track here in this question.

So max recursion depth is $21 - 2 = 19$

(Do dfs from extreme ends such that max recursion depth will occur i.e. take Left - top node as initial node for DFS as shown in below img.)



Note:- Backtrack means it reduces recursion depth in stack.

9 votes

-- Rajesh Pradhan (16.9k points)

19. apply DFS.

11 votes

-- Gate Keeda (18.8k points)

1.5.42 Graph Algorithms: GATE2015-1_45 [top](#)

<http://gateoverflow.in/8321>



Selected Answer

2 is the answer.

$d(u) - d(v) = 0$ is possible when both u and v have an edge from t and t is in the shortest path from s to u or v .

$d(u) - d(v) = 1$ is possible when v and t are in the shortest path from s to u and both t and v are siblings- same distance from s to both t and v causing $t - u$ edge to be in BFS tree and not $v - u$.

$d(u) - d(v) = -1$ is possible as explained above by interchanging u and v .

$d(u) - d(v) = 2$ is not possible. This is because on BFS traversal we either visit u first or v . Let's take u first. Now, we put all neighbors of u on queue. Since v is a neighbor and v is not visited before as assumed, $d(v)$ will become $d(u) + 1$. Similarly, for v being visited first.

18 votes

-- Arjun Suresh (287k points)

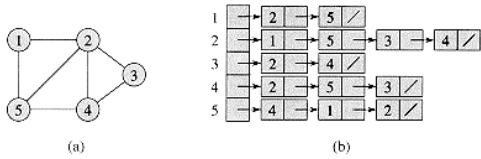
1.5.43 Graph Algorithms: GATE2016-2-41 [top](#)

<http://gateoverflow.in/39620>



Selected Answer

Applying BFS on Undirected graph give you twin pointer .Visit every vertex level-wise for every vertex fill adjacent vertex in the adjacency list. BFS take $O(m+n)$ time.



take extra field for storing no. of linked list for particular vertex. take extra $m+n$ time(m vertex and n edges).

So B is answer

15 votes

-- Prashant Singh (47.6k points)

1.5.44 Graph Algorithms: GATE2017-1-26 [top](#)

<http://gateoverflow.in/118306>

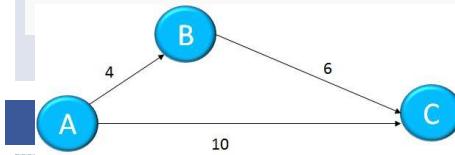


Selected Answer

Ans: A

> MST is not unique only when edges are not distinct. Here the edges are distinct. Be careful for the keyword DISTINCT.

> Shortest Path can be different even if the edges are distinct. Example is shown below. Shortest path from A to C is not unique here.



4 votes

-- Ahwan Mishra (2.6k points)

1.5.45 Graph Algorithms: GATE2017-2-15 [top](#)

<http://gateoverflow.in/118196>



Selected Answer

In BFS starting from a node, we traverse all node adjacent to it at first then repeat same for next nodes

Here you can see only D is following BFS sequence properly

As per BFS if we start from M then RQN has to come after it in any order but in A here O comes so it is not

As per BFS if we start from N then QMO has to come after it in any order but in A here P comes so it is not

As per BFS if we start from M then MNOP has to come after it in any order but in A here R comes so it is not

but D following the sequences

so D is correct answer

more editing coming soon

6 votes

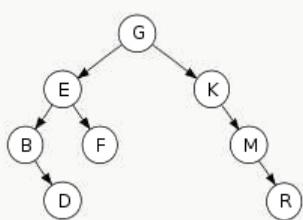
-- Aboveallplayer (18.5k points)

1.5.46 Graph Algorithms: Gate2000-2.19 [top](#)

<http://gateoverflow.in/4208>



Selected Answer



let this be the dfs order of tree then

$u = D$ and $v = F$

so what we conclude

1. its not necessary their is edge b/w them .
2. if thier is no edge then u must be leaf i.e. D is leaf here.
3. it not always possible u and v have same parent . but they have same ancestor ||

14 votes

-- Prashant Singh (47.6k points)

1.5.47 Graph Algorithms: TIFR2013-B-5 [top](#)

<http://gateoverflow.in/25668>

I think arbitrary large weights means having positive weight cycle...so bellmanford algo can be used.. $O(VE)$changing sign of weights of edges....

2 votes

-- papesh pathare (467 points)

1.5.48 Graph Algorithms: TIFR2014-B-3 [top](#)

<http://gateoverflow.in/27137>



Since they said that whenever their is a choice we will have to select the node which is alphabetically earlier , therefore we choose the starting node as A .

The tree then becomes A-B-E-C . Therefore no of tree edges is 3 that is ($T=3$) .

Now there is one cycle B-E-C so we will get a back edge from C to B while performing DFS. Hence $B=1$.

Now D becomes disconnected node and it can only contribute in forming cross edge . There are 2 cross edges D-A , D-B . Therefore $C=2$.

Answer is Option D .

Correct me if am going wrong.

9 votes

-- Riya Roy(Arayana) (7k points)

1.6

Greedy Algorithm(8) [top](#)

1.6.1 Greedy Algorithm: CMI2015-B-04 [top](#)

<http://gateoverflow.in/47063>

You are given n positive integers, $d_1, d_2 \dots d_n$, each greater than 0. Design a greedy algorithm to test whether these integers correspond to the degrees of some n -vertex simple undirected graph $G = (V, E)$. [A simple graph has no self-loops and at most one edge between any pair of vertices].

cmi2015 descriptive algorithms greedy-algorithm

Answer

1.6.2 Greedy Algorithm: GATE1999-2.20 [top](#)

<http://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

[Answer](#)

1.6.3 Greedy Algorithm: GATE2003-69 [top](#)

<http://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

[Answer](#)

1.6.4 Greedy Algorithm: GATE2005-84a [top](#)

<http://gateoverflow.in/1406>

We are given 9 tasks T_1, T_2, \dots, 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

[Answer](#)

1.6.5 Greedy Algorithm: GATE2005-84b [top](#)

<http://gateoverflow.in/82514>

We are given 9 tasks T_1, T_2, \dots, 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](#)
Answer

1.6.6 Greedy Algorithm: GATE2006-IT-48 [top](#)

<http://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](#)
Answer

1.6.7 Greedy Algorithm: GATE2007-76 [top](#)

<http://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](#)
Answer

1.6.8 Greedy Algorithm: GATE2007-77 [top](#)

<http://gateoverflow.in/4351>

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](#)
Answer

Answers: Greedy Algorithm

1.6.1 Greedy Algorithm: CMI2015-B-04 [top](#)

<http://gateoverflow.in/47063>

1. Sort the degrees in non-increasing order.
2. Pick up the highest degree (let us say it a), remove it from the list of degrees and subtract 1 from next a degrees in list.
3. Repeat Step 2 until :
 - If we get all 0 entries in list

- ⇒ Simple Graph exists
- If we get a negative entry or not enough entries to subtract 1 in step 2
⇒ Simple Graph does not exist

Read More : <http://goo.gl/4u3nfh>

Let's take a example : 3,2,1,2

Step 1 : Sort the degree sequence : 3,2,2,1

Step 2: Pick 3, Remove 3 from list and from next 3 elements subtract 1, Result : (1,1,0)

Again repeat step 2 : select 1, Remove 1 and from next 1 subtract 1, Result : (0,0,0)

Thus, a simple graph exists for the following degree-sequence.

1 upvotes

-- Manish Joshi (25k points)

1.6.2 Greedy Algorithm: GATE1999-2.20 [top](#)

<http://gateoverflow.in/468>



Selected Answer

Arrange files in increasing order of records

D A C B E

5 10 15 20 25

75

30

45

15

15(C) 20 (B) 25(E)

5(D) 10(A)

No of movements=15+30+45+75=165

1 upvotes

-- Pooja Palod (31.2k points)

1.6.3 Greedy Algorithm: GATE2003-69 [top](#)

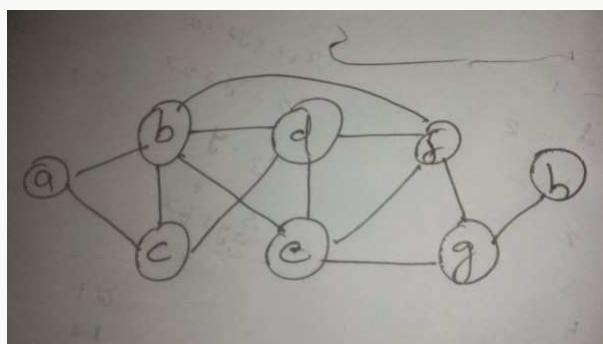
<http://gateoverflow.in/956>



Selected Answer

Solution: B

The problem can be modeled as a graph coloring problem. Construct a graph with one node corresponding to each activity \$A,B,C,D,E,F,G and H. Connect the activities that occur between the start and end time of an activity now the chromatic number of the graph is the number of rooms required.



1 upvotes

-- Gowthaman Arumugam (1.4k points)

1.6.4 Greedy Algorithm: GATE2005-84a [top](#)

<http://gateoverflow.in/1406>



Selected Answer

The most important statement in question is

each task requires one unit of time

This shows that we can greedily choose the better task and that should give us the optimal solution. The best task would be the one with maximum profit. Thus we can sort the tasks based on deadline and then profit as follows:

Task	T7	T2	T9	T4	T5	T3	T6	T8	T1
Deadline	2	2	3	3	4	5	5	7	7

0 ----T7 ----- 1----T2-----2----T9-----3----T5-----4----T3-----5----T8-----6----T1-----7

T4 and T6 left out

D is answer.

9 votes

-- Pooja Palod (31.2k points)

1.6.5 Greedy Algorithm: GATE2005-84b [top](#)



Selected Answer

The most important statement in question is

each task requires one unit of time

This shows that we can greedily choose the better task and that should give us the optimal solution. The best task would be the one with maximum profit. Thus we can sort the tasks based on deadline and then profit as follows:

Task	T7	T2	T9	T4	T5	T3	T6	T8	T1
Deadline	2	2	3	3	4	5	5	7	7

0 --T7 -- 1 -- T2 -- 2 -- T9 -- 3 -- T5 -- 4 -- T3 -- 5 -- T8 -- 6 -- T1 -- 7

so we know that T4 and T6 are left

so profit will not include T4 and T6 = 15 + 20 + 30 + 18 + 16 + 23 + 25 = 147

A is answer

6 votes

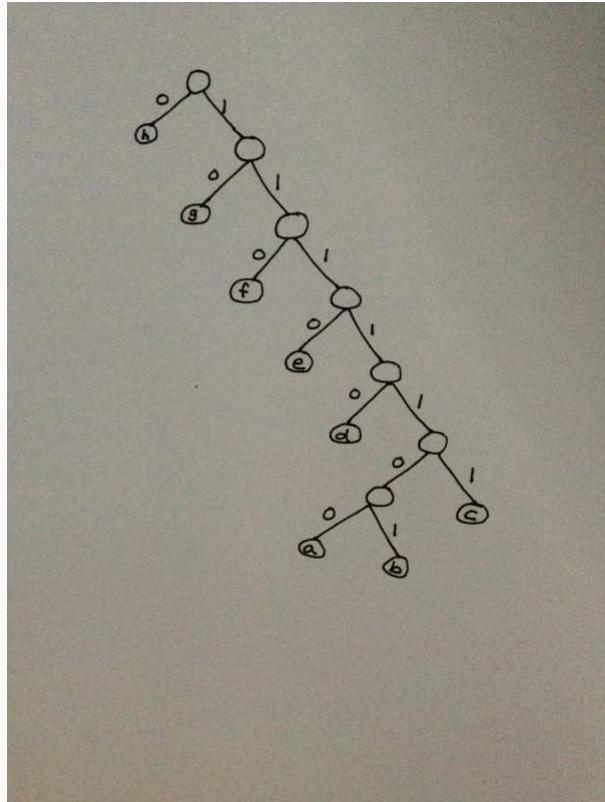
-- Prashant Singh (47.6k points)

1.6.6 Greedy Algorithm: GATE2006-IT-48 [top](#)



Selected Answer

Answer is A. Huffman's tree is as follows. The two least frequent characters are taken as the children of a newly made node and the frequency of the newly made node is made equal to the sum of those two child nodes. Then the same procedure is repeated till all nodes are finished.



$110111100111010 = 110\ 11110\ 0\ 1110\ 10 = fdheg$

9 votes

-- Arjun Suresh (287k points)

1.6.7 Greedy Algorithm: GATE2007-76 [top](#)

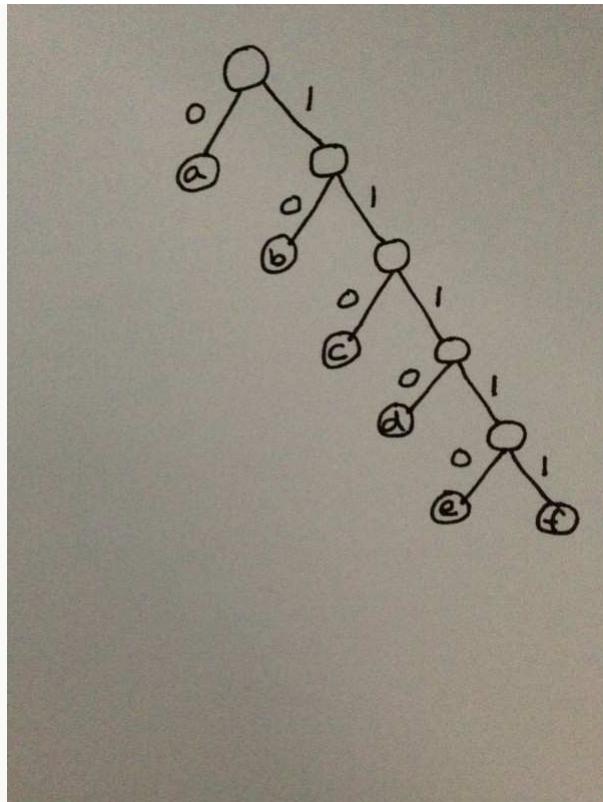


Selected Answer

Based on the probabilities, we can say the probable frequency of the letters will be

16, 8, 4, 2, 1, 1

Now, the Huffman tree can be constructed as follows:



So, A is the answer for 76.

https://www.siggraph.org/education/materials/HyperGraph/video/mpeg/mpegfaq/huffman_tutorial.html

7 votes

-- Arjun Suresh (287k points)

1.6.8 Greedy Algorithm: GATE2007-77 [top](#)

<http://gateoverflow.in/4351>



Selected Answer

Ans should be D)

$$\begin{aligned}
 & \text{Ans} = \text{Dumb} \\
 & b = 1 \\
 & C = 2 \\
 & d = 4 \\
 & e = 8 \\
 & f = 16 \\
 & \text{Total length} = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{32} \\
 & \text{Avg length} = \frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{16} \times 4 + \frac{1}{32} \times 5 + \frac{1}{32} \times 5 \\
 & = \frac{16 + 12 + 8 + 5 + 5}{32} \\
 & = 1.9375
 \end{aligned}$$

5 votes

-- sonam vyas (12.8k points)

1.7

Identify Function(35) [top](#)

<http://gateoverflow.in/4659>

1.7.1 Identify Function: CMI2013-A-09 [top](#)

The below question is based on the following program.

```

procedure mystery (A : array [1..100] of int)
    int i,j,position,tmp;
    begin

```

```

for j := 1 to 100 do
    position := j;
    for i := j to 100 do
        if (A[i] > A[position]) then
            position := i;
        endfor
        tmp := A[j];
        A[j] := A[position];
        A[position] := tmp;
    endfor
end

```

When the procedure terminates, the array A has been:

- A. Reversed
- B. Sorted in descending order
- C. Left unaltered
- D. Sorted in ascending order

[cmi2013](#) [algorithms](#) [identify-function](#)

[Answer](#)

1.7.2 Identify Function: GATE1991-03-viii [top](#)

<http://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](#)

[Answer](#)

1.7.3 Identify Function: GATE1993_7.4 [top](#)

<http://gateoverflow.in/229>

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](#)

[Answer](#)

1.7.4 Identify Function: GATE1994_6 [top](#)

<http://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](#)

[Answer](#)

1.7.5 Identify Function: GATE1995_1.4 [top](#)

<http://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](#)

[Answer](#)

1.7.6 Identify Function: GATE1995_2.3 [top](#)

<http://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](#)

[Answer](#)

1.7.7 Identify Function: GATE1995_4 [top](#)

<http://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
        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

Answer

1.7.8 Identify Function: GATE1998_2.12 [top](#)

<http://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

Answer

1.7.9 Identify Function: GATE1999_2.24 [top](#)

<http://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

Answer

1.7.10 Identify Function: GATE2000-2.15 [top](#)

<http://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 ins between positions i and j (both inclusive). What does the following sequence do, where $1 \leq k \leq n$:

```
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

Answer

1.7.11 Identify Function: GATE2003-1 [top](#)

<http://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

Answer

1.7.12 Identify Function: GATE2003-88 [top](#)

<http://gateoverflow.in/871>

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

[gate2003](#) [algorithms](#) [identify-function](#) [normal](#)
Answer

1.7.13 Identify Function: GATE2004-41 [top](#)

<http://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. $x \bmod y$ 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](#)
Answer

1.7.14 Identify Function: GATE2004-42 [top](#)

<http://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;
}
print(x);
```

- A. $\log m$
- B. $m^{\frac{1}{2}}$
- C. $m^{\frac{1}{3}}$
- D. $m^{\frac{1}{4}}$

[gate2004](#) [algorithms](#) [identify-function](#) [normal](#)
Answer

1.7.15 Identify Function: GATE2005-31 [top](#)

<http://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](#)

[Answer](#)

1.7.16 Identify Function: GATE2005-IT-57 [top](#)

<http://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](#)

[Answer](#)

1.7.17 Identify Function: GATE2006-50 [top](#)

<http://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](#)

[Answer](#)

1.7.18 Identify Function: GATE2006-53 [top](#)

<http://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](#)

[Answer](#)

1.7.19 Identify Function: GATE2006-IT-52 [top](#)

<http://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](#)

[Answer](#)

1.7.20 Identify Function: GATE2008-IT-82 [top](#)

<http://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](#)

[Answer](#)

1.7.21 Identify Function: GATE2008-IT-83 [top](#)

<http://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);
    }
}
```

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

P1	P2
<pre>void f (int n) { if (n/2) { f(n/2); } printf ("%d", n%2); }</pre>	<pre>void f (int n) { if (n <=1) { printf ("%d", n); } else { printf ("%d", n%2); f (n/2); } }</pre>

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

gate2008-it algorithms recursion identify-function normal

Answer

1.7.22 Identify Function: GATE2009-18 [top](#)

<http://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

Answer

1.7.23 Identify Function: GATE2010-35 [top](#)

<http://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](#)

[Answer](#)

1.7.24 Identify Function: GATE2011-48 [top](#)

<http://gateoverflow.in/2154>

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 `foo(345, 10)`?

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

[gate2011](#) [algorithms](#) [recursion](#) [identify-function](#) [normal](#)

[Answer](#)

1.7.25 Identify Function: GATE2011-49 [top](#)

<http://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 `foo(513, 2)`?

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

[gate2011](#) [algorithms](#) [recursion](#) [identify-function](#) [normal](#) [numerical-answers](#)

[Answer](#)

1.7.26 Identify Function: GATE2013_31 [top](#)

<http://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

[Answer](#)

1.7.27 Identify Function: GATE2014-1-41 [top](#)

<http://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**.

gate2014-1 algorithms identify-function normal

[Answer](#)

1.7.28 Identify Function: GATE2014-2-10 [top](#)

<http://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

[Answer](#)

1.7.29 Identify Function: GATE2014-3-10 [top](#)

<http://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

Answer

1.7.30 Identify Function: GATE2015-1-31 [top](#)

<http://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

Answer

1.7.31 Identify Function: GATE2015-2_11 [top](#)

<http://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;
}

```

The return value of fun(5) is _____.

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

Answer

1.7.32 Identify Function: GATE2015-3_49 [top](#)

<http://gateoverflow.in/8558>

Suppose $c = \langle c[0], \dots, c[k-1] \rangle$ 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←1
for

```

```

 $i \leftarrow 0 \text{ to } k - 1$ 
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](#)

Answer

1.7.33 Identify Function: TIFR2010-B-24 [top](#)

<http://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, X, 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](#)

Answer

1.7.34 Identify Function: TIFR2014-B-2 [top](#)

<http://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](#)

Answer

1.7.35 Identify Function: TIFR2017-A-12 [top](#)

<http://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

[Answer](#)

Answers: Identify Function

1.7.1 Identify Function: CMI2013-A-09 [top](#)

<http://gateoverflow.in/46599>



Selected Answer

Ans- B .Sorted in descending order (selection sorting algorithm is used)

3 votes

-- Dhananjay Kumar Sharma (24.9k points)

1.7.2 Identify Function: GATE1991-03-viii [top](#)

<http://gateoverflow.in/523>



Selected Answer

For N=9, it returns 3.
For N=10 it returns 4.
For N=16 it returns 4.
For N=17 it returns 5.
So answer should be C.

5 votes

-- Taymiyyah Bhat (3.8k points)

1.7.3 Identify Function: GATE1993_7.4 [top](#)

<http://gateoverflow.in/229>



Selected Answer

Answer is simply A i.e. it swaps the values of the two.. Take any two values for A and B. and perform the given operations over them.

9 votes

-- Gate Keeda (18.8k points)

1.7.4 Identify Function: GATE1994_6 [top](#)

<http://gateoverflow.in/250>



Selected Answer

answer - x^n

5 votes

-- ankitrokdeonsns (9.1k points)

1.7.5 Identify Function: GATE1995_1.4 [top](#)

<http://gateoverflow.in/259>



Selected Answer

Ans of X remains unchanged. As the if condition becomes false.

X := -10

ans is C . This is classic example of if-else issue. Always else matches for nesting to closest if in C Programming & Pascal .

https://en.wikipedia.org/wiki/Dangling_else

```
if (x>y)
{
    if (x<0)
        x=abs (x)
    else
        x=2*x
}
```

7 votes

-- Akash (41.5k points)

1.7.6 Identify Function: GATE1995_2.3 [top](#)

<http://gateoverflow.in/2615>



Selected Answer

Answer: C

Let X = 3 and Y = 7.

1st pass: X=3, Y=4

2nd pass: X=3, Y=1

3rd pass: X=2, Y=1

4th pass: X=1, Y=1

write (X), which writes 1.

Ref: <http://www.naturalnumbers.org/EuclidSubtract.html>

8 votes

-- Rajarshi Sarkar (34.4k points)

1.7.7 Identify Function: GATE1995_4 [top](#)

<http://gateoverflow.in/2640>



Selected Answer

A. = 3

For B.

begin

for I:=2 to N do

for J:=1 to (I-1) do

begin

TMP:= A[I, J];

```
A[I, J]:= A[J, I];
```

```
A[J, I]:= TMP
```

```
end
```

Should be the condition...

2 votes

-- papesh (22.8k points)

1.7.8 Identify Function: GATE1998_2.12 [top](#)

<http://gateoverflow.in/1684>



Selected Answer

$fun(95) = fun(fun(106)) = fun(96) = fun(fun(107)) = fun(97) = fun(fun(108)) = fun(98) = fun(fun(109)) =$

12 votes

-- Digvijay (46k points)

1.7.9 Identify Function: GATE1999_2.24 [top](#)

<http://gateoverflow.in/1501>



Selected Answer

a	b	c	Return
---	---	---	--------

1	1	1	The final return statement is c < b, so this never returns. Answer D.
---	---	---	---

16 votes

-- Arjun Suresh (287k points)

1.7.10 Identify Function: GATE2000-2.15 [top](#)

<http://gateoverflow.in/662>



Selected Answer

Answer is (a)

Effect of the above 3 reversal for any K is equivalent to left rotation of the array of size n by k.

Let , S[1.....7]

1	2	3	4	5	6	7
---	---	---	---	---	---	---

so, n=7 ,k = 2

reverse(S,1,2) we get [2,1,3,4,5,6,7]

reverse(S,3,7) we get [2,1,7,6,5,4,3]

reverse(S,1,7) we get [3,4,5,6,7,1,2]

hence option (a) Rotates s left by k position is correct

17 votes

-- Kalpana Bhargav (3.2k points)

1.7.11 Identify Function: GATE2003-1 [top](#)

<http://gateoverflow.in/892>



Selected Answer

$$i = 1 \text{ then } p = x \text{ & } s = 1 + x$$

$$i = 2 \text{ then } p = \frac{x^2}{2} \text{ & } s = 1 + x + \frac{x^2}{2}$$

As y tends to infinity, s tends to e^x .

Hence, the correct answer is option B.

16 votes

-- Pooja Palod (31.2k points)

1.7.12 Identify Function: GATE2003-88 [top](#)

<http://gateoverflow.in/971>



Selected Answer

The nested loop is taking all integers from 2 to $2 * \log_2 n$ and take all their non-multiples before n , and make the corresponding entry in A as 1. For example, for 2, and $n = 10$, A[3], A[5], A[7], and A[9] are made 1. Similarly for 3, 4, ... till $2 * \log_2 n$. So, if any entry A[p] is 1 means it must be a multiple of 2, 3, ..., $2\log_2 n$, which is $(2 \log n)!$ and is greater than n . So, for no index p, A[p] will be 0. So, answer is D.

Suppose the line

```
A[j] = A[j] || (j%k);
```

is replaced with

```
A[j] = A[j] || ! (j%k);
```

Now, the nested loop is taking all integers from 2 to $\log_2 n$, take all their multiples before n , and make the corresponding entry in A as 1. For example, for 2, and $n = 10$, A[4], A[6], A[8] and A[10] are made 1. Similarly for 3, 4, ... till $2 * \log_2 n$. So, for all non-prime indices of A, we will have a 1, and for prime indices we have a 0. And we print i if A[j] is 0 meaning j is prime.

7 votes

-- Arjun Suresh (287k points)

1.7.13 Identify Function: GATE2004-41 [top](#)

<http://gateoverflow.in/1038>



Selected Answer

It is a simple algorithm of gcd

here while loop executes until m==n

take any two number as m,n and compute it , get the answer

Ans will be (C)

3 votes

-- srestha (55.2k points)

1.7.14 Identify Function: GATE2004-42 [top](#)

<http://gateoverflow.in/1039>



Selected Answer

by putting $y = m/x$ into $x = (x + y)/2$

$$x = (x + m/x)/2$$

$$\Rightarrow 2x^2 = x^2 + m$$

$$\Rightarrow x = m^{1/2}$$

or we can check by putting 2-3 different values also.

14 votes

-- gate_asp (655 points)

1.7.15 Identify Function: GATE2005-31 [top](#)

<http://gateoverflow.in/1367>



Selected Answer

Option d.

foo is printing the lowest digit. But the printf inside it is after the recursive call. This forces the output to be in reverse order

2, 0, 4, 8

The final value "sum" printed will be 0 as C uses pass by value and hence the modified value inside foo won't be visible inside main.

10 votes

-- anshu (3k points)

1.7.16 Identify Function: GATE2005-IT-57 [top](#)

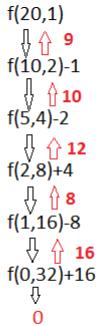
<http://gateoverflow.in/3818>



Selected Answer

See the following calling sequence

Hence Ans is Option C



Red Color Shows return values

8 votes

-- Rajesh Pradhan (16.9k points)

1.7.17 Identify Function: GATE2006-50 [top](#)

<http://gateoverflow.in/1828>



Selected Answer

Option (d)

In the given algorithm the for loop contains a logical expression

$$z[i] = (x[i] \wedge \sim y[i]) \vee (\sim x[i] \wedge y[i]);$$

X, Y[i] = Y then

$$z = (X \wedge Y') \vee (X' \wedge Y)$$

$$z = (X - Y) \vee (Y - X) [A \wedge B' = A - B]$$

6 votes

-- Prasanna Ranganathan (3.9k points)

1.7.18 Identify Function: GATE2006-53 [top](#)

<http://gateoverflow.in/1831>



Selected Answer

The while loop add elements from a and b (whichever is smaller) to c and terminates when either of them exhausts. So, when loop terminates either $i = n$ or $j = m$.

Suppose $i = n$. This would mean all elements from array a are added to c $\Rightarrow k$ must be incremented by n . c would also contain j elements from array b. So, number of elements in c would be $n+j$ and hence $k = n + j$.

Similarly, when $j = m$, $k = m + i$.

Hence, option (D) is correct. (Had k started from -1 and not 0 and we used $++k$ inside loop, answer would have been option (C))

23 votes

-- Arjun Suresh (287k points)

1.7.19 Identify Function: GATE2006-IT-52 [top](#)

<http://gateoverflow.in/359>

Selected Answer

Answer: C

Because $\binom{m}{0} = 1$ and $\binom{n}{n} = 1$.

14 votes

-- Rajarshi Sarkar (34.4k points)

1.7.20 Identify Function: GATE2008-IT-82 [top](#)

<http://gateoverflow.in/340>

Selected Answer

Answer: D

The function prints the binary equivalent of the number n.

Binary equivalent of 173 is 10101101.

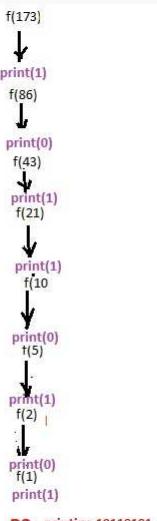
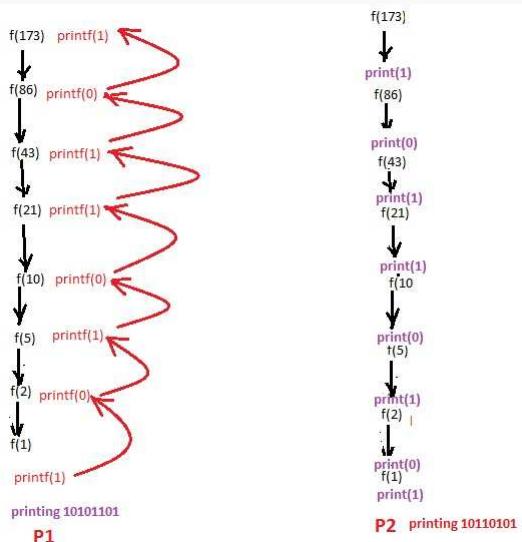
10 votes

-- Rajarshi Sarkar (34.4k points)

1.7.21 Identify Function: GATE2008-IT-83 [top](#)

<http://gateoverflow.in/340>

Selected Answer



here P1 and P2 will print opposite in direction as shown in diagram

and given code fragment will print like P1 and not like P2

Hence ans will be (C)

10 votes

-- srestha (55.2k points)

1.7.22 Identify Function: GATE2009-18 [top](#)

<http://gateoverflow.in/1310>

Selected Answer

The answer is B.

Let the address of x be 1000.

$$1.f(5,1000) = 8$$

$$2.f(4,1000) = 5$$

$$3.f(3,1000) = 3$$

$$4.f(2,1000) = 2$$

$$5.f(1,1000) = 1.$$

The evaluation is done from 5 to 1. Since recursion is used.

6 votes

-- Gate Keeda (18.8k points)

1.7.23 Identify Function: GATE2010-35 [top](#)

<http://gateoverflow.in/2338>

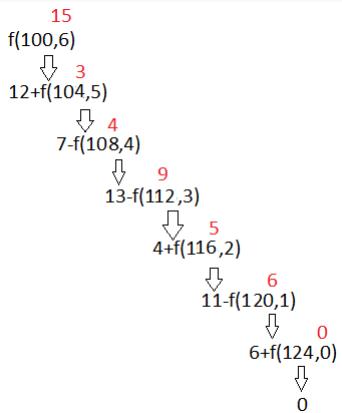


Selected Answer

Suppose int array takes 4 byte for each element and stored at base address 100.

Follow below img and Red color shows the return value.

So 15 is Ans.



6 votes

-- Rajesh Pradhan (16.9k points)

It will print

$$\begin{aligned} & 12 + (7 - (13 - (4 + (11 - (6 + 0))))) \\ &= 12 + (7 - (13 - (4 + (11 - 6)))) \\ &= 12 + 7 - 13 + 9 \\ &= 15 \end{aligned}$$

14 votes

-- gatecse (13.1k points)

1.7.24 Identify Function: GATE2011-48 [top](#)

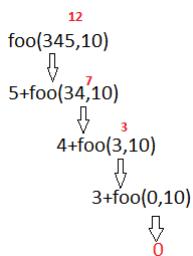
<http://gateoverflow.in/2154>



Selected Answer

Red color represent return values

Ans is 12



9 votes

-- Rajesh Pradhan (16.9k points)

1.7.25 Identify Function: GATE2011-49 [top](#)

<http://gateoverflow.in/43324>

The function returns the sum of digits in a binary representation of the given number

so $1+0+0+0+0+0+0+0+0+1 = 2$

6 votes

-- Sandeep_Uniyal (7.3k points)

1.7.26 Identify Function: GATE2013_31 [top](#)

<http://gateoverflow.in/1542>

The outer loop is running for $n/2$ times and inner loop is running for $\log_2 n$ times (each iteration doubles j and j stops at n means $\log_2 n$ times j loop will iterate).

Now in each iteration k is incremented by $n/2$. So, overall k will be added $n/2 * \log n * n/2$ with an initial value of 0. So, final value of k will be $\Theta(n^2 \log n)$

26 votes

-- Arjun Suresh (287k points)

1.7.27 Identify Function: GATE2014-1-41 [top](#)

<http://gateoverflow.in/1919>

answer is (A) maximum possible sum of elements in any sub-array of array E.

```

int MyX ( int * E, unsinged int size )
{
    int Y= 0;
    int z;
    int i, j,k;
    // calculate sum of the elements of the array E and stores it in Y
    for i 0;i<size;i++)
        Y = Y+E[i];
    //calculate the sum of all possible subaarays (starting from postion 0..n-1)
    for (i=0;i<size;i++)
        for(j=i;j<size ;j++)
        {
            z = 0;
            for(k=i; k<=j;k++)
                z=z+E[k];
            // checks whether sum of elements of each subarray is greater than the current max, if so, then assign it to currentmax
            if(z>Y)
                Y = z;
        }
    // ultimately returns the maximum possible sum of elements in any sub array of given array E
    return Y;
}
  
```

13 votes

-- Kalpana Bhargav (3.2k points)

1.7.28 Identify Function: GATE2014-2-10 [top](#)<http://gateoverflow.in/1964>

Selected Answer

Ans - 9

435-(110110011)

num >= 1; implies a num is shifted one bit right in every while loop execution. While loop is executed 9 times successfully and 10th time num is zero.

So count is incremented 9 times.

Note:

Shifting a number "1" bit position to the right will have the effect of dividing by 2:

8 >> 1 = 4 // In binary: (00001000) >> 1 = (00000100)

11 votes

-- Prasanna Ranganathan (3.9k points)

1.7.29 Identify Function: GATE2014-3-10 [top](#)<http://gateoverflow.in/2044>

Selected Answer

A.

In the computation of given pseudo code for each row and column of Matrix A, each upper triangular element will be interchanged by its mirror image in the lower triangular and after that the same lower triangular element will be again re-interchanged by its mirror image in the upper triangular, resulting the final computed Matrix A same as input Matrix A.

14 votes

-- Gate Keeda (18.8k points)

1.7.30 Identify Function: GATE2015-1-31 [top](#)<http://gateoverflow.in/6263>

Selected Answer

i loop is executing n times. j loop is executing log n times for each i, and so value of p is log n. k loop is executing log p times, which is log log n times for each iteration of i. In each of these q is incremented. So, over all iterations of i, q will be incremented n log log n times. So, D choice.

24 votes

-- Arjun Suresh (287k points)

1.7.31 Identify Function: GATE2015-2_11 [top](#)<http://gateoverflow.in/8060>

Selected Answer

```
fun(1) = 1;
fun(2) = 1 + fun(1) * fun(1) = 1 + 1 = 2;
fun(3) = 1 + fun(1) * fun(2) + fun(2) * fun(1) = 5;
fun(4) = 1 + fun(1) * fun(3) + fun(2) * fun(2) + fun(3) * fun(1) = 1 + 5 + 4 + 5 = 15;
fun(5) = 1 + fun(1) * fun(4) + fun(2) * fun(3) + fun(3) * fun(2) + fun(4) * fun(1) = 1 + 15 + 10 + 10 + 15 = 51;
```

More formal way:

The recurrence relation is

$$f(n) = \begin{cases} 1, & n=1 \\ 1 + \sum_{i=1}^{n-1} f(i) \times f(n-i), & n>1 \end{cases}$$

$$f(1) = 1$$

$$f(2) = 1 + f(1). f(1) = 1 + 1.1 = 2$$

$$f(3) = 1 + f(1). f(2) + f(2). f(1) = 1 + 1.2 + 2.1 = 5$$

$$f(4) = 1 + f(1). f(3) + f(2). f(2) + f(3). f(2) = 1 + 1.5 + 2.2 + 5.1 = 15$$

$$f(5) = 1 + f(1). f(4) + f(2). f(3) + f(3). f(2) + f(4). f(1) = 1 + 1.15 + 2.5 + 5.2 + 15.1 = 51$$

26 votes

-- Arjun Suresh (287k points)

```
fun(1) =1

fun(2) : x=1+fun(1)*fun(1)
         x=1+1*1
         x=2
fun(2) = 2

fun(3) : x = x+fun(1)*fun(2)
         x= 1+1*2
         x= 3

         x= 3+fun(2)*fun(1)
         x= 3+2*1
         x= 5
fun(3) = 5

fun(4) : x = x+fun(1)*fun(3)
         x= 1+1*5
         x= 6

         x = x+fun(2)*fun(2)
         x= 6+2*2
         x= 10

         x = x+fun(3)*fun(1)
         x= 10+5*1
         x= 15

fun(5): x = x+fun(1)*fun(4)
        x= 1+1*15
        x= 16

        x = x+fun(2)*fun(3)
        x= 16+2*5
        x= 26

        x = x+fun(3)*fun(2)
        x= 26+5*2
        x= 36

        x = x+fun(4)*fun(1)
        x= 36+15*1
        x= 51
```

10 votes

-- Prasanna Ranganathan (3.9k points)

1.7.32 Identify Function: GATE2015-3_49 [top](#)

<http://gateoverflow.in/8558>



Selected Answer

Initially $k = 4$, $c = [1, 0, 1, 1]$, $a = 2$, $n = 8$

now let's iterate through the function step by step :

$z = 1$ (at the start of do-something)

$i = 0$ (start of external for loop)

in the do loop

$z = 1 * 1 \% 8 = 1$ (non zero value so considered as true and continue)

$c[0] = 1$ so in the if clause $z = 1 * 2 \% 8 = 2$

in the do loop

$z = 2 * 2 \% 8 = 4$ (since now $z = 2$) (non zero value so considered as true and continue)

$c[0] = 1$ so in the if clause $z = 4 * 2 \% 8 = 0$

now no need to check further :

reason all the operations that update Z are multiplicative operations and hence the value of Z will never change from 0.

14 votes

-- Tamojit Chatterjee (2.2k points)

1.7.33 Identify Function: TIFR2010-B-24 [top](#)

<http://gateoverflow.in/16742>

It prints with $\text{gcd}(x,y)$ and $\text{lcm}(x,y)$

consider $x,y,u,v=17,3,3,17$

$X=14, v=20$

$X=11, v=23$

$X=8, v=26$

$X=5, v=29$

$X=2, v=32$

$Y=1, u=35$

$X=1, v=67$

This is the value obtained

lastly print $(x+y)/2$ and $(v+u)/2$ gives 1 and 51

3 votes

-- zambus (253 points)

1.7.34 Identify Function: TIFR2014-B-2 [top](#)

<http://gateoverflow.in/27138>



Selected Answer

option c. It return no of 1's in binary representation of n.
here $n \& (n-1)$ reset rightmost bit of n in each iteration.

e.g

Suppose $n=15=00001111$ (binary)

$n-1=14(00001110)$

$$\begin{array}{r} 00001111 \\ \wedge \quad 00001110 \\ \hline 00001110 \end{array}$$

4 votes

-- Avdhesh Singh Rana (2.2k points)

1.7.35 Identify Function: TIFR2017-A-12 [top](#)<http://gateoverflow.in/95299>

Selected Answer

```

1. for i=1 to n:
2.     for j=1 to n:
3.         temp=A[i][j]+10
4.         A[i][j]=A[j][i]
5.         A[j][i]=temp-10
6.     end for
7.end for

```

The 3,4,5 lines swap $A[j][i]$ and $A[i][j]$.

The same variables are swapped twice. For eg when: $i = 5, j = 10$. $A[10][5]$ and $A[5][10]$ will be swapped. They will be swapped again when $i = 10, j = 5$.

Two times swap of same elements will lead to A remaining unchanged.

Hence, E is correct.

3 votes

-- tarun_svbk (1k points)

1.8**Minimum Maximum(4)** [top](#)<http://gateoverflow.in/1919>**1.8.1 Minimum Maximum: GATE2014-1-39** [top](#)

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

[gate2014-1](#) [algorithms](#) [numerical-answers](#) [normal](#) [minimum-maximum](#)

Answer

1.8.2 Minimum Maximum: TIFR2014-B-10 [top](#)<http://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 $O(\log^{100} n)$ comparisons.
- B. $O(\log^{100} n)$ 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](#)

Answer

1.8.3 Minimum Maximum: TIFR2014-B-6 [top](#)<http://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 $\{1, \dots, n\}$ 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

Answer

1.8.4 Minimum Maximum: TIFR2014-B-9 [top](#)<http://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 $O(\log^2 n)$ comparisons.
- B. $O(\log^2 n)$ 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

Answer

Answers: Minimum Maximum**1.8.1 Minimum Maximum: GATE2014-1-39** [top](#)<http://gateoverflow.in/1917>

Selected Answer

Ans: minimum number of comparison require to find minimum and maximum is: Approach is divide and conquer

```
T(n) = T(floor(n/2)) + T(ceil(n/2)) + 2
T(2) = 1 // if two element then compare both and return max and min
T(1) = 0 // if one element then return both max and min same
```

If n is a power of 2, then we can write $T(n)$ as:

$$T(n) = 2T(n/2) + 2$$

After solving above recursion, we get

$$T(n) = 3/2n - 2$$

Thus, the approach does $3/2n - 2$ comparisons if n is a power of 2. And it **does more than $3/2n - 2$ comparisons if n is not a power of 2**.

So, here in this case put $n=100$ and we will get $(3/2)(100) - 2 = 148$ comparison

23 votes

-- Jay (1.2k points)

1.8.2 Minimum Maximum: TIFR2014-B-10 [top](#)<http://gateoverflow.in/27195>

Selected Answer

I think ans will be C)

to be accurate it will be need $3n/2 - 2$ comparisons .

4 votes

-- Pranay Datta (9.6k points)

1.8.3 Minimum Maximum: TIFR2014-B-6 [top](#)<http://gateoverflow.in/27183>

Selected Answer

Let us consider 3 numbers {1,2,3}

We will consider the permutation along with min no of times MIN is updated .

Permutation : No of times MIN updated (Minimum)

1,2,3	1
1,3,2	1
2,1,3	2
2,3,1	2
3,1,2	2
3,2,1	3

Total number of times MIN updated is : 11 .

Average no of times MIN updated is : (11/6)

Now going by the options i am getting B .

$$H_3 = 1 + 1/2 + 1/3 = 11/6 .$$

H₃ is the answer and that is option B .

10 votes

-- Riya Roy(Arayana) (7k points)

1.8.4 Minimum Maximum: TIFR2014-B-9 [top](#)

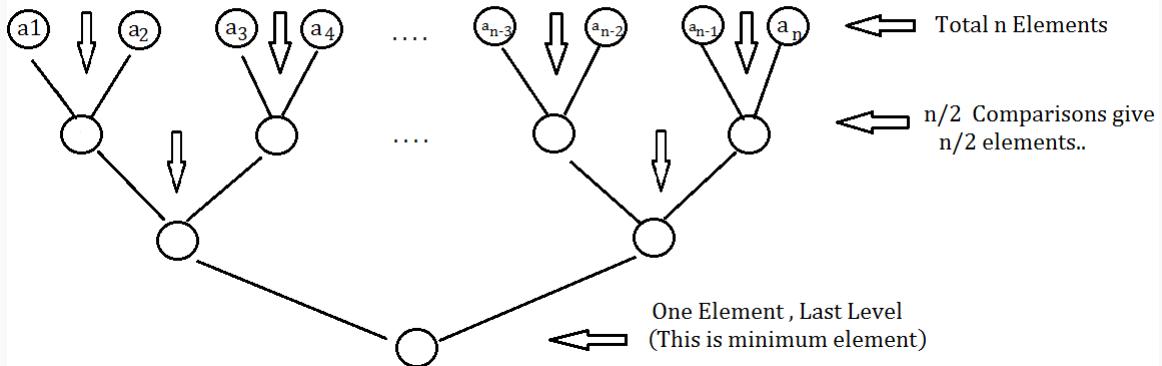
<http://gateoverflow.in/27194>



Selected Answer

Option (C) is correct .. Reason is as follows :

One Comparison
to get minimum



Here , At first level we are Given n elements , out of which we have to find smallest 3 numbers.

we compare 2 -2 elements as shown in figure & get $n/2$ elements at Second level.

Note: Minimum element is in these $n/2$ elements.

So, comparisons for this is $n/2$..

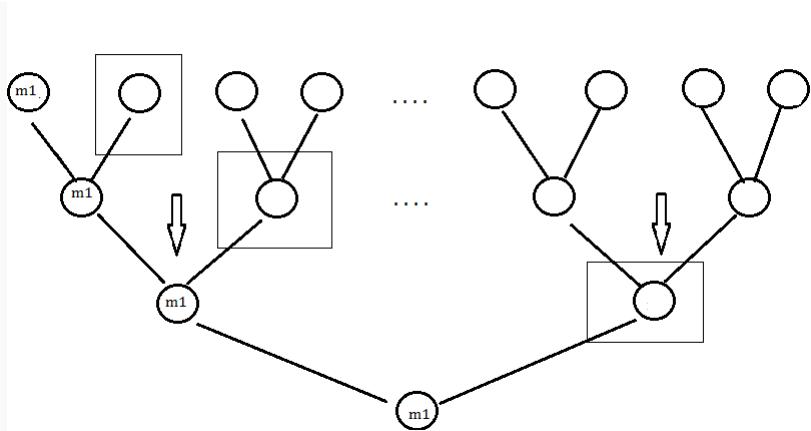
Similarly for next level we have $n/4$ Comparisons & $n/2$ elements..and so on..

Total Comparisons till now is $n/2 + n/4 + n/8 + \dots + 4 + 2 + 1 = (2^{\log n} - 1) = n-1$ {Use G.P. sum}

Now we have to get smallest 3..

We have 1st smallest at last level already $\Rightarrow 0$ Comparison for this..

$\Rightarrow 2^{\text{nd}}$ & 3^{rd} smallest can be found in $O(\log n)$ time as shown below:



Minimum Element must have descended down from some path from top to Bottom..

=> SQUARES represent Candidates for 2nd minimum..

every element that is just below m_1 (first minimum) is a candidate for second minimum..

So, $O(\log n)$ Comparisons for finding second smallest..

=> similarly for 3rd minimum we get $O(\log n)$ time.. As, every element that is just below 1st & 2nd minimum is a candidate for 3rd minimum

..

9 votes

-- Himanshu Agarwal (15,9k points)

1.9

Np Completeness(15) top

1.9.1 Np Completeness: CMI2015-A-06 top

<http://gateoverflow.in/47041>

Suppose we have constructed a polynomial time reduction from problem A to problem B . Which of the following can we infer from this fact?

- A. If the best algorithm for B takes exponential time, there is no polynomial time algorithm for A .
- B. If the best algorithm for A takes exponential time, there is no polynomial time algorithm for B .
- C. If we have a polynomial time algorithm for A , we must also have a polynomial time algorithm for B .
- D. If we don't know whether there is a polynomial time algorithm for B , there cannot be a polynomial time algorithm for A .

[cmi2015](#) [algorithms](#) [np-completeness](#)

Answer

1.9.2 Np Completeness: GATE1992_02,vi top

<http://gateoverflow.in/561>

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

(vi) 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](#) [np-completeness](#) [algorithms](#)

Answer**1.9.3 Np Completeness: GATE2003-12** [top](#)<http://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](#) [np-completeness](#) [normal](#)**Answer****1.9.4 Np Completeness: GATE2004-30, ISRO2017-10** [top](#)<http://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](#) [np-completeness](#) [easy](#) [isro2017](#)**Answer****1.9.5 Np Completeness: GATE2005-58** [top](#)<http://gateoverflow.in/1381>

Consider the following two problems on undirected graphs:

- α : Given $G(V, E)$, does G have an independent set of size $|V| - 4$?
- β : Given $G(V, E)$, does G have an independent set of size 5?

Which one of the following is TRUE?

- A. α is in P and β is NP-complete
- B. α is NP-complete and β is in P
- C. Both α and β are NP-complete
- D. Both α and β are in P

[gate2005](#) [algorithms](#) [np-completeness](#) [normal](#)**Answer****1.9.6 Np Completeness: GATE2006-16** [top](#)<http://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?

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

[gate2006](#) [algorithms](#) [np-completeness](#) [normal](#)**Answer**

1.9.7 Np Completeness: GATE2006-31 [top](#)

<http://gateoverflow.in/894>

Let SHAM₃ be the problem of finding a Hamiltonian cycle in a graph $G = (V, E)$ with $|V|$ divisible by 3 and DHAM₃ be the problem of determining if a Hamiltonian cycle exists in such graphs. Which one of the following is true?

- A. Both DHAM₃ and SHAM₃ are NP-hard
- B. SHAM₃ is NP-hard, but DHAM₃ is not
- C. DHAM₃ is NP-hard, but SHAM₃ is not
- D. Neither DHAM₃ nor SHAM₃ is NP-hard

[gate2006](#) [algorithms](#) [np-completeness](#) [normal](#)

[Answer](#)

1.9.8 Np Completeness: GATE2008-44 [top](#)

<http://gateoverflow.in/458>

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](#) [np-completeness](#) [normal](#)

[Answer](#)

1.9.9 Np Completeness: GATE2009-14 [top](#)

<http://gateoverflow.in/1306>

Let π_A be a problem that belongs to the class NP. Then which one of the following is TRUE?

- A. There is no polynomial time algorithm for π_A .
- B. If π_A can be solved deterministically in polynomial time, then P = NP.
- C. If π_A is NP-hard, then it is NP-complete.
- D. π_A may be undecidable.

[gate2009](#) [algorithms](#) [np-completeness](#)

[Answer](#)

1.9.10 Np Completeness: GATE2012-4 [top](#)

<http://gateoverflow.in/38>

Assuming P \neq NP, which of the following is **TRUE**?

- A. NP-complete = NP
- B. NP-complete \cap P = \emptyset
- C. NP-hard = NP
- D. P = NP-complete

[gate2012](#) [algorithms](#) [np-completeness](#)

[Answer](#)

1.9.11 Np Completeness: GATE2013_18 [top](#)

<http://gateoverflow.in/1440>

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

1. The problem of determining whether there exists a cycle in an undirected graph is in P.
2. The problem of determining whether there exists a cycle in an undirected graph is in NP.
3. If a problem A is NP-Complete, there exists a non-deterministic polynomial time algorithm to solve A.

- (A) 1, 2 and 3 (B) 1 and 2 only (C) 2 and 3 only (D) 1 and 3 only

gate2013 algorithms np-completeness normal

[Answer](#)

1.9.12 Np Completeness: TIFR2010-B-39 [top](#)

<http://gateoverflow.in/18754>

Suppose a language L is **NP** complete. Then which of the following is FALSE?

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

tifr2010 algorithms np-completeness

[Answer](#)

1.9.13 Np Completeness: TIFR2011-B-37 [top](#)

<http://gateoverflow.in/20923>

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 $a^b \pmod{b}$. 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 np-completeness

[Answer](#)

1.9.14 Np Completeness: TIFR2013-B-7 [top](#)

<http://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 np-completeness

[Answer](#)

1.9.15 Np Completeness: TIFR2017-B-2 [top](#)

<http://gateoverflow.in/85673>

Consider the following statements:

Checking if a given *undirected* graph has a cycle is in **P**

Checking if a given *undirected* graph has a cycle is in **NP**

Checking if a given *directed* graph has a cycle is in **P**

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.

Only i and ii

Only ii and iv

Only ii, iii, and iv

Only i, ii and iv

All of them

tifr2017 | algorithms | np-completeness

[Answer](#)

Answers: Np Completeness

1.9.1 Np Completeness: CMI2015-A-06 [top](#)



Selected Answer

Problem A reduces to Problem B,

Option B) If the best algorithm for A takes exponential time, there is no polynomial time algorithm for B. will be correct.

5 votes

-- Muktinath Vishwakarma (33.9k points)

1.9.2 Np Completeness: GATE1992_02,vi [top](#)



Selected Answer

a. Is NPC and hence NP hard.

b. Is again NP hard (optimization version is NP hard and decision version is NPC). Ref: <http://stackoverflow.com/questions/3907545/how-to-understand-the-knapsack-problem-is-np-complete>

c. Is in P. See the algorithm here based on DFS: http://en.wikipedia.org/wiki/Biconnected_component

d. NPC and hence NP hard.

7 votes

-- Arjun Suresh (287k points)

1.9.3 Np Completeness: GATE2003-12 [top](#)



Selected Answer

C

Ram's reduction shows \sqcap is NP hard.

Shyam's reduction shows \sqcap is in NP.

So NPC.

8 votes

-- Anurag Semwal (7.6k points)

1.9.4 Np Completeness: GATE2004-30, ISRO2017-10 [top](#)



Selected Answer

Option c.

<http://cstheory.stackexchange.com/questions/6864/why-is-2sat-in-p>

5 votes

-- anshu (3k points)

1.9.5 Np Completeness: GATE2005-58 [top](#)

<http://gateoverflow.in/1381>

Selected Answer

Independent Set- a set of vertices in a graph no two of which are adjacent.

Maximal Independent set problem - Given a graph G , find the size of the maximal independent set in it. This problem is NP-hard.

Independent set decision problem - Given a graph G and a number k , does G have an independent set of size k . This problem is NP-complete (NP-hard but in NP).

Now, in the given problem β corresponds to the Independent set decision problem. But there is a difference there. We have 5 instead of k . And this drastically changes the problem statement. We can now give a polynomial time **deterministic** algorithm for β .

- Find all vertex sets of size 5 . We get $|V|C_5$ such vertex sets
- For each of them check if there is any adjacent vertices. This check can be done in constant time if we use an Adjacency matrix representation

Thus the whole time complexity reduces to $|V|C_5$ which is $O(|V|^5)$ and hence polynomial. ($|V|C_k$ is not polynomial but $|V|C_5$ is).

Problem α is asking for an independent set of size $|V| - 4$. This is equivalent to asking if G has a vertex cover* of size 4. Following a similar approach as done for β this problem also can be done in polynomial time.

So, both α and β are in P .

D choice.

Vertex cover of a graph G is the set of vertices such that each edge of the graph is incident on atleast one of those vertices.

Independent Set and Vertex cover Reduction: <https://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/npcomplete.pdf>

7 votes

-- Arjun Suresh (287k points)

1.9.6 Np Completeness: GATE2006-16 [top](#)

<http://gateoverflow.in/977>

Selected Answer

Q cannot be NP hard as no np hard problems(unless they are np) can be polynomial time reducible to np complete. Answer is B, as npc problem can be reducible to np hard problem. But there is confusion if Q is not NP hard then what complexity class it is in!!

10 votes

-- Shaun Patel (6.9k points)

1.9.7 Np Completeness: GATE2006-31 [top](#)

<http://gateoverflow.in/994>

Selected Answer

The only difference between SHAM and DHAM, in SHAM $|V|$ is divisible by 3.. which can be checked in constant amount of time.. So the hardness of the two problems will be the same... Next, finding hamiltonian cycle comes under NPC problem and NPC problem is a subset of NPH, so both are NPH..

So, option (A)

7 votes

-- Vicky Bajoria (4.7k points)

1.9.8 Np Completeness: GATE2008-44 [top](#)

<http://gateoverflow.in/456>

Selected Answer

Subset problem is NP-Complete - there is reduction proof but I don't remember (Can see the below link). So, (C) and (D) are true as an NPC problem is in NP as well as NPH.

https://en.wikipedia.org/wiki/Subset_sum_problem

Now, complexity of Q is $O(nW)$, where W is an integer.

(a) Input is encoded in unary. So, length of input is equal to the value of the input. So, complexity = $O(nW)$ where both n and W are linear multiples of the length of the inputs. So, the complexity is polynomial in terms of the input length. So, (a) is true.

(b) Input is encoded in binary. So, length of W will be $\lg W$. (for $W=1024$, input length will be just 10). So, now W is exponential in terms of the input length of W and $O(nW)$ also becomes exponential in terms of the input lengths. So, Q is not a polynomial time algorithm. So, (B) is false.

9 votes

-- Arjun Suresh (287k points)

1.9.9 Np Completeness: GATE2009-14 [top](#)



Selected Answer

A problem which is in P, is also in NP- so A is false. If problem can be solved deterministically in Polynomial time, then also we can't comment anything about P=NP, we just put this problem in P. So, B also false. C is TRUE because that is the definition of NP-complete.

D is false because all NP problems are not only decidable but decidable in polynomial time using a non-deterministic Turing machine.

7 votes

-- shreya ghosh (3.4k points)

1.9.10 Np Completeness: GATE2012-4 [top](#)



Selected Answer

Answer is (B) NP-complete $\cap P = \emptyset$

Since, $P \neq NP$, there is at least one problem in NP, which is harder than all P problems. Lets take the hardest such problem, say X . Since, $P \neq NP$, $X \notin P$.

Now, by definition, NP-complete problems are the hardest problems in NP and so X problem is in NP-complete. And being in NP, X can be reduced to all problems in NP-complete, making any other NP-complete problem as hard as X . So, since $X \notin P$, none of the other NP-complete problems also cannot be in P.

14 votes

-- Arjun Suresh (287k points)

1.9.11 Np Completeness: GATE2013_18 [top](#)



Selected Answer

Cycle detection in graph is in P as it can be done using a graph traversal ($O(V+E)$)

Ref: <http://www.geeksforgeeks.org/detect-cycle-undirected-graph/>

If a problem is in P then it is also in NP as P is a subset of NP. So, both 1 and 2 are TRUE.

Statement 3 is also true as NP-Complete requires a problem to be in NP and for any problem in NP, we have a non-deterministic polynomial time algorithm.

So, answer is A- 1, 2 and 3 are TRUE.

8 votes

-- Arjun Suresh (287k points)

1.9.12 Np Completeness: TIFR2010-B-39 [top](#)



Selected Answer

Option E leads to a contradiction, hence is false.

We know that L is **NPC**, hence $\in \text{NP}$. If $\text{P} \neq \text{NP}$, then L can't be in P

4 votes

-- Pragy Agarwal (19.3k points)

1.9.13 Np Completeness: TIFR2011-B-37 [top](#)<http://gateoverflow.in/20922>

Selected Answer

The reverse problem can be solved in polynomial time as a^b requires at most $\log b$ recursive calls using the approach given below:

```
pow(int a, int b)
{
    if(b%2)
        return a* pow(a*a, b/2);
    else
        return pow(a*a, b/2);
}
```

Now, the forward problem is also solvable in polynomial time. We need to check for all the roots of n (from \sqrt{n} till $n^{\frac{1}{\log n}}$) whether it is an integer. But each of these check can be done in $\log n$ time using a binary search on the set of integers from $2..n$ and so, the overall complexity will be $(\log n)^2$ which is polynomial in $\log n$ ($\log n$ is the size of input). So, (a) must be the answer.

4 votes

-- gatecse (13.1k points)

1.9.14 Np Completeness: TIFR2013-B-7 [top](#)<http://gateoverflow.in/25668>

Selected Answer

I believe Umang is right, Option B is the correct answer.

Intractability : We are looking for EFFICIENT algorithms.

Intractable Problems are problems that are decidable, although the algorithm to decide that problem might be efficient (P) or inefficient (NP), but at least an algorithm exists to solve these problems.

Here we talk about efficient vs inefficient computations.

Thus the language of problems in P and NP classes is the language of Decidable Problems i.e. Recursive Language.

Undecidability: We are looking for algorithms.

Undecidable Problems are problems for which there is no algorithm to solve these problems.

Here we talk about what can or can not be computed.

The language of Undecidable Problems are "Recursively Enumerable but not recursive languages" & "Not Recursively Enumerable Languages".

Clearly we can talk about the intractability of any problem if we know at least one algorithm to solve the problem, if there is no algorithm to solve a problem how can we talk about efficiency?

Halting Problem is undecidable.

I guess, all other problems mentioned here are decidable.

I don't know the most efficient algorithms to solve these problems but at least I can say that Brute force approach will work on all the other options except the Halting Problem.

What P = NP implies?

"Any problem that is solved by a non deterministic Turing machine in polynomial time also be solved by some deterministic Turing machine in polynomial time, even if the degree of the polynomial is higher."

and

There is neither a Non Deterministic Turing Machine nor Deterministic Turing Machine that can solve the Halting Problem.

So **any inference about P & NP is not going to affect the solvability of Halting Problem, since it is undecidable.**

6 votes

-- Anurag Pandey (12.8k points)

1.9.15 Np Completeness: TIFR2017-B-2 [top](#)

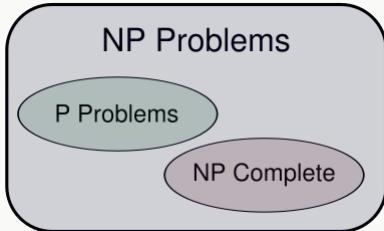
<http://gateoverflow.in/95673>



Selected Answer

E) all of them
because all of them can be solved by Depth first traversal.

Every P problem is a subset of NP.



6 votes

-- Motamarri Anusha (11.4k points)

1.10

Numerical Computation(2) [top](#)

<http://gateoverflow.in/1915>

1.10.1 Numerical Computation: GATE2014-1-37 [top](#)

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 numerical-computation

Answer

1.10.2 Numerical Computation: TIFR2014-B-20 [top](#)

<http://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 numerical-computation

Answer

Answers: Numerical Computation

1.10.1 Numerical Computation: GATE2014-1-37 [top](#)

<http://gateoverflow.in/1915>



Selected Answer

Suppose x is no of coins of 11gm and y is no of 10 gm coins

so according to question $11x + 10y = 323 \dots(1)$

$X+Y=31 \dots(2)$ solving (1),(2) we get $X=13$ and $Y=18$ so here coins of 11gm are 13 ,and one and only possible combination for 13= 1 coin from bag1+4 coins from bag3 +8 coins from bag4...

so product of label of bags will be $=1*3*4=12..$

16 votes

-- sonam vyas (12.8k points)

There are 5 bags , I assumed initially all bags are having 10 gm coins, and picked them as per the given condition

1,2,4,8,16 of all bags have 10 gm coins then total weight will come to

$10 + 20 + 40 + 80 + 160 = 310$ but total weight should be 323, but 13 is less, i divided 13 into $1 + 4 + 8$
 $11 + 20 + 44 + 88 + 160$, means 1st, 3rd and 4th bags have 11 gm coins. so product of labels will be $1*3*4 = 12$

15 votes

-- Manu Thakur (5.8k points)

1.10.2 Numerical Computation: TIFR2014-B-20 [top](#)

<http://gateoverflow.in/27354>



Selected Answer

OPTION d is correct

Here the list is (48, 99, 120, 165 ,273.)

$\text{Gcd}(48,99)=3$,means if we subtract($99-48=51$) that no is also % 3,

so the no's like (3,6,9----99) are added , total no's= $99/3=33$

//y $\text{Gcd}(48,120)=24$,so the no's %24 are added like (24,48,---120) ,total no's= $120/24=5$

//y $\text{Gcd}(48,165)=3$,so the nos(3,6,9,--24--48--99--120---165) are added ,totally $165/3=55$

at end $\text{Gcd}(48,273)=3$,so the no's(3,6,9--24---48--99---120--165---273) are added(which covers all the above no's)

so total no"s added to this list= $273/3=91$

8 votes

-- venky.victory35 (643 points)

1.11

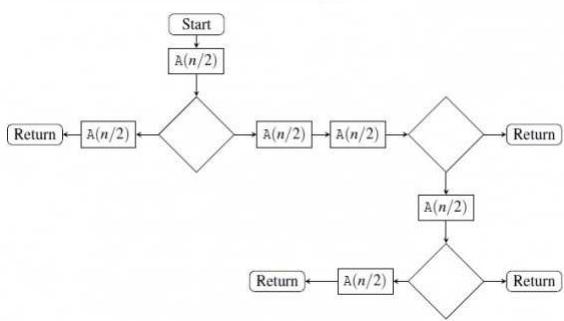
Recurrence(29) [top](#)

1.11.1 Recurrence: GATE 2016-2-39 [top](#)

<http://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

[Answer](#)

1.11.2 Recurrence: GATE1987-10a [top](#)

<http://gateoverflow.in/82450>

Solve the recurrence equations

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

gate1987 | algorithms | recurrence

[Answer](#)

1.11.3 Recurrence: GATE1990-17a [top](#)

<http://gateoverflow.in/86878>

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

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

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

gate1990 | descriptive | algorithms | recurrence

[Answer](#)

1.11.4 Recurrence: GATE1992-07a [top](#)

<http://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

[Answer](#)

1.11.5 Recurrence: GATE1992-07b [top](#)

<http://gateoverflow.in/4360>

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](#)

[Answer](#)

1.11.6 Recurrence: GATE1993_15 [top](#)

<http://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
```

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](#)

[Answer](#)

1.11.7 Recurrence: GATE1994-1.7, ISRO2017-14 [top](#)

<http://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](#)

[Answer](#)

1.11.8 Recurrence: GATE1996_2.12 [top](#)

<http://gateoverflow.in/2741>

The recurrence relation

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

has the solution $T(n)$ equal to

- $O(n)$
- $O(\log n)$
- $O\left(n^{\frac{3}{4}}\right)$
- None of the above

gate1996 algorithms recurrence normal

Answer

1.11.9 Recurrence: GATE1997_4.6 [top](#)

<http://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?

- $T(n) = O\sqrt{n}$
- $T(n) = O(n)$
- $T(n) = O(\log n)$
- None of the above

gate1997 algorithms recurrence normal

Answer

1.11.10 Recurrence: GATE1998-6a [top](#)

<http://gateoverflow.in/44584>

Solve the following recurrence relation

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

$$x_1 = 2$$

gate1998 algorithms recurrence descriptive

Answer

1.11.11 Recurrence: GATE2002-1.3 [top](#)

<http://gateoverflow.in/807>

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

- 2^k
- $\frac{(3^{k+1}-1)}{2}$
- $3^{\log_2 k}$
- $2^{\log_3 k}$

gate2002 algorithms recurrence normal

Answer

1.11.12 Recurrence: GATE2002-2.11 [top](#)

<http://gateoverflow.in/841>

The running time of the following algorithm

Procedure A(n)

If $n \leq 2$ 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

Answer

1.11.13 Recurrence: GATE2003-35 [top](#)

<http://gateoverflow.in/925>

Consider the following recurrence relation

$$T(1) = 1$$

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

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

Answer

1.11.14 Recurrence: GATE2004-83, ISRO2015-40 [top](#)

<http://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

Answer

1.11.15 Recurrence: GATE2004-84 [top](#)

<http://gateoverflow.in/1078>

The recurrence equation

$$T(1) = 1$$

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

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

Answer

1.11.16 Recurrence: GATE2004-IT-57 [top](#)<http://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

Answer

1.11.17 Recurrence: GATE2005-IT-51 [top](#)<http://gateoverflow.in/3812>

Let $T(n)$ be a function defined by the recurrence

$$T(n) = 2T(n/2) + \sqrt{n} \text{ for } n \geq 2 \text{ and} \\ T(1) = 1$$

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

Answer

1.11.18 Recurrence: GATE2006-51, ISRO2016-34 [top](#)<http://gateoverflow.in/1829>

Consider the following recurrence:

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

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

Answer

1.11.19 Recurrence: GATE2008-78 [top](#)<http://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. $x_n = x_{n-1} + x_{n-2}$

[gate2008](#) [algorithms](#) [recurrence](#) [normal](#)

[Answer](#)

1.11.20 Recurrence: GATE2008-79 [top](#)

<http://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](#)

[Answer](#)

1.11.21 Recurrence: GATE2008-IT-44 [top](#)

<http://gateoverflow.in/3354>

When $n = 2^{2k}$ for some $k \geq 0$, 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](#)

[Answer](#)

1.11.22 Recurrence: GATE2009-35 [top](#)

<http://gateoverflow.in/1324>

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](#)

[Answer](#)

1.11.23 Recurrence: GATE2012-16 [top](#)

<http://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

Answer

1.11.24 Recurrence: GATE2014-2-13 [top](#)

<http://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

Answer

1.11.25 Recurrence: GATE2015-1_49 [top](#)

<http://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. $a_{n-2} + 2a_{n-1} + 2^{n-2}$
 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

Answer

1.11.26 Recurrence: GATE2015-3_39 [top](#)

<http://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 $\text{get}(6)$ function is being called in $\text{main}()$ then how many times will the $\text{get}()$ function be invoked before returning to the $\text{main}()$?

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

gate2015-3 | algorithms | recurrence | normal

Answer

1.11.27 Recurrence: GATE2017-2-30 [top](#)

<http://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

[Answer](#)

1.11.28 Recurrence: TIFR2014-B-11 [top](#)

<http://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

[Answer](#)

1.11.29 Recurrence: TIFR2015-B-1 [top](#)

<http://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 $O(\log n \cdot \log \log n)$ but not $O(\log n)$.
- c. $T(n)$ is $O(\log^{3/2} n)$ but not $O(\log n \cdot \log \log n)$.
- d. $T(n)$ is $O(\log^2 n)$ but not $O(\log^{3/2} n)$.
- e. $T(n)$ is $O(\log^2 n \cdot \log \log n)$ but not $O(\log^2 n)$.

tifr2015 | algorithms | recurrence | time-complexity

[Answer](#)

Answers: Recurrence

1.11.1 Recurrence: GATE 2016-2-39 [top](#)

<http://gateoverflow.in/39581>



Selected Answer

If they are asking for worst case complexity hence,
By calling $A(n)$ we get $A(n/2)$ 5 times,

$$A(n) = 5A(n/2) + O(1)$$

Hence by applying masters theorem,
Case 1 : $a > b^k$

$$n^{\log_2 5}$$

Thus value of alpha will be 2.32

25 votes

-- Shashank Chavan (3.3k points)

1.11.2 Recurrence: GATE1987-10a [top](#)

<http://gateoverflow.in/82450>



Selected Answer

The handwritten solution shows the recursive expansion of $T(n)$:

$$\begin{aligned} T(n) &= T(n-1) + n \\ &= T(n-2) + (n-1) + n \\ &= T(n-3) + (n-2) + (n-1) + n \\ &= T(n-3) + 3n - \{1+2\} \\ &= T(n-4) + 4n - \{1+2+3\} \\ &\vdots \\ &= T(1) + (n-1)n - \left\{ \frac{1+2+\dots+(n-1)}{2} \right\} \\ &\approx O(n^2) \end{aligned}$$

4 votes

-- kirti singh (3.3k points)

1.11.3 Recurrence: GATE1990-17a [top](#)

<http://gateoverflow.in/86878>

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

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

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

substituting value of $T(n-1)$ from eqn (2) in eqn (1)

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

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

now substituting value of $T(n-2)$ in above eqn

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

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

so on

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

$$T(n) = \frac{n+1}{n-k+1} T(n-k) + (n+1) * \left(\frac{1}{n} + \frac{1}{n-1} + \dots + \frac{1}{n-k+2} \right) + 1$$

now let $n-k=1$ so $k = n-1$, substitute value of k in above eqn

$$T(n) = \frac{n+1}{n-(n-1)+1} T(1) + (n+1) * \left(\frac{1}{n} + \frac{1}{n-1} + \dots + \frac{1}{n-(n-1)+2} \right) + 1$$

$$T(n) = \frac{n+1}{2} + (n+1) * \left(\frac{1}{n} + \frac{1}{n-1} + \dots + \frac{1}{3} \right) + 1$$

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

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

$$\mathbf{T(n) = (n+1)*H_n - n}$$

Now $H_n \approx \log n + \gamma$

where γ is the Euler-Mascheroni constant.

$$\mathbf{T(n) = O(n \log n)}$$

2 votes

-- Digvijaysingh Gautam (7.8k points)

1.11.4 Recurrence: GATE1992-07a [top](#)

<http://gateoverflow.in/586>



Selected Answer

1 - The function $F(n)$ is NOT a recursive function. You can't have a recurrence relation for it in the first place!

2 - $F(n)$ calculates $(n-1)^n$.

The equivalent C++ code is as follows: ([You can try it out here: http://ideone.com/w0u4lk](http://ideone.com/w0u4lk))

```
long F(long n) {
    long F1 = 1;

    if(n==1) { return 3; }
    else {
        for(long i = 1; i <= n; i++) {
            long C = 0;
            // Note: the before For loop only has one line
            for(long j = 1; j <= n-1; j++) { C = C+1; }
            // At the end of this for loop, C will be = (n-1)
            F1 = F1 * C;
        }
    }
    return F1;
}
```

It is clear that the inner for loop can be replaced by a single statement as follows:

```
long F(long n) {
    long F1 = 1;

    if(n==1) { return 3; }
    else {
        for(long i = 1; i <= n; i++)
            F1 = F1 * (n-1);
    }
    return F1;
}
```

And this calculates
 $(n-1)^n$

7 votes

-- Pragy Agarwal (19.3k points)

1.11.5 Recurrence: GATE1992-07b [top](#)

<http://gateoverflow.in/4360>

$$F(n) = (n-1)^n \quad \text{When } n >= 2$$

$$F(n) = 3 \quad \text{When } n == 1$$

1 upvotes

-- Rdr Deva (1.6k points)

1.11.6 Recurrence: GATE1993_15 [top](#)

<http://gateoverflow.in/2312>



Selected Answer

$$a_n = a_{n-1} + n-1 \quad (\text{n-1 comparisons for } n \text{ numbers})$$

$$a_n = a_{n-2} + (n-2) + (n-1)$$

$$a_n = a_{n-3} + (n-3) + (n-2) + (n-1)$$

.

.

$$a_n = a_{n-n} + (n-n) + (n-(n-1)) + \dots + (n-3) + (n-2) + (n-1)$$

$$a_n = 0 + 1 + 2 + \dots + (n-3) + (n-2) + (n-1)$$

$$\text{which given } a_n = \frac{(n-1) \times (n)}{2}$$

1 upvotes

-- Rajarshi Sarkar (34.4k points)

1.11.7 Recurrence: GATE1994-1.7, ISRO2017-14 [top](#)

<http://gateoverflow.in/2444>



Selected Answer

It is B. searching for only one half of the list. leading to $T(n/2)$ + constant time in comparing and finding mid element.

1 upvotes

-- Gate Keeda (18.8k points)

1.11.8 Recurrence: GATE1996_2.12 [top](#)

<http://gateoverflow.in/2741>



Selected Answer

Answer: A

According to Master theorem,

$T(n) = aT(\frac{n}{b}) + f(n)$ can be expressed as:

$$T(n) = [n^{\log_b a}] [T(1) + u(n)]$$

where $u(n) = \Theta(h(n))$ where $h(n) = \frac{f(n)}{n^{\log_b a}} = \frac{n}{n^{\log_b 3}} = n^{1-\log_b 3}$ as $h(n) = n^r$ where $r > 0$.

$$\text{So, } T(n) = [n^{\log_b a}] [T(1) + u(n)] = T(n) = [n^{\log_b 3}] [T(1) + \Theta(n^{1-\log_b 3})] = \Theta(n^1) .$$

1 upvotes

-- Rajarshi Sarkar (34.4k points)

1.11.9 Recurrence: GATE1997_4.6 [top](#)

<http://gateoverflow.in/2247>



Selected Answer

answer B

using master method (case 1)

where $a = 2, b = 2$

$$O(n^{1/2}) < O(n^{\log_b a})$$

$$O(n^{1/2}) < O(n^{\log_2 2})$$

7 votes

-- ankitrokdeonsns (9.1k points)

1.11.10 Recurrence: GATE1998-6a [top](#)

<http://gateoverflow.in/4458>



Selected Answer

$$T(n) = 2T(n-1) - 1 = 2(2T(n-2) - 1) - 1 = 2^2 T(n-2) - 2 - 1 = 2^2 (2T(n-3) - 1) - 2 - 1 = 2^3 T(n-3) - 2^2 - \dots$$

6 votes

-- srestha (55.2k points)

1.11.11 Recurrence: GATE2002-1.3 [top](#)

<http://gateoverflow.in/807>



Selected Answer

Let $x = 2^k$

$$T(x) = 3T\left(\frac{x}{2}\right) + 1$$

We can apply Master's theorem case 1 with $a = 3$ and $b = 2$ as $f(x) = 1 = O(x^{\log_2 3 - \epsilon}), \epsilon > 0$

$$\text{So, } T(x) = \Theta(x^{\log_2 3}) = \Theta(2^{\log_2 3}) = \Theta(2^{\log_2 3^k}) = \Theta(3^k)$$

So, only option possible is B.

We can also directly solve as follows:

$$T(x) = 3T\left(\frac{x}{2}\right) + 1 = 9T\left(\frac{x}{4}\right) + 1 + 3\dots = 3^{\log_2 2^k} + (1 + 3 + 9 + \dots + 3^{\log_2 2^k - 1}) \quad (\text{recursion depth is } \log_2 x \text{ and } x = 2^k)$$

OR

$$T(2^k) = 3T(2^{k-1}) + 1 = 3^2 T(2^{k-2}) + 1 + 3\dots = 3^k T(2^{k-k}) + (1 + 3 + 9 + \dots + 3^{k-1}) \quad (\text{recursion depth is } k) = 3^k +$$

16 votes

-- Arjun Suresh (287k points)

1.11.12 Recurrence: GATE2002-2.11 [top](#)

<http://gateoverflow.in/841>



Selected Answer

The complexity will be the number of times the recursion happens which is equal to the number of times we can take square root of n recursively, till n becomes 2.

$$T(n) = T(\lceil \sqrt{n} \rceil) + 1$$

$$T(2) = 1$$

$$T(2^2) = T(2) + 1 = 2$$

$$T(2^{2^2}) = T(4) + 1 = 3$$

$$T(2^{2^3}) = T(16) + 1 = 4$$

So, $T(n) = \lg \lg n + 1 = O(\log \log n)$

11 votes

-- Arjun Suresh (287k points)

1.11.13 Recurrence: GATE2003-35 [top](#)

<http://gateoverflow.in/925>



Selected Answer

$$\begin{aligned} T(m^2) &= T(m^2 - 1) + \lfloor \sqrt{(m^2)} \rfloor \\ &= T(m^2 - 2) + \lfloor \sqrt{(m^2 - 1)} \rfloor + \lfloor \sqrt{(m^2)} \rfloor \\ &= T(m^2 - 3) + \lfloor \sqrt{(m^2 - 2)} \rfloor + \lfloor \sqrt{(m^2 - 1)} \rfloor + \lfloor \sqrt{(m^2)} \rfloor \\ &= \dots \\ &= T(1) + \lfloor \sqrt{(2)} \rfloor + \lfloor \sqrt{(3)} \rfloor + \dots + \lfloor \sqrt{(m^2)} \rfloor \end{aligned}$$

$= 3 \times 1 + 5 \times 2 + \dots + (2m - 1) \times (m - 1) + m$ (We are taking floor of square root of numbers, and between successive square roots number of numbers are in the series 3, 5, 7... like 3 numbers from 1..4, 5 numbers from 5 - 9 and so on).

We can try out options here or solve as shown at end:

Put $m = 5, T(25) = 3 \times 1 + 5 \times 2 + 7 \times 3 + 9 \times 4 + 5 = 75$

Option A: 59

Option B: 75

Option C: non-integer

Option D: 297.5

So, answer must be B.

$$T(m^2) = 3 \times 1 + 5 \times 2 + \dots + (2m - 1) \times (m - 1) + m = m + \sum_i = 1^{m-1} (2i + 1). (i) = m + \sum_i = 1^{m-1} 2i^2 + i = m$$

[Sum of the first
 n natural numbers
 $= \frac{n(n+1)}{2}$.]

Sum of the squares of first
 n natural numbers
 $= \frac{n(n+1)(2n+1)}{6}$.]

11 votes

-- Arjun Suresh (287k points)

1.11.14 Recurrence: GATE2004-83, ISRO2015-40 [top](#)

<http://gateoverflow.in/1077>



Selected Answer

option D

```
int recursive (int n) {
    if(n == 1)           // takes constant time say 'A' time
        return (1);       // takes constant time say 'A' time
    else
        return (recursive (n-1) + recursive (n-1)); // takes T(n-1) + T(n-1) time
}
```

$T(n) = 2T(n - 1) + a$ is the recurrence equation found from the pseudo code .

Solving the Recurrence Equation By Back Substitution Method

$$T(n) = 2T(n-1) + a \quad \text{----- (equation 1)}$$

$$T(n-1) = 2T(n-2) + a$$

$$T(n-2) = 2T(n-3) + a$$

We can re write Equation 1 as

$$\begin{aligned} T(n) &= 2[2T(n-2) + a] + a = 4T(n-2) + 3a \\ &= 2[2T(n-3) + a] + 3a = 8T(n-3) + 7a \\ &\quad \dots \\ &= 2^k T(n-k) + 2^{k-1} a \end{aligned} \quad \text{----- (Equation 2)}$$

On Substituting Limiting Condition

$$T(1) = 1 \text{ implies } n - k = 1 \implies k = n - 1$$

Therefore Equation 2 becomes

$$2^{n-1} + 2^{n-2} a = O(2^n)$$

14 votes

-- pC (20.7k points)

1.11.15 Recurrence: GATE2004-84 [top](#)

<http://gateoverflow.in/1078>



Selected Answer

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

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

$$= n(1 + 2 + \dots + 2^{n-1}) - (1 \cdot 2 + 2 \cdot 2^2 + 3 \cdot 2^3 + \dots + (n-1) \cdot 2^{n-1})$$

$$= n(2^n - 1) - (n \cdot 2^n - 2^{n+1} + 2)$$

$$= 2^{n+1} - n - 2$$

23 votes

-- suraj (5k points)

$$T(1) = 1$$

$$T(2) = 4$$

$$T(3) = 11$$

$$T(4) = 26$$

$$T(5) = 57$$

$$T(6) = 120$$

$$T(7) = 247$$

So,

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

27 votes

-- Arjun Suresh (287k points)

1.11.16 Recurrence: GATE2004-IT-57 [top](#)

<http://gateoverflow.in/3700>



Selected Answer

answer B

5 votes

-- Sankaranarayanan P.N (11k points)

1.11.17 Recurrence: GATE2005-TT-51 [top](#)



Selected Answer

Option C it can be done by Master theorem.

$$n^{\log_2 a} = n^{\log_2 2} = n.$$

$$f(n) = \sqrt{n} = n^{\frac{1}{2}}.$$

So, $f(n) = O(n^{\log_2 a - \epsilon})$ is true for any real ϵ , $0 < \epsilon < \frac{1}{2}$. Hence Master theorem Case 1 satisfied,

$$T(n) = \Theta(n^{\log_2 a}) = \Theta(n).$$

11 votes

-- Bhagirathi Nayak (13.1k points)

1.11.18 Recurrence: GATE2006-51, ISRO2016-34 [top](#)



Selected Answer

$$T(n) = 2T\left(n^{\frac{1}{2}}\right) + 1 = 2\left(2T\left(n^{\frac{1}{2^2}}\right) + 1\right) + 1 = 4 \times T\left(n^{\frac{1}{2^2}}\right) + 5 = 8 \times T\left(n^{\frac{1}{2^3}}\right) + 13 \dots = 2^{(\lg \lg n)} + 2 \times \lg \lg n + 1$$

$n^{\frac{1}{2^k}} = 2$ (Putting 2 so that we can take log. One more step of recurrence can't change the complexity.) $\implies \frac{1}{2^k} \lg n = 1$

So, answer is B, $T(n) = \Theta(\log n)$

17 votes

-- Arjun Suresh (287k points)

1.11.19 Recurrence: GATE2008-78 [top](#)



Selected Answer

0 1 -2
01 10 11 -3
010 011 101 110 111 -5
0101 0110 0111 1010 1011 1101 1110 1111 -8

So, $x_n = x_{n-1} + x_{n-2}$ (For all the strings ending in 1, we get two new strings and for all strings ending in 0, we get a new string. So, the new set of strings for $n+1$, will have exactly n strings ending in 1)

$$x_5 = 8 + 5 = 13$$

10 votes

-- Arjun Suresh (287k points)

1.11.20 Recurrence: GATE2008-79 [top](#)



Selected Answer

Number of binary strings of length n that contain no consecutive 0s, following will be the required recurrence relation:

$T(n) = T(n-1) + T(n-2)$ $n > 2$
 base condition $T(1) = 2$ and $T(2) = 3$

$T(1) = 2$ There will be 2 strings of length 1, i.e 0 & 1
 $T(2) = 3$ There will be 3 strings of length 2, i.e. 01,10,11
 $T(3) = T(1) + T(2) = 2+3 = 5$
 $T(4) = T(3) + T(2) = 5 + 3 = 8$
 $T(5) = T(4) + T(3) = 8 + 5 = 13$

Hence, answer is 13, but no option matches!

3 votes

-- Vijay Thakur (14.7k points)

1.11.21 Recurrence: GATE2008-IT-44 [top](#)

<http://gateoverflow.in/3354>



Selected Answer

$$\begin{aligned} T(n) &= \sqrt{2}T\left(\frac{n}{2}\right) + \sqrt{n} \\ &= \sqrt{2}^2 T\left(\frac{n}{2^2}\right) + \sqrt{2}\sqrt{\frac{n}{2}} + \sqrt{n} \\ &\dots \\ &= \sqrt{2^{\lg n}} T(1) + \lg n \sqrt{n} \\ &= \sqrt{n} + \lg n \sqrt{n} \\ &= \sqrt{n} (\lg n + 1) \end{aligned}$$

If we use Master theorem we get option B. But one must know that Master theorem is used to find the asymptotic bound and not an EXACT value. And in the question here it explicitly says "**evaluates to**".

24 votes

-- Arjun Suresh (287k points)

1.11.22 Recurrence: GATE2009-35 [top](#)

<http://gateoverflow.in/1321>



Selected Answer

$$a = 1, b = 3, \log_b a = 0$$

$$\text{So } n^{\log_b a} = n^0 = 1$$

$$f(n) = n$$

$$\text{So, } f(n) = \Omega(1)$$

To, check Master theorem case 3, we need $c > 0$,

$$f(n/3) \leq c f(n)$$

$$c = 1$$

So using case three of master theorem

$$T(n) = \Theta(f(n)) = \Theta(n)$$

answer is a

9 votes

-- Pooja Palod (31.2k points)

1.11.23 Recurrence: GATE2012-16 [top](#)

<http://gateoverflow.in/48>



Selected Answer

Recurrence relation for **Towers of Hanoi** is

$$T(1) = 1$$

$$T(n) = 2 T(n-1) + 1$$

So Answer should be (D)

13 votes

-- Narayan Kunal (419 points)

1.11.24 Recurrence: GATE2014-2-13 [top](#)



Selected Answer

$$f(n) = \log n$$

$$a = 2, b = 2 \implies n^{\log_b a} = n$$

So, $f(n) = \log n = O(n^{1-\epsilon})$, we can take any ϵ from 0-1 for example 0.5 which gives $\log n = O(\sqrt{n})$, whose proof is given here: <http://math.stackexchange.com/questions/145739/prove-that-log-n-is-sqrt-n>

So, Master theorem Case 1, and answer will be $O(n^{\log_2 2}) = O(n)$

Alternate way:

$$T(1) = 1 \\ T(2) = 2T(1) + \log 2 = 3 \\ T(3) = 2T(2) + \log 3 = 3n \\ T(4) = 2T(3) + \log 4 = 3n - 2T(2) \\ T(5) = 2T(4) + \log 5 = 3n - 4T(3) \\ T(6) = 2T(5) + \log 6 = 3n - 6T(4) \\ T(7) = 2T(6) + \log 7 = 3n - 8T(5) \\ T(8) = 2T(7) + \log 8 = 3n - 10T(6) \\ \vdots$$

The second term being subtracted is growing at a lower rate than the first term. So, we can say $T(n) = O(n)$.

18 votes

-- Arjun Suresh (287k points)

1.11.25 Recurrence: GATE2015-1_49 [top](#)



Selected Answer

Counting the number of bit strings NOT containing two consecutive 1's. (It is easy to derive a recurrence relation for the NOT case as shown below.)

0 1

00 01 10 - 3 (append both 0 and 1 to any string ending in 0, and append 0 to any string ending in 1)
000 001 010 100 101 - 5 (all strings ending in 0 give two strings and those ending in 1 give 1 string)

0000 0001 0010 0100 0101 1000 1001 1010 - 8

....

$a_n' = a_{n-1}' + a_{n-2}'$ (where a_n denote the number of bit strings of length n containing two consecutive 1s)

$$2^n - a_n = (2^{n-1} - a_{n-1}) + (2^{n-2} - a_{n-2})$$

$$a_n = 2^{n-2}(4 - 2 - 1) + a_{n-1} + a_{n-2}$$

$$a_n = a_{n-1} + a_{n-2} + 2^{n-2}$$

A choice.

19 votes

-- Arjun Suresh (287k points)

Strings of Length	Number of valid Strings
1	0
2	1
3	3
4	8

apply value putting and try to satisfy options using these values.
only option A holds good.

answer = **option A**

 10 votes

-- Amar Vashishth (27.9k points)

1.11.26 Recurrence: GATE2015-3_39 [top](#)

<http://gateoverflow.in/8498>

Selected Answer

$T(n) = T(n-1) + T(n-3) + 2$, here $T(n)$ denotes the number of times a recursive call is made for input n . 2 denotes the two direct recursive calls.

$T(n \leq 0) = 0$
 $T(1) = 2$
 $T(2) = 4$
 $T(3) = 6$
 $T(4) = 10$
 $T(5) = 16$
 $T(6) = 24$

So, answer is $24 + 1$ call from main = 25.

 27 votes

-- Arjun Suresh (287k points)

1.11.27 Recurrence: GATE2017-2-30 [top](#)

<http://gateoverflow.in/118623>

Selected Answer

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

Put, $n = 2^m$

$$T(2^m) = 2T(2^{m/2}) + 1$$

$$\text{put, } T(2^m) = s(m)$$

$$s(m) = 2s(m/2) + 1$$

Using case 1 of master method ,

$$= \Theta(m) = \Theta(\log n)$$

<http://gateoverflow.in/1829/gate2006-51-isro2016-34?show=37791#c37791>

 6 votes

-- jatin saini (1.6k points)

1.11.28 Recurrence: TIFR2014-B-11 [top](#)

<http://gateoverflow.in/27308>

OPTION b is false

c) apply Akra-Bazzi method .

for ex k=4, then eq is $T(n)=T(n/4)+T(3n/4)+n$

let $g(n)=n$ and $a_1=1, b_1=1/4, a_2=1, b_2=3/4$

then $\sum a_i b_i p = 1 \cdot (1/4)p + 1 \cdot (3/4)p = 1 \rightarrow p=1$

then $T(n)=O(np(1+1∫n \ g(u) / u_1+p).du)$

$$=n(1+1∫n \ u/u_2 du)$$

$$=n(1+\log n)$$

$=O(n \ log n)$::option c is correct

d) apply back substitution

if $k=5$ $T(n)=T(n/5) + T(3n/4)+n$ ---eq1

$T(n/5)=T(n/5^2) + T((1/5)(3/4)n)+n/5$

$T(3n/4)=T((3/4)((1/5)n)+T((3/4)2n)+3n/4$ substitute in eq1

we got $T(n)=T(n/5^2) + 2 T((1/5)(3/4)n)+T((3/4)2n)+n(1/5+3/4) +n$

in the next we get $T(n)=T(n/5^3) + \dots + n(1/5+3/4)2+n(1/5+3/4) +n$

$$T(n)=T(n/5^3) + \dots + n(19/20)2+n(19/20) +n$$

$$T(n)=T(n/5^k) + \dots + n(1+(19/20)+(19/20)2) + \dots + (19/20)^{k-1}$$

$$T(n)=T(n/5^k) + \dots + n(-(19/20)^k + 1)/(1/20))$$

$$n/5^k=1 \rightarrow k=\log n$$

$$\therefore T(n)=1+20n-20n(19/20)\log n$$

$\therefore T(n)=O(n)$ which inturn $O(n \ log n)$ both d,e are correct

a) by observing option a & b $T(n)=(n \ \sqrt{n})$ $T(n)=O(n \ log n)$ and $\log n=O(\sqrt{n})$

so if option b is correct then option a is also correct ----> so option b is false (we can eliminate)

3 votes

-- venky.victory35 (643 points)

1.11.29 Recurrence: TIFR2015-B-1 [top](#)

<http://gateoverflow.in/2965>



Selected Answer

Let $n = 2^k$

$$T(2^k) = 2T(2^{k/2}) + k$$

$$\text{Let } T(2^k) = S(k) \implies T(2^{k/2}) = S(k/2)$$

$$S(k) = 2S(k/2) + k$$

This gives $S(k) = \Theta(k \log k)$ (Master theorem case 2)

$$= \Theta(\log n \log \log n)$$

So ans is b

8 votes

-- Pooja Palod (31.2k points)

1.12

Searching(7) [top](#)

1.12.1 Searching: GATE1996-18 [top](#)

<http://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 not in the array')
else
  writeln ('x is not in the array')
end;
```

[gate1996](#) [algorithms](#) [searching](#) [normal](#)

Answer

1.12.2 Searching: GATE1996-2.13, ISRO2016-28 [top](#)

<http://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](#)

Answer

1.12.3 Searching: GATE2008-84 [top](#)

<http://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](#)

Answer

1.12.4 Searching: GATE2008-85 [top](#)

<http://gateoverflow.in/4358>

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));

[gate2008](#) [algorithms](#) [searching](#) [normal](#)

[Answer](#)

1.12.5 Searching: GATE2017-1-48 [top](#)

<http://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](#)

[Answer](#)

1.12.6 Searching: TIFR2010-B-29 [top](#)

<http://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, $A[1] \leq A[3] \leq \dots \leq A[2n-1]$.

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

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](#)

[Answer](#)

1.12.7 Searching: TIFR2012-B-11 [top](#)

<http://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

[Answer](#)

Answers: Searching

1.12.1 Searching: GATE1996-18 [top](#)

<http://gateoverflow.in/2770>



Selected Answer

the code is wrong here $k=(i+j) / 2;$

```
if(a[k] < x) then i = k ;
else j = k ;
```

the (correct) code should be ..

```
k=(i+j) / 2;
```

```
if(a[k] < x) then i = k + 1 ;
else j = k - 1 ;
```

try an examplewith given code in question

let an array a[1,2,3,4,5,6,7,8,9,10]
index number 1,2,3,4,5,6,7,8,9,10
and x=10 ; now run the code ;

initially i = 1 ,j=10;

first time k = $(i+j) / 2 = 11/2 = 5.5 = 5$ (because of integer type) =i
second time = k = $(i+j) / 2 = 15/2 = 7.5 = 7 = i$
third time = k = $(i+j) / 2 = 17/2 = 8.5 = 8 = i$

fourth time = k = $(i+j) / 2 = 18/2 = 9 = i$

fifth time = k = $(i+j) / 2 = 19/2 = 9.5 = 9 = i$

sixth time = k = $(i+j) / 2 = 19/2 = 9.5 = 9 = i$

seventh time = k = $(i+j) / 2 = 19/2 = 9.5 = 9 = i$

.....

going to infinite loop (run time error) ;

for terminating loop , it should be , i = k + 1 instead of i =k ;and j = k - 1 instead of j = k ;
correct me???

9 votes

-- Mithlesh Upadhyay (5.3k points)

1.12.2 Searching: GATE1996-2.13, ISRO2016-28 [top](#)

<http://gateoverflow.in/2742>



Selected Answer

Expected number of comparisons

= 1 × Probability of first element be x + 2 × Probability of second element be x + + n × Probability of last element be x.

$$= \frac{1}{n} + \frac{2}{n} + \frac{3}{n} + \dots + \frac{n}{n}$$

$$= \frac{\left(\frac{n(n+1)}{2}\right)}{n}$$

$$= \frac{n+1}{2}$$

25 votes

-- Arjun Suresh (287k points)

First consider smaller example...

say list given = {3,1,2} and say you want to search element '2' in sequential way. So first you will visit first element and compare it with '2'. If it is '2' then your search will end at first element with only 1 comparison. But if it is not equal to '2', then you compare it with second element. so second element is '1' so again search was unsuccessful and comparison required was total '2' i.e. b/w '2' & '3' and b/w '2' & '1' and so on.

So if required element is found at first position , no of comparison = 1;
if required element is found at second position , no of comparison = 2 ...and so on.

Now since our list is not sorted so it can be anything e.g. list can be {1,2,3} or {3,2,1} or {2,3,1} etc. So the element we are looking for may be present at any of these three positions with equal chances of 1/3.

Now consider our list containing 'n' elements. So element to be searched can be present at any of these 'n' positions in the list with equal chance(probability) of 1/n.

Total comparison required = No.of comparison if element present in 1st position + No.of comparison if element present in 2nd position ++ No.of comparison if element present in nth position
 $= 1 + 2 + 3 + \dots + n = n(n+1)/2$

Since there are 'n' elements in the list.

So avg. no. of comparison = Total comparison/total no of elements = $[n(n+1)/2] / n = (n+1)/2$.

10 votes

-- Shashank Kumar (3.4k points)

1.12.3 Searching: GATE2008-84 [top](#)

<http://gateoverflow.in/394>



Selected Answer

for Q.84

when it is **option C** the control will continue to iterate as $i = 8$ and $j = 9$;
 again and again i will be assigned k which itself equals 8 as $\frac{8+9}{2}$ being stored in an integer type variable, will evaluate to 8.

For option A, with $x = 9$, k will take the following values:

- 4
- 6
- 7
- $8 - y[8] = 9$, x found

For option D, with $x = 10$, k will take the following values:

- 4, $y[4] = 10$, x found

12 votes

-- Amar Vashishth (27.9k points)

1.12.4 Searching: GATE2008-85 [top](#)

<http://gateoverflow.in/43508>



Selected Answer

Ans should be A

```
if( Y[k] < x) then i = k + 1;
```

if given element that we are searching is greater then searching will be continued upper half of array

otherwise $j = k - 1$;

lower half.

Take few case in consideration i.e.

1. all elements are same
2. increasing order with no repetition
3. increasing order with repetition.

5 votes

-- Manoj Kumar (37k points)

1.12.5 Searching: GATE2017-1-48 [top](#)

<http://gateoverflow.in/118331>



Here since 0's are followed by 1's so we have a sorted sequence and we can apply binary search..

at each stage we compare with $(\text{low} + \text{high})/2$ th element index and if its 1 we check left and if its 0 we check right...

total worst case probes is $\lceil \log_3(10) \rceil = 5$

so answer is 5

11 votes

-- sriv_shubham (2.4k points)

1.12.6 Searching: TIFR2010-B-29 [top](#)

<http://gateoverflow.in/16752>



Option D is the correct answer.

We can simply use clever indexing to binary search the element in the odd positions, and in the even positions separately.

This will take $O(\log n)$ time and $O(1)$ space in the worst case.

A: Sorting using Quicksort will take $O(n^2)$ time.

B: Merging will take $O(n)$ time and $O(n)$ space.

C: Binary search only works on a sorted array.

E: Sequential search will take $O(n)$ time.

8 votes

-- Pragy Agarwal (19.3k points)

1.12.7 Searching: TIFR2012-B-11 [top](#)

<http://gateoverflow.in/25140>



first program wont work if array has elements same..it may go into infinite loop .To make it work it properly we have to do following changes j=k-1 and i=k+1

For second program a[k]==x condition is missing so it is wrong

Third program is also wrong as j!=k-1 and condition a[k]==x is missing

So ans is e

4 votes

-- Pooja Palod (31.2k points)

1.13

Sorting(52) [top](#)

1.13.1 Sorting: CMI2013-A-05 [top](#)

<http://gateoverflow.in/46595>

You have n lists, each consisting of m integers sorted in ascending order. Merging these lists into a single sorted list will take time:

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

cmi2013 | algorithms | sorting

[Answer](#)

1.13.2 Sorting: GATE 2016-1-13 [top](#)

<http://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

[Answer](#)

1.13.3 Sorting: GATE 2016-2-13 [top](#)

<http://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

[Answer](#)

1.13.4 Sorting: GATE1987-1-xviii [top](#)

<http://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

[Answer](#)

1.13.5 Sorting: GATE1988-1iii [top](#)

<http://gateoverflow.in/91338>

Quicksort is _____ efficient than heapsort in the worst case.

gate1988 | algorithms | sorting

[Answer](#)

1.13.6 Sorting: GATE1991_01,vii [top](#)

<http://gateoverflow.in/505>

The minimum number of comparisons required to sort 5 elements is _____

[gate1991](#) [normal](#) [algorithms](#) [sorting](#)[Answer](#)

1.13.7 Sorting: GATE1991_13 [top](#)

<http://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](#)[Answer](#)

1.13.8 Sorting: GATE1992_02,ix [top](#)

<http://gateoverflow.in/559>

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

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

- (a). Heap sort
- (b). Quick sort
- (c). Merge sort
- (d). Radix sort

[gate1992](#) [easy](#) [algorithms](#) [sorting](#)[Answer](#)

1.13.9 Sorting: GATE1992_03,iv [top](#)

<http://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 $[1...n]$ are to be sorted, give an input for which Quicksort takes maximum time.

[gate1992](#) [algorithms](#) [sorting](#) [easy](#)[Answer](#)

1.13.10 Sorting: GATE1994-1.19, ISRO2016-31 [top](#)

<http://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](#)[Answer](#)

1.13.11 Sorting: GATE1995_1.16 [top](#)

<http://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

Answer

1.13.12 Sorting: GATE1995_1.5 [top](#)

<http://gateoverflow.in/2592>

Merge sort uses

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

gate1995 algorithms sorting easy

Answer

1.13.13 Sorting: GATE1995_12 [top](#)

<http://gateoverflow.in/2648>

Consider the following sequence of numbers

92, 37, 52, 12, 11, 25

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

gate1995 algorithms sorting easy

Answer

1.13.14 Sorting: GATE1996_14 [top](#)

<http://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:

- The smallest item in the array is at $A[i][j]$ where $i = \underline{\hspace{2cm}}$ and $j = \underline{\hspace{2cm}}$.
- 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
                      
      A[i][j]:=             
    end
  end
end
```

gate1996 algorithms sorting normal

Answer

1.13.15 Sorting: GATE1996_2.15 [top](#)

<http://gateoverflow.in/2744>

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

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

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](#)

[Answer](#)

1.13.16 Sorting: GATE1998_1.22 [top](#)

<http://gateoverflow.in/1659>

Give the correct matching for the following pairs:

- | | |
|-------------------|--------------------|
| (A) $O(\log n)$ | (P) Selection |
| (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](#)

[Answer](#)

1.13.17 Sorting: GATE1999-1.14, ISRO2015-42 [top](#)

<http://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](#)

[Answer](#)

1.13.18 Sorting: GATE1999-8 [top](#)

<http://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 1st, 2nd and 3rd smallest elements in minimum number of comparisons.

[gate1999](#) [algorithms](#) [sorting](#) [normal](#) [descriptive](#)

Answer**1.13.19 Sorting: GATE1999_1.12** [top](#)<http://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](#)
Answer**1.13.20 Sorting: GATE2000-17** [top](#)<http://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](#)
Answer**1.13.21 Sorting: GATE2001-1.14** [top](#)<http://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](#)
Answer**1.13.22 Sorting: GATE2003-22** [top](#)<http://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](#)
Answer**1.13.23 Sorting: GATE2003-61** [top](#)<http://gateoverflow.in/949>

In a permutation $a_1 \dots a_n$, 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 $1 \dots n$?

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

gate2003 algorithms sorting normal

Answer

1.13.24 Sorting: GATE2003-62 [top](#)

<http://gateoverflow.in/43576>

In a permutation $a_1 \dots a_n$, 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 $1 \dots n$ 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

Answer

1.13.25 Sorting: GATE2005-39 [top](#)

<http://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. $\Omega(n^{3/2})$

gate2005 algorithms sorting normal

Answer

1.13.26 Sorting: GATE2005-IT-59 [top](#)

<http://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

Answer

1.13.27 Sorting: GATE2006-14, ISRO2011-14 [top](#)

<http://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

[Answer](#)

1.13.28 Sorting: GATE2006-52 [top](#)

<http://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

[Answer](#)

1.13.29 Sorting: GATE2007-14 [top](#)

<http://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

[Answer](#)

1.13.30 Sorting: GATE2008-43 [top](#)

<http://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

[Answer](#)

1.13.31 Sorting: GATE2008-IT-43 [top](#)

<http://gateoverflow.in/3353>

If we use Radix Sort to sort n integers in the range $(n^{k/2}, n^k]$, 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

Answer**1.13.32 Sorting: GATE2009-11** [top](#)<http://gateoverflow.in/1303>

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

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

[gate2009](#) [algorithms](#) [sorting](#) [easy](#)**Answer****1.13.33 Sorting: GATE2009-39** [top](#)<http://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](#)**Answer****1.13.34 Sorting: GATE2012_39** [top](#)<http://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](#)**Answer****1.13.35 Sorting: GATE2013_30** [top](#)<http://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](#)**Answer**

1.13.36 Sorting: GATE2013_6 [top](#)<http://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](#)
[Answer](#)**1.13.37 Sorting: GATE2014-1-14** [top](#)<http://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](#)
[Answer](#)**1.13.38 Sorting: GATE2014-2-38** [top](#)<http://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](#)
[Answer](#)**1.13.39 Sorting: GATE2014-3-14** [top](#)<http://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](#)
[Answer](#)**1.13.40 Sorting: GATE2015-1_2** [top](#)<http://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. $T(n) = 2 T(n/2) + cn$
- 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](#)
[Answer](#)**1.13.41 Sorting: GATE2015-2-45** [top](#)<http://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](#)

[Answer](#)

1.13.42 Sorting: GATE2015-3_27 [top](#)

<http://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](#)

[Answer](#)

1.13.43 Sorting: ISI2011-A-2a [top](#)

<http://gateoverflow.in/48028>

Give a strategy to sort four given distinct integers a, b, c, d in increasing order that minimizes the number of pairwise comparisons needed to sort any permutation of a, b, c, d .

[descriptive](#) [isi2011](#) [algorithms](#) [sorting](#)

[Answer](#)

1.13.44 Sorting: TIFR2010-B-23 [top](#)

<http://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. $O(n^2)$ but not better.
- C. $O(n \log n)$
- D. Same as heap sort.
- E. $O(n^{1.5})$ but not better.

[tifr2010](#) [algorithms](#) [time-complexity](#) [sorting](#)

Answer

1.13.45 Sorting: TIFR2010-B-27 [top](#)

<http://gateoverflow.in/19036>

Consider the Insertion Sort procedure given below, which sorts an array L of size n (≥ 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?

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

[tifr2010](#) [algorithms](#) [sorting](#)

Answer

1.13.46 Sorting: TIFR2011-B-21 [top](#)

<http://gateoverflow.in/20324>

Let $S = \{x_1, \dots, x_n\}$ be a set of n numbers. Consider the problem of storing the elements of S in an array $A[1\dots n]$ 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?

- This problem can be solved in $O(\log n)$ time.
- This problem can be solved in $O(n)$ time but not in $O(\log n)$ time.
- This problem can be solved in $O(n \log n)$ time but not in $O(n)$ time.
- This problem can be solved in $O(n^2)$ time but not in $O(n \log n)$ time.
- None of the above.

[tifr2011](#) [algorithms](#) [sorting](#)

Answer

1.13.47 Sorting: TIFR2011-B-31 [top](#)

<http://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?

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

[tifr2011](#) [algorithms](#) [sorting](#)

Answer

1.13.48 Sorting: TIFR2011-B-39 [top](#)

<http://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 ?

- At least n comparisons are necessary in the worst case.
- 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](#)

[Answer](#)

1.13.49 Sorting: TIFR2012-B-13 [top](#)

<http://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 $k^2 n$ comparisons but not fewer.

[tifr2012](#) | [algorithms](#) | [sorting](#)

[Answer](#)

1.13.50 Sorting: TIFR2012-B-14 [top](#)

<http://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](#)

[Answer](#)

1.13.51 Sorting: TIFR2013-B-20 [top](#)

<http://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 P_1 should have the minimum value and the processor P_n 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](#)

[Answer](#)

1.13.52 Sorting: TIFR2017-B-7 [top](#)

<http://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 ?

- 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

Answer

Answers: Sorting

1.13.1 Sorting: CMI2013-A-05 [top](#)

<http://gateoverflow.in/46595>



Since n lists of each size m .

Since each list is sorted in ascending order use directly merge procedure of merge sort algo.

take two list and merge..so one pair will take **2m time**.

so total pairs in first level will be $n/2$. So total cost for one lev1 is **(n/2)*2m=nm**.

In next level cost for one pair is $4m$ and no of pairs will be $n/4$.. so next level cost will be nm .

so like this each level will have cost nm .

No of levels will be when we have one complete list..

$n/2^k = 1..$

$k = \log_2 n$.

So total cost will be **log n * (nm)**

4 votes

-- sonu (2.1k points)

1.13.2 Sorting: GATE 2016-1-13 [top](#)

<http://gateoverflow.in/39660>



Answer is D..

Insertion sort = $\Theta(n^2)$

Merge sort = $\Theta(n \log n)$

Quick sort = $\Theta(n^2)$

Note : here Θ is not average case since question asked worst case so Θ represent worst case only

16 votes

-- Abhilash Panicker (8.7k points)

1.13.3 Sorting: GATE 2016-2-13 [top](#)

<http://gateoverflow.in/39561>



$Q = Q$ can be used instead of Theta

I) Quicksort takes $Q(N^2)$ in case of already sorted input. This is true

II) This is false. If no swap happens then bubble sort can stop in single loop. $O(N)$ is best case. This is false !

iii) Mergesort never takes more than $O(N \log N)$ This is false

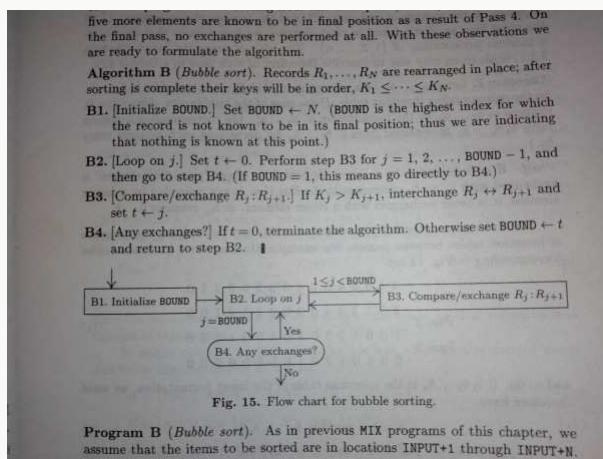
IV) This is true. Insertion sort will finish in $O(N)$ time in case of sorted input.

Answer D) I and IV

Proof Bubble sort has best case $O(N) \Rightarrow$

Ref -> https://en.wikipedia.org/wiki/Bubble_sort, Aduni lecture

Following Photo from Art of Computer Programming , Sorting and Searching (Volume 3)



Now quicksort taking $O(N)$ can happen in some cases but not all cases, so that is why I) should be considered true. Whereas Bubble sort time complexity in best case is always $O(N)$. So D is any time stronger answer than C

19 votes

-- Akash (41.5k points)

1.13.4 Sorting: GATE1987-1-xviii [top](#)

<http://gateoverflow.in/80366>



Selected Answer

ACTUALLY IN BOTH THE CASES IT WILL TAKE $O(n^2)$ TIME($O(n)$ TIME FOR PARTITION ALGORITHM AND $T(n-1)$ TIME FOR SUB PROBLEM.AS n IS THE NUMBER OF INPUTS AND IN THE 2ND CASE INPUTS ARE 5(GREATER THAN 1ST ONE THAT IS 4) I THINK $t_1 < t_2$

5 votes

-- Rohan Ghosh (1.9k points)

1.13.5 Sorting: GATE1988-1-iii [top](#)

<http://gateoverflow.in/91338>



Selected Answer

Answer is LESS

as worst case time for Quicksort is $O(n^2)$ and worst case for heap sort is $O(n \log n)$

4 votes

-- Pavan Kumar Munnam (9.6k points)

1.13.6 Sorting: GATE1991_01,vii [top](#)

<http://gateoverflow.in/505>



Selected Answer

Answer: 7

Minimum number of comparisons = $\lceil \log(n!) \rceil = \lceil \log(5!) \rceil = \lceil \log(120) \rceil = 7$.Ref: http://en.wikipedia.org/wiki/Comparison_sort#Number_of_comparisons_required_to_sort_a_list

Upvote 8 votes

-- Rajarshi Sarkar (34.4k points)

1.13.7 Sorting: GATE1991_13 [top](#)<http://gateoverflow.in/540>use a hash map with chaining to sort all the numbers in $\mathcal{O}(n)$ time and constant space.

Upvote 2 votes

-- Amar Vashishth (27.9k points)

1.13.8 Sorting: GATE1992_02,ix [top](#)<http://gateoverflow.in/559>

Answer: D

Radix sort complexity is $O(wn)$ for n keys which are integers of word size w .

Upvote 5 votes

-- Rajarshi Sarkar (34.4k points)

1.13.9 Sorting: GATE1992_03,iv [top](#)<http://gateoverflow.in/581>

Selected Answer

The algorithm will take maximum time when:

- 1) The array is already sorted in same order.
- 2) The array is already sorted in reverse order.
- 3) All elements are same in the array.

Upvote 14 votes

-- Rajarshi Sarkar (34.4k points)

1.13.10 Sorting: GATE1994-1.19, ISRO2016-31 [top](#)<http://gateoverflow.in/2462>

Selected Answer

C. it is one of the efficient algorithms in Divide and Conquer strategy.

Upvote 13 votes

-- Gate Keeda (18.8k points)

1.13.11 Sorting: GATE1995_1.16 [top](#)<http://gateoverflow.in/2603>

Selected Answer

It is C.

The number of moves are however always $m+n$ so that we can term it as $\theta(m+n)$. But the number of comparisons vary as per the input. In the best case the comparisons are $\min(m,n)$ and in worst case they are $m+n-1$.

Upvote 14 votes

-- Gate Keeda (18.8k points)

1.13.12 Sorting: GATE1995_1.5 [top](#)<http://gateoverflow.in/2592>

Selected Answer

It is A.

One of the best examples of Divide and conquer strategy.

11 votes

-- Gate Keeda (18.8k points)

1.13.13 Sorting: GATE1995_12 [top](#)<http://gateoverflow.in/2648>

Selected Answer

1st Pass: 37 52 12 11 25 922nd Pass: 37 12 11 25 52 923rd Pass: 12 11 25 37 52 924th Pass: 11 12 25 37 52 925th Pass: 11 12 25 37 52 92

11 votes

-- Gate Keeda (18.8k points)

1.13.14 Sorting: GATE1996_14 [top](#)<http://gateoverflow.in/2766>

Selected Answer

(a). i=1 and j=1 (smallest item in the array)

(b). here in while loop ($x > A[i+1][j]$ or $x > A[i][j+1]$)

```
A[i][j]:=A[i+1][j]; i:=i+1;
```

else begin

```
A[i][j]:=A[i][j+1]; j=j+1;
```

end

```
A[i][j]:=x;
```

7 votes

-- munna kumar (369 points)

1.13.15 Sorting: GATE1996_2.15 [top](#)<http://gateoverflow.in/2744>

Selected Answer

C.

both are the worst cases of quick sort. (assuming pivot is either first or last element)

i) is sorted in ascending order.

ii) is sorted in descending order.

11 votes

-- Gate Keeda (18.8k points)

1.13.16 Sorting: GATE1998_1.22 [top](#)<http://gateoverflow.in/1659>

Selected Answer

selection sort $O(n^2)$ merge sort $O(n \log n)$ binary search ($\log n$)insertion sort $O(n)$ note if you use $O(n^2)$ here you will not be left with any choice to fill selection sort

8 votes

-- Bhagirathi Nayak (13.1k points)

1.13.17 Sorting: GATE1999-1.14, ISRO2015-42 [top](#)

<http://gateoverflow.in/146>

Selected Answer

```

20 47 15 8 9 4 40 30 12 17
 \ / \ / \ / \ / \ / \ / after 1st pass
20 47 8 15 4 9 30 40 12 17
 \ / \ / \ / \ / \ / \ / after 2nd pass
8,15,20,47   4,9, 30,40   12,17

```

Ans. **B** 30 votes

-- Vikrant Singh (13.4k points)

1.13.18 Sorting: GATE1999-8 [top](#)

<http://gateoverflow.in/150>

If all elements are distinct we have to check 1st row and 1st column. So complexity will be $O(n^2)$

Here minimum no of comparison could be 1 , because $a[0][0]$ will be minimum always, now we have to check between $a[0][1]$ and $a[1][0]$

	0	1	2
0	1	2	7
1	3	4	8
2	5	6	9

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

if all elements are not distinct we have to search all elements of the array

	0	1	2
0	1	1	1
1	1	1	8
2	1	6	9

So, minimum comparison could be $O(1)$

 2 votes

-- srestha (55.2k points)

1.13.19 Sorting: GATE1999_1.12 [top](#)

<http://gateoverflow.in/146>

Selected Answer

(b) If it maintains the relative order of occurrence of non-distinct elements.

(from definition of stable sorting)

 7 votes

-- Arjun Suresh (287k points)

1.13.20 Sorting: GATE2000-17 [top](#)

<http://gateoverflow.in/688>



Selected Answer

Since swaps are needed to be of adjacent elements only, the algorithm is actually Bubble sort.

In bubble sort, all smaller elements to right of an element are required to be swapped. So if have ordering

[2,2,2,1,1,1,1,0,0,0,0] , then we need total 47 swaps, and this will be the worst case.

so it answers actually both parts.

12 votes

-- Happy Mittal (10.7k points)

1.13.21 Sorting: GATE2001-1.14 [top](#)

<http://gateoverflow.in/707>



Selected Answer

In worst case, we may pick pivot elements in the increasing order (input also given in sorted order) which will result in running time of $O(n^2)$

Both the deterministic and randomized quicksort algorithms have the same best-case running times of $O(n \lg n)$ and the same worst-case running times of $O(n^2)$. The difference is that with the deterministic algorithm, a particular input can elicit that worst-case behavior. With the randomized algorithm, however, no input can always elicit the worst-case behavior.

The reason it matters is that, depending on how partitioning is implemented, an input that is already sorted--or almost sorted--can elicit the worst-case behavior in deterministic quicksort.

source: Thomas Cormen

Ans. C

21 votes

-- Vikrant Singh (13.4k points)

1.13.22 Sorting: GATE2003-22 [top](#)

<http://gateoverflow.in/912>



Selected Answer

In insertion sort, with linear search, it takes

(worst case) n comparisons for searching the right position, and n swaps to make room to place the element.

Hence for n elements, a total of $n \times (n + n)$; n for search and n for swaps.

$$= \Theta(2n^2) = \Theta(n^2)$$

If we replace it with binary search, it takes

(worst case)

$\log n$ comparisons for searching the right position, and n swaps to make room to place the element.

Hence for n elements, a total of $n \times (\log n + n)$;

n for search and

n for swaps.

$$= \Theta(n \times \log n + n^2) = \Theta(n^2)$$

Hence **answer is A**

20 votes

-- ryan sequiera (2.9k points)

A. Complexity remains same. $\Theta(n^2)$

To place the element x in the correct position first we are finding its correct position in the sorted array using binary search but we have to make the space for it by shifting all elements to the right, which in worst case may be equal to the size of the sorted array.

12 votes

-- Vikrant Singh (13.4k points)

1.13.23 Sorting: GATE2003-61 [top](#)

<http://gateoverflow.in/949>



Selected Answer

They are asking the average number of inversion. basically what i learned about averages from dbms indexing is.

average apart from the standard definition can be calculated as (best case + worst case)/2

and inversion is like 9,5.

so best case will be sorted array - 1,2,3,4,5 no inversion . = zero

worst case = 9,5,4,3,2,1 . here total number of inversion will be $n(n-1)/2$ as . 9 can be paired with any 5 elements (5,4,3,2,1) will form a inversion pair. similarly 5 with 4.elements .

so we can say if we have n elements then. it will be $(n-1)+(n-2)+(n-3)...+2+1$ which is the sum of first $n-1$ natural numbers. so it is $n(n-1)/2$

so expected average number of inversion = $(n(n-1)/2 + zero (best case)) /2= n(n-1)/4$

so option b.

second question.

we all know that insertion sort has complexity due to swapping and movements, if we have n inversion pair then the movements and comparision will be restricted to n only . like if inversion is 1 , then array must be sorted and only the inversion should exist at the end, like 1,2,3,5,4. otherwise more than one inversion pair will form. so to sort this. for two it will be 1,2,3,7,5,4. so to sort this type of array using insertion sort atmost N swaps wiill be required, so d

14 votes

-- No Need (13.7k points)

1.13.24 Sorting: GATE2003-62 [top](#)

<http://gateoverflow.in/43576>



Selected Answer

ANSWER: D. $\Theta(n)$

REASON:

Count of number of times the inner loop of insertion sort executes is actually equal to number of inversions in input permutation a_1, a_2, \dots, a_n . Since for each value of $i = 1..n$, j take the value $1..i-1$, which means for every $j < i$ it checks if $a[j] > a[i]$.

In any given permutation, maximum number of inversions possible is $n(n-1)/2$ which is $O(n^2)$. It is the case where the array is sorted in reverse order. Therefore, to resolve all inversions i.e., worst case time complexity of insertion sort is $\Theta(n^2)$.

However, as per the question the number of inversion in input array is restricted to n . The worst case time complexity of insertion sort reduces to $\Theta(n)$.

[INSERTION SORT ALGORITHM](#) (for reference)

13 votes

-- Prateek Dwivedi (2.5k points)

1.13.25 Sorting: GATE2005-39 [top](#)

<http://gateoverflow.in/784>



Selected Answer

Since we have $\log n$ lists we can make a min-heap of $\log n$ elements by taking the first element from each of the $\log n$ sorted lists. Now, we start deleting the min-element from the heap and put the next element from the sorted list from which that element was added to the heap. (This identity can be done by making a structure of two values, one for the number and one for identifying the origin sorted list of that number and storing this structure in the heap). In this way each delete and the corresponding insert will take $O(\log \log n)$ time as delete in heap of size n is $O(\log n)$ and inserting an element on a heap of size n is also $O(\log n)$. (here, heap size is $\log n$). Now, we have a total of $\log n \times \frac{n}{\log n} = n$ elements. So, total time will be $O(n \log \log n)$.

14 votes

-- gatecse (13.1k points)

1.13.26 Sorting: GATE2005-IT-59 [top](#)



Selected Answer

 $a[i] \geq b[i]$

Since both a and b are sorted in the beginning, there are i elements smaller than or equal to $a[i]$ (i starts from 0), and similarly i elements smaller than or equal to $b[i]$. So, $a[i] \geq b[i]$ means there are $2i$ elements smaller than or equal to $a[i]$, and hence in the merged array $a[i]$ will come after these $2i$ elements (its index will be $> 2i$). So, $c[2i] \leq a[i]$ (equality takes care of the "equal to" case which comes when array contains repeated elements).

Similarly, $a[i] \geq b[i]$ says for b that, there are not more than $2i$ elements smaller than $b[i]$ in the sorted array (i elements from b , and maximum another i elements from a). So, $b[i] \leq c[2i]$

So, II and III are correct -> option (C)

14 votes

-- Arjun Suresh (287k points)

1.13.27 Sorting: GATE2006-14, ISRO2011-14 [top](#)



Selected Answer

(C) Selection sort

14 votes

-- Keith Kr (6.2k points)

1.13.28 Sorting: GATE2006-52 [top](#)



Selected Answer

as we choose the pivot a median element ...so every time we are going to have good splits guaranteed so the best case $O(n \log n)$

14 votes

-- Bhagirathi Nayak (13.1k points)

1.13.29 Sorting: GATE2007-14 [top](#)



Selected Answer

A.

Irrespective of the input, Merge sort always have a time complexity of $\Theta(n \log n)$.

14 votes

-- Gate Keeda (18.8k points)

1.13.30 Sorting: GATE2008-43 [top](#)

<http://gateoverflow.in/455>



Selected Answer

$$T(n) \leq T(n/5) + T(4n/5) + n$$

One part contains $n/5$ elements
and the other part contains $4n/5$ elements
 $+n$ is common to all options, so we need not to worry about it.

Hence, answer = **option B**

10 votes

-- Amar Vashishth (27.9k points)

1.13.31 Sorting: GATE2008-IT-43 [top](#)



Selected Answer

Answer: C

The complexity of Radix Sort is $O(wn)$, for n keys which are integers of word size w .

Here, $w = \log_2(n^k) = k \times \log_2(n)$

So, the complexity is $O(wn) = O(k \times \log_2(n) \times n)$, which leads to option C.

16 votes

-- Rajarshi Sarkar (34.4k points)

1.13.32 Sorting: GATE2009-11 [top](#)



Selected Answer

The answer is A.

we have 1 swap in each loop and hence n swaps at max for 1 to n . Therefore the worst case number of swaps is $\Theta(n)$

12 votes

-- Gate Keeda (18.8k points)

1.13.33 Sorting: GATE2009-39 [top](#)



Selected Answer

B.

$T(n) = O(n)$ pivot selection time + $T(n/4 - 1) + T(3n/4)$

which'll give $\Theta(n \log n)$.

Pivot selection complexity is given in questions. Pivot being the $(n/4)$ th smallest element, once it is found, we have two sub arrays- one of size $(n/4 - 1)$ and other of size $(3n/4)$ and for both of these we solve recursively.

18 votes

-- Gate Keeda (18.8k points)

1.13.34 Sorting: GATE2012_39 [top](#)



Selected Answer

i thought like this:

you are given the first character of each n strings to sort it will take $O(n \log n)$ time..in the worst case we may have to do the above process 2 times,3 times,.....,n times so $n * O(n \log n) = O(n^2 \log n)$ please correct me if my approach is wrong...

18 votes

-- Bhagirathi Nayak (13.1k points)

1.13.35 Sorting: GATE2013_30 top

<http://gateoverflow.in/1541>



Selected Answer

To sort k elements in a heap, complexity is $\Theta(k \log k)$. Lets assume there are $\frac{\log n}{\log \log n}$ elements in the heap.

$$\text{Complexity} = \Theta\left(\frac{\log n}{\log \log n} \log\left(\frac{\log n}{\log \log n}\right)\right)$$

$$= \Theta\left(\frac{\log n}{\log \log n} (\log \log n - \log \log \log n)\right)$$

$$= \Theta\left(\log n - \frac{\log n \log \log n}{\log \log n}\right)$$

$$= \Theta(\log n) \text{ (as shown below)}$$

So, (c) is the answer.

$$\log \log n > \log \log \log n$$

$$\implies \frac{\log \log \log n}{\log \log n} < 1$$

$$\implies \frac{\log n \log \log \log n}{\log \log n} < \log n$$

$$\implies \Theta\left(\log n - \frac{\log n \log \log \log n}{\log \log n}\right) = \Theta(\log n)$$

49 votes

-- Arjun Suresh (287k points)

answer = option C

Heap sort sorts k elements in time $\Theta(k \log k)$. Let's try this with the given choices:

1. $k = \Theta(1)$
 $\Theta(k \log k) = \Theta(1 \log 1) = \Theta(1)$, wrong (too fast).
2. $k = \Theta(\sqrt{\log n})$
 $\Theta(k \log k) = \Theta(\sqrt{\log n} \log \sqrt{\log n})$
 $= \Theta(\sqrt{\log n \log \log n})$, wrong (too fast).
3. $k = \Theta\left(\frac{\log n}{\log \log n}\right)$
 $\Theta(k \log k) = \Theta\left(\frac{\log n \log \log n}{\log \log n}\right)$
 $= \Theta\left(\frac{\log n}{\log \log n} (\log \log n - \log \log \log n)\right)$
 $= \Theta\left(\frac{\log n}{\log \log n} \log \log n\right)$
 $= \Theta(\log n)$, right!
4. $k = \Theta(\log n)$
 $\Theta(k \log k) = \Theta(\log n \log \log n)$, wrong (too slow).

11 votes

-- Amar Vashishth (27.9k points)

1.13.36 Sorting: GATE2013_6 top

<http://gateoverflow.in/1415>



Selected Answer

In selection max you can do is n swaps..selecting the smallest element from all the elements and replacing it correct position so $O(n)$

9 votes

-- Bhagirathi Nayak (13.1k points)

1.13.37 Sorting: GATE2014-1-14 top

<http://gateoverflow.in/1780>



Selected Answer

it would be $t_1 > t_2$, because the first case is the worst case of quicksort i.e. minimum number is chosen as pivot. Hence in the worst case the comparisons are high.

The splitting occurs as

- [1] [2345]
- [2] [345]
- [3] [45]
- [4] [5]

and

- [123] [45]
- [1] [23] [4][5]
- [2] [3]

Number of recursive calls remain the same, but in second case the number of elements passed for the recursive call is less and hence the number of comparisons also less.

13 votes

-- Parul Agarwal (873 points)

1.13.38 Sorting: GATE2014-2-38 [top](#)

<http://gateoverflow.in/1997>



Selected Answer

The optimal algorithm always chooses the smallest sequences for merging.

- 20 24 -44, 43 comparisons
- 30 35 -65, 64 comparisons
- 44 50 -94, 93 comparisons
- 65 94 -159, 158 comparisons

so, totally $43 + 64 + 93 + 158 = 358$ comparisons.

36 votes

-- Arjun Suresh (287k points)

1.13.39 Sorting: GATE2014-3-14 [top](#)

<http://gateoverflow.in/2048>



Selected Answer

(A) $O(n^2)$ is the answer. When we choose the first element as the pivot, the worst case of quick sort comes if the input is sorted- either in ascending or descending order. Now, when we choose the middle element as pivot, sorted input no longer gives worst case behaviour. But, there will be some permutation of the input numbers which will be giving the same worst case behaviour. For example,

1 2 3 4 5 6 7

This array gives worst case behaviour for quick sort when the first element is pivot.

6 4 2 1 3 5 7

This array gives the worst case behaviour of $O(n^2)$ if we take middle element as the pivot- each split will be 1 element on one side and $n-1$ elements on other side. Similarly, for any input, we can have a permutation where the behaviour is like this. So, whichever element we take as pivot it gives worst case complexity of $O(n^2)$ as long as pivot is from a fixed position (not random position as in randomized quick sort).

25 votes

-- Arjun Suresh (287k points)

1.13.40 Sorting: GATE2015-1_2 [top](#)

<http://gateoverflow.in/8017>

 Selected Answer

B.

Worst case for quick sort happens when 1 element is on one list and $n-1$ elements on another list.

17 votes -- Arjun Suresh (287k points)

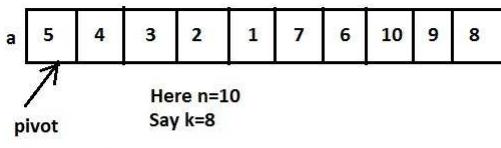
1.13.41 Sorting: GATE2015-2-45 top

<http://gateoverflow.in/8243>

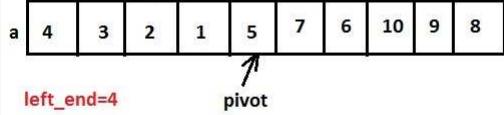
First of all here The return value is the number of elements less than the pivot

Pivot is just to minimize searching

So, now we are assuming our array has 10 elements , $N=10$, $k=8$

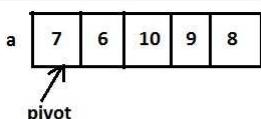


STEP 1: after partition()

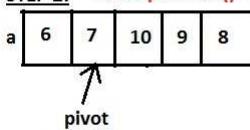


So, $(a+left_end+1, n-left_end-1, k-left_end-1)$

$(a+5, 5, 3)$



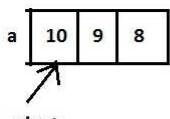
STEP 2: after partition()



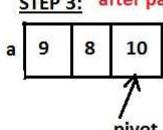
left_end=1

left_end+1<k

So, $(a+2, 3, 1)$



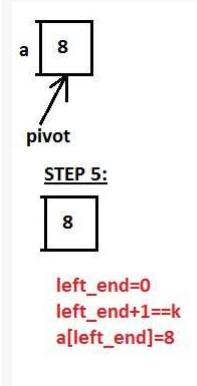
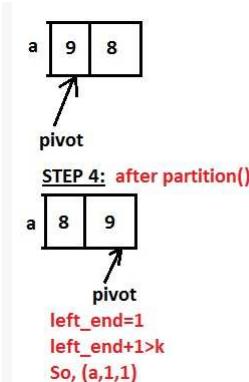
STEP 3: after partition()



left_end=2

left_end+1>k

So, $(a, 2, 1)$



So, in STEP 1 and STEP 2 'else' condition satisfying, and STEP3 and STEP 4 'if' condition satisfying

Here partition is calling and returning left_end value

Ans will be (A)

7 votes

-- srestha (55.2k points)

1.13.42 Sorting: GATE2015-3_27 [top](#)

<http://gateoverflow.in/8480>



Selected Answer

The worst case time complexity of Mergesort is $k \times n \log n$ for an input of size n .

For an input of size 64, the algorithm takes 30s. Therefore,

$$\begin{aligned} k \times 64 \log_2 64 &= 30s \\ k \times 384 &= 30s \\ \implies k &= 0.078125s \end{aligned}$$

Let the size of the problem that can be solved in 6 minutes be x . Then,

$$k \times x \log_2 x = 360s$$

From this, we get:

$$\begin{aligned} x \log_2 x &= \frac{360s}{0.078125s} \\ \implies x &= 512 \end{aligned}$$

27 votes

-- Pragy Agarwal (19.3k points)

1.13.43 Sorting: ISI2011-A-2a [top](#)

<http://gateoverflow.in/48028>

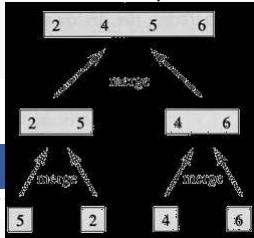
it will be merge sort

a,b,c,d

1+1 comparisons for lowest step

+3 comparisons for upper one

so total 5 comparisons



↳ 2 votes

-- Tanushree (385 points)

1.13.44 Sorting: TIFR2010-B-23 [top](#)

<http://gateoverflow.in/18623>

Answer should be (D) same as Heap Sort, the details what they have mentioned in the question points out the heap-sort procedure, every time we remove max element and then find next max element to be removed and so on..

↳ 3 votes

-- Vijay Thakur (14.7k points)

1.13.45 Sorting: TIFR2010-B-27 [top](#)

<http://gateoverflow.in/19036>



Selected Answer

In worst case Insertion sort will have $N(N-1)/2$ comparisons i.e. when input is sorted in descending order.

50 40 30 20 10

pass 1 50 40 30 20 10.....n 0 compare

pass 2 40 50 30 20 10.....n 1 compare

.

.

.

pass n n10 20 30 40 50 n-1 compare

$1+2+3+...+n-1 = N(N-1)/2$ comparisons

↳ 3 votes

-- Umang Raman (14.5k points)

1.13.46 Sorting: TIFR2011-B-21 [top](#)

<http://gateoverflow.in/20324>



Selected Answer

store the elements in an array and then call build_heap(A). the build_heap takes $O(n)$ time.

so, option 'b' is correct.

but, if we try building heap by inserting each element one by one, the total complexity will be then $O(n \log n)$. cause insertion takes $O(\log n)$ and inserting 'n' elements will take $O(n \log n)$.

↳ 10 votes

-- Sujit Kumar Muduli (209 points)

1.13.47 Sorting: TIFR2011-B-31 [top](#)

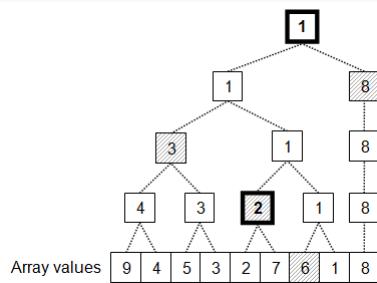
<http://gateoverflow.in/20617>



Option c) $n + k - 2$

Here is a nice explanation of the algorithm: <http://www.codinghelmet.com/?path=exercises/two-smallest>

it is solution to the problem known for ages, and it has to do with tennis tournaments. The question was, knowing the outcome of the tennis tournament, how can we tell which player was the second best? The defeated finalist is a good candidate, but there are other players that were defeated directly by the tournament winner and any of them could also be a good candidate for the second best. So the solution to the problem is quite simple: Once the tournament finishes, pick up the $\log N$ competitors that were beaten by the tournament winner and hold a mini-tournament to find which one is the best among them. If we imagine that better players correspond with smaller numbers, the algorithm now goes like this. Hold the tournament to find the smallest number (requires $N-1$ comparisons). During this step, for each number construct the list of numbers it was smaller than. Finally, pick the list of numbers associated with the smallest number and find their minimum in $\log N-1$ steps. This algorithm requires $N+\log N-2$ comparisons to complete, but unfortunately it requires additional space proportional to N (each element except the winner will ultimately be added to someone's list); it also requires more time per step because of the relatively complex enlisting logic involved in each comparison. When this optimized algorithm is applied to example array, we get the following figure.



Tournament held among numbers promotes value 1 as the smallest number. That operation, performed on an array with nine numbers, requires exactly eight comparisons. While promoting the smallest number, this operation has also flagged four numbers that were removed from competition by direct comparison with the future winner: 6, 2, 3 and 8 in that order. Another sequence of three comparisons is required to promote number 2 as the second-smallest number in the array. This totals 11 comparisons, while naive algorithm requires 17 comparisons to come up with the same result.

All in all, this algorithm that minimizes number of comparisons looks to be good only for real tournaments, while number cracking algorithms should keep with the simple logic explained above. Implementation of simple algorithm may look like this:

```
a = array containing n elements
min1 = a[0] - candidate for the smallest value
min2 = a[1] - candidate for the second smallest value
if min2 < min1
    min1 = a[1]
    min2 = a[0]
for i = 2 to n - 1
    if a[i] < min1
        min2 = min1
        min1 = a[i]
    else if a[i] < min2
        min2 = a[i]
```

by Zoran Horvat [@zoranh75](#)

14 votes

-- Pragy Agarwal (19.3k points)

1.13.48 Sorting: TIFR2011-B-39 [top](#)



d)
 $O(\log n)$ comparisons suffice.

Since it is possible that $m \gg n$, we need to restrict ourselves to the first $O(n)$ elements to perform the binary search.

We start with the first element (index $i = 1$), and check if it is equal to 0. If not, we double the value of i , and check

again. We repeat this process until we hit a 0.

```
i = 1;
while(arr[i] != 0)
    i *= 2;
```

Once we hit a 0, the largest possible value (worst case) of i can be $2n - 2$. This will happen if $n = 2^k + 1$ for some k . Then, our 2nd last value of i will be 2^k , and then we get 2^{k+1} , which is equal to $2n - 2$.

Now that we've hit a 0, and the array contains positive numbers in decreasing order, if x is present in L , it must be in the first i elements.

We can binary search the first i elements in $O(\log i)$ comparisons.

Since the largest possible value of $i = 2n - 2$, our algorithm takes $O(\log(2n - 2)) = O(\log n)$ comparisons.

8 votes

-- Pragy Agarwal (19.3k points)

1.13.49 Sorting: TIFR2012-B-13 [top](#)

<http://gateoverflow.in/25207>



Selected Answer

Let Array element be {4,3,2,1,7,5,6,9,10,8} and K be 3 here no element is more than 3 distance away from its final position

So if we take

arr(1 to 6) and sort then surely first three element will be sorted in its final position {12345769108} $O(6\log 6)$
then sort arr(3 to 9) then 3 to 6 will be sorted {12345679108} $O(6\log 6)$
then at last arr(6 to 9) less than $O(6\log 6)$ {12345678910}

in general

Sort arr(0 to $2k$)

Now we know that arr[0 to k] are in their final sorted positions
and arr(k to $2k$) may be not sorted.

Sort arr(k to $3k$)

Now we know that arr[k to $2k$] are in their final sorted positions
and arr($2k$ to $3k$) may be not sorted.

.

.

.

sort till arr($ik..N$)

in final sorting there will be less than $2k$ element.

in each step it will take $O(2k\log 2k)$
and there will $\frac{n}{k}$ steps so $O(n\log k)$
option D.

6 votes

-- Umang Raman (14.5k points)

1.13.50 Sorting: TIFR2012-B-14 [top](#)

<http://gateoverflow.in/25209>



Selected Answer

Algorithm is choosing median = $n/2$ smallest element as pivot.

Hence, the array is divided as:

$(\frac{n}{2}-1)$ elements	Median at $\frac{n}{2}$ th location	$(n-\frac{n}{2})$ elements
-------------------------------	---	-------------------------------

Therefore quick sort recurrence relation is given by:

$$\begin{aligned} T(n) &= T\left(\frac{n}{2}-1\right) + T\left(n-\frac{n}{2}\right) + \Theta(n) \\ &= \Theta(n \log n) \end{aligned}$$

Hence, Option B is the correct answer.

9 votes

-- Umang Raman (14.5k points)

1.13.51 Sorting: TIFR2013-B-20 [top](#)

<http://gateoverflow.in/25878>

OPTION C is correct.

5 votes

-- venky.victory35 (643 points)

1.13.52 Sorting: TIFR2017-B-7 [top](#)

<http://gateoverflow.in/95699>

A pairwise swap will make the sorted array unsorted. Hence, the option (b) is correct.

For eg - if an array is 1 2 3 4 5 6 7 8

The array will become after a pair wise swap to 2 1 4 3 6 5 8 7. For all i between 2 and n-1, $a[i]$ is either lower, or either greater than their adjacent elements.

Since, each element is being swapped exactly once. The operation has $O(n)$ time complexity.

2 votes

-- tarun_svbk (1k points)

1.14

Spanning Tree(31) [top](#)

<http://gateoverflow.in/39673>

1.14.1 Spanning Tree: GATE 2016-1-14 [top](#)

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

Answer

1.14.2 Spanning Tree: GATE 2016-1-39 [top](#)

<http://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](#)
Answer

1.14.3 Spanning Tree: GATE 2016-1-40 [top](#)

<http://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 (MST 's) 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 .
- A. I only.
 B. II only.
 C. Both I and II.
 D. Neither I nor II.

[gate2016-1](#) | [algorithms](#) | [spanning-tree](#) | [normal](#)
Answer

1.14.4 Spanning Tree: GATE1991_03,vi [top](#)

<http://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](#)
Answer

1.14.5 Spanning Tree: GATE1992_01,ix [top](#)

<http://gateoverflow.in/549>

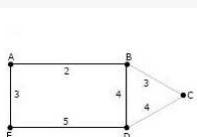
Complexity of Kruskal's algorithm for finding the minimum spanning tree of an undirected graph containing n vertices and m edges if the edges are sorted is _____

[gate1992](#) | [spanning-tree](#) | [algorithms](#) | [time-complexity](#) | [easy](#)
Answer

1.14.6 Spanning Tree: GATE1995_22 [top](#)

<http://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](#)
Answer

1.14.7 Spanning Tree: GATE1996_16 [top](#)

<http://gateoverflow.in/2768>

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

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

gate1996 algorithms graph-algorithms spanning-tree normal

[Answer](#)

1.14.8 Spanning Tree: GATE1997_9 [top](#)

<http://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 $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

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

[Answer](#)

1.14.9 Spanning Tree: GATE2000-2.18 [top](#)

<http://gateoverflow.in/665>

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

[Answer](#)

1.14.10 Spanning Tree: GATE2001-15 [top](#)

<http://gateoverflow.in/756>

Consider a weighted undirected graph with vertex set $V = \{n_1, n_2, n_3, n_4, n_5, n_6\}$ and edge set $E = \{(n_1, n_2, 2), (n_1, n_3, 8), (n_1, n_6, 3), (n_2, n_4, 4), (n_2, n_5, 12), (n_3, n_4, 7), (n_4, n_5, 9), (n_4, n_6, 4)\}$. The third value in each tuple represents the weight of the edge specified in the tuple.

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

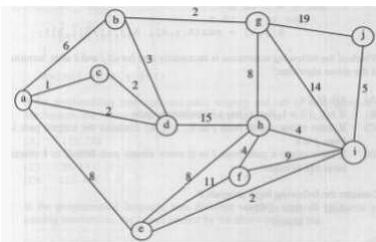
gate2001 algorithms spanning-tree normal descriptive

[Answer](#)

1.14.11 Spanning Tree: GATE2003-68 [top](#)

<http://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

[Answer](#)

1.14.12 Spanning Tree: GATE2005-6 [top](#)

<http://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

[Answer](#)

1.14.13 Spanning Tree: GATE2005-IT-52 [top](#)

<http://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.
D. All edges in G have the same weight.

gate2005-it algorithms spanning-tree normal

[Answer](#)

1.14.14 Spanning Tree: GATE2006-11 [top](#)

<http://gateoverflow.in/890>

Consider a weighted complete graph G on the vertex set $\{v_1, v_2, \dots, v_n\}$ 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. $2n-3$
C. $\binom{n}{2}$
D. n^2

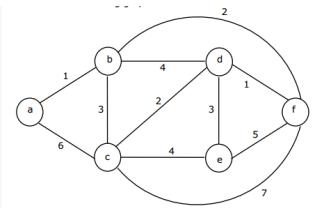
gate2006 algorithms spanning-tree normal

[Answer](#)

1.14.15 Spanning Tree: GATE2006-47 [top](#)

<http://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](#)

[Answer](#)

1.14.16 Spanning Tree: GATE2007-49 [top](#)

<http://gateoverflow.in/124>

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

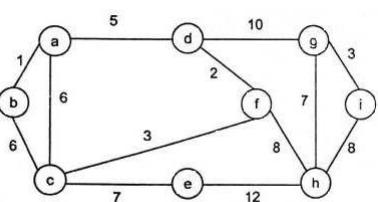
[gate2007](#) [algorithms](#) [spanning-tree](#) [normal](#)

[Answer](#)

1.14.17 Spanning Tree: GATE2008-IT-45 [top](#)

<http://gateoverflow.in/335>

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

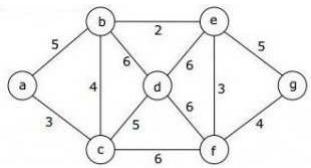
[gate2008-it](#) [algorithms](#) [graph-algorithms](#) [spanning-tree](#) [normal](#)

[Answer](#)

1.14.18 Spanning Tree: GATE2009-38 [top](#)

<http://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](#)

Answer

1.14.19 Spanning Tree: GATE2010-50 [top](#)

<http://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](#)

Answer

1.14.20 Spanning Tree: GATE2010-51 [top](#)

<http://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

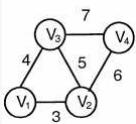
gate2010 | normal | algorithms | spanning-tree

[Answer](#)

1.14.21 Spanning Tree: GATE2011-54 [top](#)

<http://gateoverflow.in/2162>

An undirected graph $G(V, E)$ contains n ($n > 2$) nodes named v_1, v_2, \dots, 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.



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. $2n + 1$

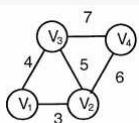
gate2011 | algorithms | graph-algorithms | spanning-tree | normal

[Answer](#)

1.14.22 Spanning Tree: GATE2011-55 [top](#)

<http://gateoverflow.in/43325>

An undirected graph $G(V, E)$ contains n ($n > 2$) nodes named v_1, v_2, \dots, 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

[Answer](#)

1.14.23 Spanning Tree: GATE2012_29 [top](#)

<http://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

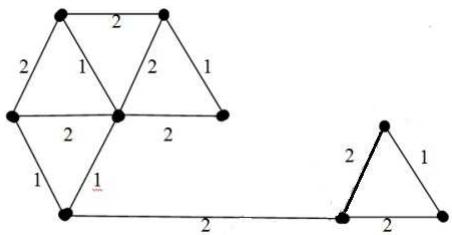
gate2012 | algorithms | spanning-tree | normal | marks-to-all

Answer

1.14.24 Spanning Tree: GATE2014-2-52 [top](#)

<http://gateoverflow.in/2019>

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



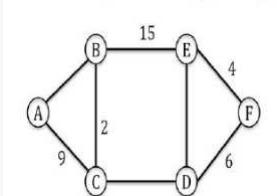
gate2014-2 | algorithms | spanning-tree | numerical-answers | normal

Answer

1.14.25 Spanning Tree: GATE2015-1_43 [top](#)

<http://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 _____.



gate2015-1 | algorithms | spanning-tree | normal | numerical-answers

Answer

1.14.26 Spanning Tree: GATE2015-3_40 [top](#)

<http://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

Answer

1.14.27 Spanning Tree: TIFR2011-B-35 [top](#)

<http://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, \dots, e_m be an ordering of the edges in decreasing order of weight. Which of the following statements is FALSE?

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

tifr2011 | algorithms | graph-algorithms | spanning-tree

Answer

1.14.28 Spanning Tree: TIFR2013-B-17 [top](#)<http://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

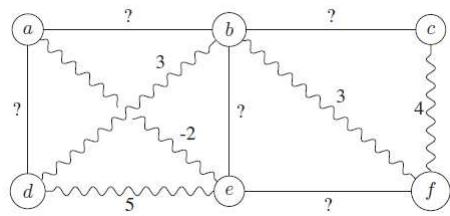
- a. 1
- b. n
- c. equal to number of edges in the graph.
- d. equal to maximum weight of an edge of the graph.
- e. n^{n-2}

[tifr2013](#) [spanning-tree](#)

Answer

1.14.29 Spanning Tree: TIFR2014-B-4 [top](#)<http://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](#)

Answer

1.14.30 Spanning Tree: TIFR2014-B-5 [top](#)<http://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 $w : E \rightarrow \mathbb{R}$ on its edge set. Let $w(e_1) < w(e_2) < \dots < w(e_m)$, where $E = \{e_1, e_2, \dots, e_m\}$. Suppose T is a minimum spanning tree of G . Which of the following statements is FALSE?

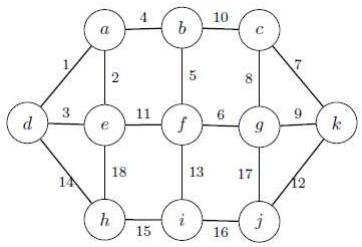
- a. The tree T has to contain the edge e_1 .
- b. The tree T has to contain the edge e_2 .
- c. The minimum weight edge incident on each vertex has to be present in T .
- d. T is the unique minimum spanning tree in G .
- e. If we replace each edge weight $w_i = w(e_i)$ by its square w_i^2 , then T must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

[tifr2014](#) [algorithms](#) [spanning-tree](#)

Answer

1.14.31 Spanning Tree: TIFR2015-B-2 [top](#)<http://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)

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

tifr2015 | spanning-tree

Answer

Answers: Spanning Tree

1.14.1 Spanning Tree: GATE 2016-1-14 [top](#)

<http://gateoverflow.in/39673>



Selected Answer

Statement P is true.

For statement Q consider a simple graph with 3 nodes.

	A	B	C
A	0	1	100
B	1	0	2
C	100	2	0

Shortest path from A to C is A-B-C = 1 + 2 = 3

Now if the value of each edge is increased by 100,

	A	B	C
A	0	101	200
B	101	0	102
C	200	102	0

The shortest path from A to C is A-C = 200, (A-B-C = 101 + 102 = 203)

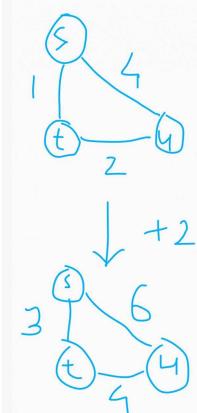
Hence option **A is correct.**

31 votes

-- ryan sequeira (2.9k points)

P is True:-> Bcz distinct edge weights and all the edge cost are increasing with same value so edges which forms MST will remain intact.

Q is false . See counter example



See here shortest path from S to U was initially S-T-U but after increasing there value by 2 it becomes S-U.

Note here Min Spanning tree edges remains same ST and TU.

So P true and Q false So Option A is Ans.

15 votes

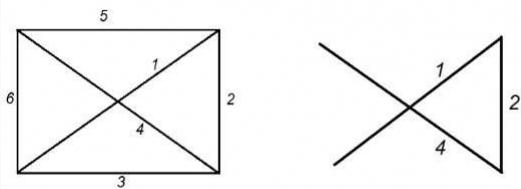
-- Rajesh Pradhan (16.9k points)

1.14.2 Spanning Tree: GATE 2016-1-39 [top](#)

<http://gateoverflow.in/39725>



Graph G can be like this:



Weight: $1 + 2 + 4 = 7$

26 votes

-- shaiklam09 (265 points)

ans is 7.

it is said maximum weight pssbl.

draw a triangle. 3 sides weight 1 2 3. and 4th point is in center. join it with triangle vertices.. got more 3 sides. new side weight 4 5 6. now draw mst. take 1 take 2. cant take 3, so take 4. $1+2+4=7$

13 votes

-- Debasmita Bhoumik (2.2k points)

1.14.3 Spanning Tree: GATE 2016-1-40 [top](#)

<http://gateoverflow.in/39726>



I think answer is option B

Statement 2 is correct absolutely. if e is the heaviest edge in cycle every mst excludes it.

Regarding statement 1, It is not fully right i think. When we think of a complete graph with 4 vertices and edge weights 1,2,5,6 in non diagonal and diagonal edges 3 and 4. 4,5,6 will create a cycle and we can exclude the highest edge e (4) from it, in a MST

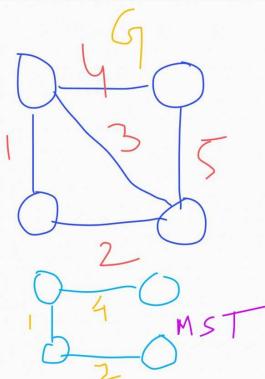
So i think answer could be B

25 votes

-- Sreyas S (1.8k points)

Statement 1:- False by [Cut Property of MST]

See counter example :- Here in below Graph G in (cycle 3,4,5) 3 is the lightest edge and still it is not included in MST.



Statement 2:->True by[Cycle property of MST] :->(in above Grahp G 1-2-3 is a cycle and 3 is the heaviest edge) If heaviest edge is in cycle then we will always exclude that bcz Cycle is their means we can have other choice of low cost edges.

So Option B is Ans.

Must visit Links

<http://www.cs.princeton.edu/courses/archive/spr07/cos226/lectures/mst.pdf>

<http://www.cs.toronto.edu/~vassos/teaching/c73/handouts/cut-property.pdf>

16 votes

-- Rajesh Pradhan (16.9k points)

1.14.4 Spanning Tree: GATE1991_03,vi top

<http://gateoverflow.in/521>



Answer: D, B, E

When Union-Find algorithm is used to detect cycle while constructing the MST time complexity is $O(m \log n)$ where m is the number of edges, and n is the number of vertices. Since $n = O(m^2)$ in a graph, options B and E are also correct as big-O specifies asymptotic upper bound only.

Ref: <http://www.geeksforgeeks.org/greedy-algorithms-set-2-kruskals-minimum-spanning-tree-mst/>

7 votes

-- Rajarshi Sarkar (34.4k points)

1.14.5 Spanning Tree: GATE1992_01,ix top

<http://gateoverflow.in/549>



if all edges are already sorted then this problem will reduced to union-find problem on a graph with E edges and V vertices.

```
for each edge (u,v) in E
    if(FIND-SET(u) != FIND-SET(v))
        UNION(u,v)
```

FIND-SET(v) and UNION(u,v) runs in $\alpha(|V|)$

where $\alpha(n)$ is inverse ackermann function i.e $\log^*(n)$

So overall complexity becomes $O(|E| \cdot \alpha(|V|))$

13 votes

-- Vikrant Singh (13.4k points)

1.14.6 Spanning Tree: GATE1995_22 [top](#)



Selected Answer

2 only.

{AB,BC,AE,BD} and {AB,BC,AE,CD}.

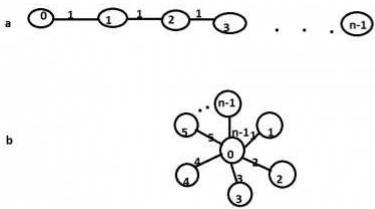
14 votes

-- Gate Keeda (18.8k points)

1.14.7 Spanning Tree: GATE1996_16 [top](#)



Selected Answer



minimum spanning tree

8 votes

-- Anu (10.3k points)

1.14.8 Spanning Tree: GATE1997_9 [top](#)



Selected Answer

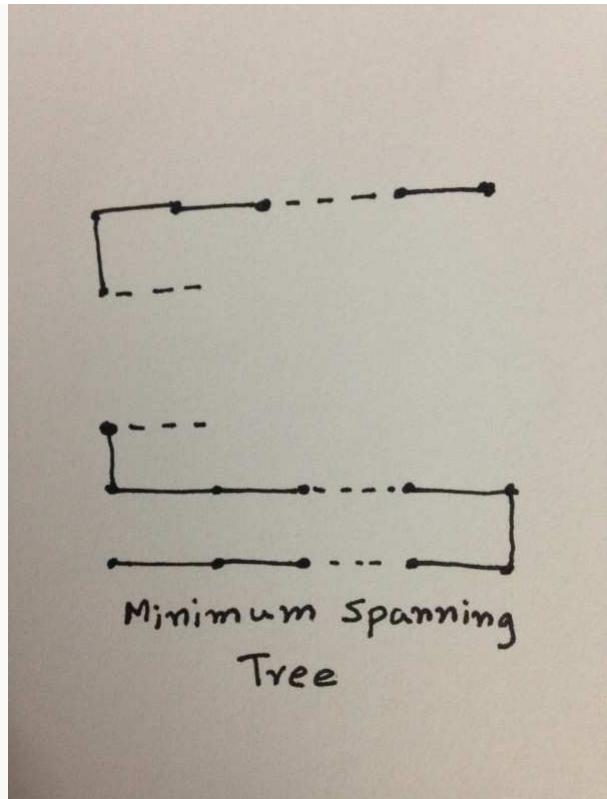
A. In the graph start applying Kruskal algorithm. So, we select all edges of cost 1 first.

For each $y = i$, there will be $n - 1$ unit length path. Total cost = $n(n - 1)$

Then we can join these n lines by one vertical line whose length is $n - 1$.

Total:

$$n(n - 1) + (n - 1) = (n - 1) \cdot (n + 1) = n^2 - 1$$



B. We can apply Kruskal's algorithm by taking maximum weight edge ([ref](#)). Maximum weight edge here is $\sqrt{2}$ as we are taking Euclidean distance as mentioned in question.

All diagonals will sum to:

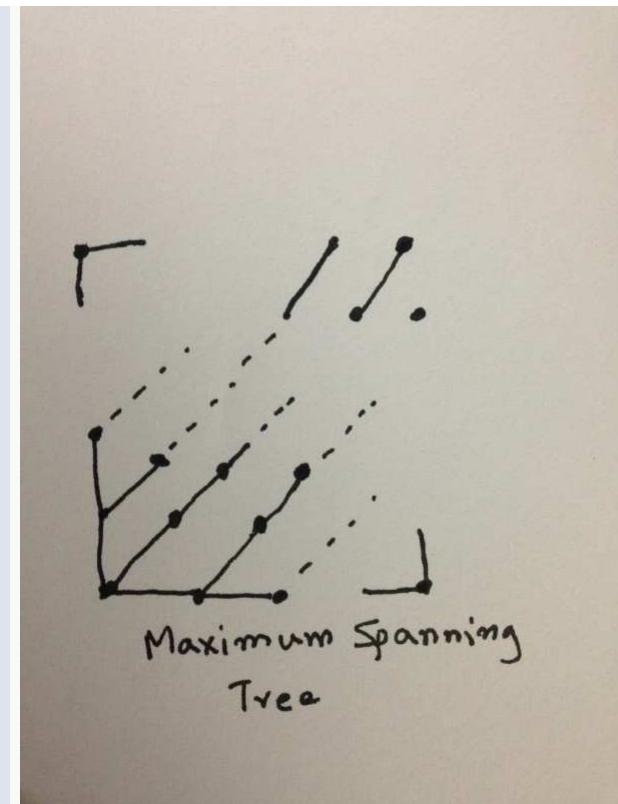
$$\sqrt{2}(n - 1) + 2 \quad ($$

(One main diagonal and
 $(n - 1)$ diagonals on either side of it).

Now, we can no more add any edge of weight $\sqrt{2}$ without forming a cycle.

So, we take $2(n - 1)$ edges of weight 1 to interconnect these diagonals

$$= \sqrt{2}(n - 1)^2 + 2(n - 1) = \sqrt{2} \cdot (n - 1) \cdot (n - 1 + \sqrt{2})$$



6 votes

-- Digvijay (46k points)

1.14.9 Spanning Tree: GATE2000-2.18 top

<http://gateoverflow.in/665>



Selected Answer

C. the case should be written as "may or may not", to be true.

D will always be true as per the question saying that the graph has distinct weights.

13 votes

-- Gate Keeda (18.8k points)

1.14.10 Spanning Tree: GATE2001-15 top

<http://gateoverflow.in/758>



Selected Answer

a) edges with weight: 2,3,4,7,9

b) no of distinct minimum spanning tree: 2 (2nd with different edge of weight 4)

c) yes

d) yes

5 votes

-- jayendra (7.7k points)

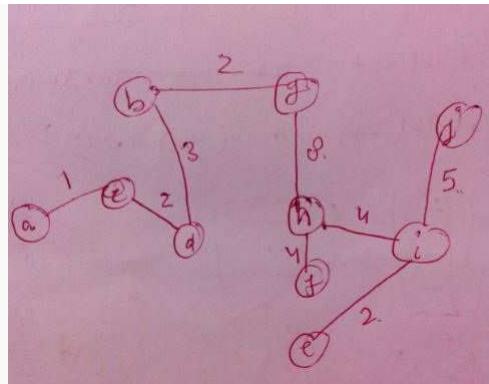
1.14.11 Spanning Tree: GATE2003-68 top

<http://gateoverflow.in/955>



Selected Answer

Apply Prim's algorithm, start from A as shown in figure below.



add all the weights in the given figure which would be equal to 31.

3 votes

-- Monanshi Jain (8.1k points)

1.14.12 Spanning Tree: GATE2005-6 [top](#)

<http://gateoverflow.in/1348>



Selected Answer

Graph G has multiple distinct MSTs, each of cost $n - 1$

From the given data given graph is a complete graph with all edge weights 1. A MST will contain $n - 1$ edges . Hence weight of MST is $n - 1$.

The graph will have multiple MST. In fact all spanning trees of the given graph will be MSTs also since all edge weights are equal.

10 votes

-- Sankaranarayanan P.N (11k points)

1.14.13 Spanning Tree: GATE2005-IT-52 [top](#)

<http://gateoverflow.in/3813>



Selected Answer

Option a is always true

Option b is not true when e is not part of a cycle.

Option c is not true when e is part of a cycle and all edge weights are same in that cycle

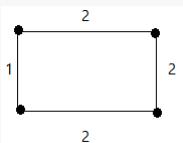
Option d is need not be true when e is not part of a cycle

Option a is always true as only the min weight edge in a cut set will be part of a minimum spanning tree.

9 votes

-- Bhagirathi Nayak (13.1k points)

Option A is correct.



Here, in this example we can easily see that B, C, D are false. So, B,C,D are not always true.

that's why A is always true.. **A (Ans)**

11 votes

-- Himanshu Agarwal (15.9k points)

1.14.14 Spanning Tree: GATE2006-11 [top](#)<http://gateoverflow.in/890>

Selected Answer

2(n-1) the spanning tree will traverse adjacent edges since they contain the least weight.

11 votes

-- anshu (3k points)

1.14.15 Spanning Tree: GATE2006-47 [top](#)<http://gateoverflow.in/1823>

Selected Answer

in kruskal's algo the edges are added in non decreasing order of their weight. But in Option D edge d-e with weight 3 is added before edge d-c with weight 2. Hence Option D is wrong option

7 votes

-- Sankaranarayanan P.N (11k points)

1.14.16 Spanning Tree: GATE2007-49 [top](#)<http://gateoverflow.in/1247>

Selected Answer

D is the false statement.

A minimum spanning tree must have the edge with the smallest weight (In Kruskal's algorithm we start from the smallest weight edge). So, C is TRUE.

If e is not part of a minimum spanning tree, then all edges which are part of a cycle with e, must have weight $\leq e$, as otherwise we can interchange that edge with e and get another minimum spanning tree of lower weight. So, B and A are also TRUE.

9 votes

-- Arjun Suresh (287k points)

1.14.17 Spanning Tree: GATE2008-IT-45 [top](#)<http://gateoverflow.in/3355>

Selected Answer

Prim's algorithm starts with from any vertex and expands the MST by adding one vertex in each step which is close to the Intermediate MST(made till previous step).

Therefore correct answer would be (C).

(A): (d,f) is chosen but neither d nor f vertices are part of the previous MST(MST made till previous step).

(B): (g,h) is chosen but neither g or h vertices are part of the previous MST(MST made till previous step).

(D): (f,c) is chosen but at that point (f,d) is close to the intermediate MST.

11 votes

-- suraj (5k points)

1.14.18 Spanning Tree: GATE2009-38 [top](#)<http://gateoverflow.in/1324>

Selected Answer

in option d b-c with weight 4 is added before a-c with weight 3 is added. In kruskal's algorithm edges should be added in non decreasing order of weight

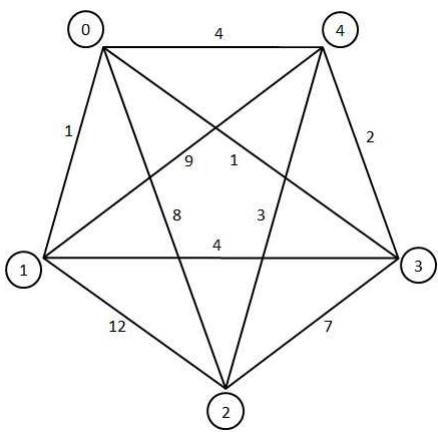
So option D may be correct

9 votes

-- Sankaranarayanan P.N (11k points)

1.14.19 Spanning Tree: GATE2010-50 [top](#)<http://gateoverflow.in/2355>

Selected Answer



Answer is (D) 10. The edges of the spanning tree are: 0 - 1, 1 - 3, 3 - 4, 4 - 2. Total Weight = 10

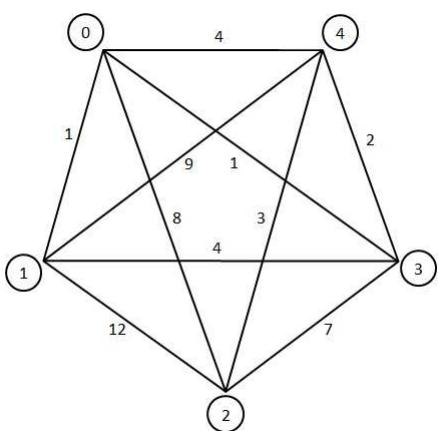
13 votes

-- Ashis Kumar Sahoo (813 points)

1.14.20 Spanning Tree: GATE2010-51 [top](#)

<http://gateoverflow.in/43328>

Selected Answer



Answer is (B) 8. The possible path is: 1 - 0, 0 - 4, 4 - 2.

7 votes

-- Ashis Kumar Sahoo (813 points)

1.14.21 Spanning Tree: GATE2011-54 [top](#)

<http://gateoverflow.in/2162>

Selected Answer

Q 54. Answer is **B**

We observe a pattern in the weight of MST being formed

For n=3 ($1 + 2 + 3$) + (1)

For $n=4$ $(1 + 2 + 3 + 4) + (1 + 2)$

For $n=5$ $(1 + 2 + 3 + 4 + 5) + (1 + 2 + 3)$

These can be obtained by drawing graphs for these graphs.

\therefore Total weight of MST is $\sum_{i=1}^n i + \sum_{i=1}^{n-2} i = n^2 - n + 1$

11 votes

-- Sona Praneeth Akula (4k points)

1.14.22 Spanning Tree: GATE2011-55 [top](#)



Selected Answer

There is no direct edge between V_i and V_{i+1} vertex in mst except V1 to V2 so path from V_i and V_{i+1} includes all edges from v1 to V_{i+1} which are in mst:

edge weights in mst:

$$= 3 + 4 + 6 + 8 + 10 + 12 + \dots + *(2(n-1))$$

$$= 1 + (2 + 4 + 6 + \dots \text{ till } n-1 \text{ terms})$$

$$= 1 + 2(1+2+3+4+\dots+n-1)$$

$$= 1 + (n-1)*n = n^2 - n + 1$$

$$\text{In this case } v_6 = 6^2 - 6 + 1 = 31$$

4 votes

-- papesh (22.8k points)

1.14.23 Spanning Tree: GATE2012_29 [top](#)



Selected Answer

When the edge weights are squared the minimum spanning tree won't change.

$t' < t^2$, because sum of squares is always less than the square of the sums except for a single element case.

Hence, B is the general answer and A is also true for a single edge graph. Hence, in GATE 2012, marks were given to all.

20 votes

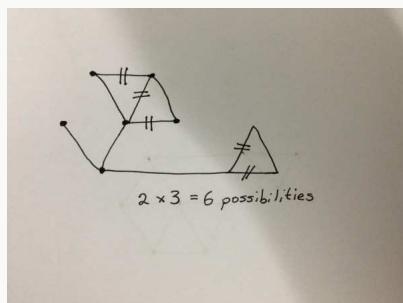
-- gatecse (13.1k points)

1.14.24 Spanning Tree: GATE2014-2-52 [top](#)



Selected Answer

6 is the answer.



17 votes

-- Arjun Suresh (287k points)

1.14.25 Spanning Tree: GATE2015-1_43 [top](#)

<http://gateoverflow.in/6313>



Selected Answer

Consider the cycle ABC. AC and AB are part of minimum spanning tree. So, AB should be greater than $\max(AC, BC)$ (greater and not equal as edge weights are given to be distinct), as otherwise we could add AB to the minimum spanning tree and removed the greater of AC, BC and we could have got another minimum spanning tree. So, $AB > 9$.

Similarly, for the cycle DEF, $ED > 6$.

And for the cycle BCDE, $CD > 15$.

So, minimum possible sum of these will be $10 + 7 + 16 = 33$. Adding the weight of spanning tree, we get the total sum of edge weights

$$= 33 + 36 = 69$$

24 votes

-- Arjun Suresh (287k points)

1.14.26 Spanning Tree: GATE2015-3_40 [top](#)

<http://gateoverflow.in/8499>



Selected Answer

first find no of edges in mst...

mst has $n-1$ edges where n is no of vertices. $100-1 = 99$ edges
each 99 edges in mst increases by 5 so weight in mst increased $99*5=495$
now total weight of mst = $500+495=995$

22 votes

-- Anoop Sonkar (4.8k points)

1.14.27 Spanning Tree: TIFR2011-B-35 [top](#)

<http://gateoverflow.in/20842>



Selected Answer

a) & c) are trivially true. Edge with max value e_1 must be present in Maximum spanning tree & same with minimum.

e) This is true, because all egde weights are distinct. maximum spanning tree is unique.

b) e_1 & e_2 must be present in Maximum spanning tree. I'll prove it using Kruskal Algorithm.

We will first insert weight with biggest value, e_1 . Then we insert e_2 (second highest) . 2 edges do not create **cycle**. Then we can go on from there inserting edges according to edge weights. As they have just asked for top 2 edges, using Kruskal Algo we can say that top 2 edges must be in Maximum spanning tree.

d)

This is false. There are chances that this e_m weight edge is **cut edge(Bridge)** Then it must be inserted to from any spanning tree.

D is answer !

We can not say the same for Top 3 as they can create cycle & They we can not take a3 to make spanning tree.

Kruskal Algo Reference -> <http://stackoverflow.com/questions/4992664/how-to-find-maximum-spanning-tree>

3 votes

-- Akash (41.5k points)

1.14.28 Spanning Tree: TIFR2013-B-17 [top](#)

<http://gateoverflow.in/25860>



Selected Answer

There will be unique min weight spanning tree since all weights are distinct.
option A.

8 votes

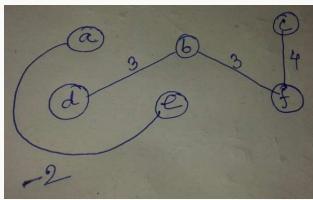
-- Umang Raman (14.5k points)

1.14.29 Spanning Tree: TIFR2014-B-4 [top](#)

<http://gateoverflow.in/27174>



Selected Answer



Now check this diagram, this is forest obtained from above given graph using Kruskal's algorithm for MST.

So according to the question edge $d - e$ has weight 5 and it is included in the formation of MST. Now if edges $b - e$ and $e - f$ has weight greater than 5 than it is not a problem for our MST because still we will get the given tree as Kruskal's algorithm takes the smallest weighted edge without forming a cycle.

Cost of edge $b - c \geq 4$ may also lead us to the same tree as above though Kruskal's algorithm will have choice between $c - f$ and $b - c$.

Now if the edge weight of $a - d$ becomes 4, it is guaranteed that Kruskal's algorithm will not select edge $d - e$ because its edge cost is 5, and hence the tree structure will change. But there can be the case where edge weight is greater than 4 and we still get the same tree (happens when $a - d \geq 5$). Because in the question they asked to point out an unnecessary condition this case is not the answer as we need $a - d \geq 5$ which implies $a - d \geq 4$.

Now notice option A. Put $a - b = 5$. The given MST would not change. So, this condition is not always necessary and hence is the answer..

Therefore option A is the answer .

8 votes

-- Riya Roy(Arayana) (7k points)

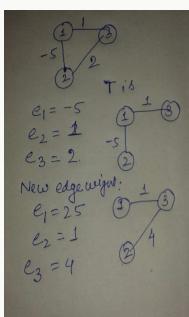
1.14.30 Spanning Tree: TIFR2014-B-5 [top](#)

<http://gateoverflow.in/27180>



Selected Answer

Answer is E . The catch here is Edges weights belongs to real number . Therefore edge weight can be negative . In that case the minimum spanning tree may be different .



EDIT-

(Here every edge weight is distinct, therefore MST is unique. You do it using any algo.)

Option A is True. If we apply kruskal's algorithm then it will choose e_1

Option B is True. If we apply kruskal's algorithm then it will also choose e_2 , and 2 edges can not forms a cycle. (e_3 is not guaranteed in MST, as it may form cycle.)

Option C is also true. If we apply prims also on any vertex (say u) then it chooses minimum weight edge incident on vertex u.



Option D is true. Because every edge weight is distinct.

9 votes

-- Riya Roy(Arayana) (7k points)

1.14.31 Spanning Tree: TIFR2015-B-2 [top](#)

<http://gateoverflow.in/29844>



Selected Answer

In the graph we have all edge weights are distinct so we will get unique minimum and maximum spanning tree.

Each Cycle must exclude maximum weight edge in minimum spanning tree.

Here we have two cycle of 3 edges , ade and cgk .

for second best minimum spanning tree = exclude ae edge and include de edge

other way : second best minimum spanning tree= exclude cg edge and include gk edge.

so e should be the ans.

3 votes

-- papesh (22.8k points)

1.15

Time Complexity(32) [top](#)

1.15.1 Time Complexity: CMI2013-A-10 [top](#)

<http://gateoverflow.in/46602>

The below question is based on following program:

```
procedure mystery (A : array [1..100] of int)
    int i,j,position,tmp;
begin
    for j := 1 to 100 do
        position := j;
        for i := j to 100 do
            if (A[i] > A[position]) then
                position := i;
            endfor;
            tmp := A[j];
            A[j] := A[position];
            A[position] := tmp;
        endfor;
end
```

The number of times the test $A[i] > A[position]$ is executed is:

- A. 100
- B. 5050
- C. 10000
- D. Depends on contents of A

[cmi2013](#) [algorithms](#) [time-complexity](#)

Answer

1.15.2 Time Complexity: CMI2015-A-08 [top](#)

<http://gateoverflow.in/47045>

How many times is the comparison $i \geq n$ performed in the following program?

```
int i=85, n=5;
main() {
    while (i >= n) {
        i=i-1;
        n=n+1;
    }
}
```

- A. 40
B. 41
C. 42
D. 43

cmi2015 | algorithms | time-complexity

[Answer](#)

1.15.3 Time Complexity: GATE1989-2-iii [top](#)

<http://gateoverflow.in/87080>

Match the pairs in the following questions:

- | | |
|-------------------|--|
| (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

[Answer](#)

1.15.4 Time Complexity: GATE1993_8.7 [top](#)

<http://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

[Answer](#)

1.15.5 Time Complexity: GATE1999-1.13 [top](#)

<http://gateoverflow.in/1469>

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

[Answer](#)

1.15.6 Time Complexity: GATE1999-1.16 [top](#)

<http://gateoverflow.in/1469>

If n is a power of 2, then the minimum number of multiplications needed to compute a^n is

- A. $\log_2 n$
- B. \sqrt{n}
- C. $n - 1$
- D. n

gate1999 | algorithms | time-complexity | normal

[Answer](#)

1.15.7 Time Complexity: GATE1999_11a [top](#)

<http://gateoverflow.in/1510>

Consider the following algorithms. Assume, procedure A and procedure B take $O(1)$ and $O(1/n)$ unit of time respectively. Derive the time complexity of the algorithm in O -notation.

```
algorithm what (n)
begin
    if n = 1 then call A
    else
        begin
            what (n-1);
            call B(n)
        end
    end.
```

gate1999 | algorithms | time-complexity | normal

[Answer](#)

1.15.8 Time Complexity: GATE2000-1.15 [top](#)

<http://gateoverflow.in/638>

Let S be a sorted array of n integers. Let $T(n)$ denote the time taken for the most efficient algorithm to determine if there are two elements with sum less than 1000 in S . Which of the following statement is true?

- A. $T(n)$ is $O(1)$
- B. $n \leq T(n) \leq n \log_2 n$
- C. $n \log_2 n \leq T(n) < \frac{n}{2}$
- D. $T(n) = \left(\frac{n}{2}\right)$

gate2000 | easy | algorithms | time-complexity

[Answer](#)

1.15.9 Time Complexity: GATE2003-66 [top](#)

<http://gateoverflow.in/258>

The cube root of a natural number n is defined as the largest natural number m such that $(m^3 \leq n)$. The complexity of computing the cube root of n (n is represented by binary notation) is

- A. $O(n)$ but not $O(n^{0.5})$
- B. $O(n^{0.5})$ but not $O((\log n)^k)$ for any constant $k > 0$
- C. $O((\log n)^k)$ for some constant $k > 0$, but not $O((\log \log n)^m)$ for any constant $m > 0$
- D. $O((\log \log n)^k)$ for some constant $k > 0.5$, but not $O((\log \log n)^{0.5})$

gate2003 | algorithms | time-complexity | normal

[Answer](#)

1.15.10 Time Complexity: GATE2004-39 [top](#)

<http://gateoverflow.in/1036>

Two matrices M_1 and M_2 are to be stored in arrays A and B respectively. Each array can be stored either in row-major or column-major order in contiguous memory locations. The time complexity of an algorithm to compute $M_1 \times M_2$ will be

- A. best if A is in row-major, and B is in column-major order
- B. best if both are in row-major order
- C. best if both are in column-major order

- D. independent of the storage scheme

[gate2004](#) [algorithms](#) [time-complexity](#) [easy](#)

[Answer](#)

1.15.11 Time Complexity: GATE2004-82 [top](#)

<http://gateoverflow.in/1076>

Let $A[1, \dots, n]$ be an array storing a bit (1 or 0) at each location, and $f(m)$ is a function whose time complexity is $\Theta(m)$. Consider the following program fragment written in a C like language:

```
counter = 0;
for (i=1; i<=n; i++)
{ if a[i] == 1) counter++;
else {f(counter); counter = 0;}
}
```

The complexity of this program fragment is

- A. $\Omega(n^2)$
- B. $\Omega(n \log n)$ and $O(n^2)$
- C. $\Theta(n)$
- D. $o(n)$

[gate2004](#) [algorithms](#) [time-complexity](#) [normal](#)

[Answer](#)

1.15.12 Time Complexity: GATE2006-15 [top](#)

<http://gateoverflow.in/976>

Consider the following C-program fragment in which i , j and n are integer variables.

```
for( i = n, j = 0; i > 0; i /= 2, j += i );
```

Let $val(j)$ denote the value stored in the variable j after termination of the for loop. Which one of the following is true?

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

[gate2006](#) [algorithms](#) [normal](#) [time-complexity](#)

[Answer](#)

1.15.13 Time Complexity: GATE2007-15,ISRO2016-26 [top](#)

<http://gateoverflow.in/56129>

Consider the following segment of C-code:

```
int j, n;
j = 1;
while (j <= n)
    j = j * 2;
```

The number of comparisons made in the execution of the loop for any $n > 0$ is:

- A. $\lceil \log_2 n \rceil + 1$
- B. n
- C. $\lceil \log_2 n \rceil$
- D. $\lfloor \log_2 n \rfloor + 1$

[gate2007](#) [algorithms](#) [time-complexity](#) [normal](#) [isro2016](#)
Answer

1.15.14 Time Complexity: GATE2007-44 [top](#)

<http://gateoverflow.in/1242>

In the following C function, let $n \geq m$.

```
int gcd(n,m) {
    if (n%m == 0) return m;
    n = n%m;
    return gcd(m,n);
}
```

How many recursive calls are made by this function?

- A. $\Theta(\log_2 n)$
- B. $\Omega(n)$
- C. $\Theta(\log_2 \log_2 n)$
- D. $\Theta(\sqrt{n})$

[gate2007](#) [algorithms](#) [time-complexity](#) [normal](#)
Answer

1.15.15 Time Complexity: GATE2007-45 [top](#)

<http://gateoverflow.in/1243>

What is the time complexity of the following recursive function?

```
int DoSomething (int n) {
    if (n <= 2)
        return 1;
    else
        return (DoSomething (floor (sqrt(n))) + n);
}
```

- A. $\Theta(n^2)$
- B. $\Theta(n \log_2 n)$
- C. $\Theta(\log_2 n)$
- D. $\Theta(\log_2 \log_2 n)$

[gate2007](#) [algorithms](#) [time-complexity](#) [normal](#)
Answer

1.15.16 Time Complexity: GATE2007-50 [top](#)

<http://gateoverflow.in/1248>

An array of n numbers is given, where n is an even number. The maximum as well as the minimum of these n numbers needs to be determined. Which of the following is TRUE about the number of comparisons needed?

- A. At least $2n - c$ comparisons, for some constant c are needed.
- B. At most $1.5n - 2$ comparisons are needed.
- C. At least $n \log_2 n$ comparisons are needed
- D. None of the above

[gate2007](#) [algorithms](#) [time-complexity](#) [easy](#)
Answer

1.15.17 Time Complexity: GATE2007-51 [top](#)

<http://gateoverflow.in/1249>

Consider the following C program segment:

```

int IsPrime(n)
{
    int i, n;
    for (i=2; i<=sqrt(n); i++)
        if (n%i == 0)
            (printf("Not Prime \n"); return 0);
    return 1;
}

```

Let $T(n)$ denote number of times the *for* loop is executed by the program on input n . Which of the following is TRUE?

- A. $T(n) = O(\sqrt{n})$ and $T(n) = \Omega(\sqrt{n})$
- B. $T(n) = O(\sqrt{n})$ and $T(n) = \Omega(1)$
- C. $T(n) = O(n)$ and $T(n) = \Omega(\sqrt{n})$
- D. None of the above

[gate2007](#) [algorithms](#) [time-complexity](#) [normal](#)

[Answer](#)

1.15.18 Time Complexity: GATE2007-IT-17 [top](#)

<http://gateoverflow.in/3450>

Exponentiation is a heavily used operation in public key cryptography. Which of the following options is the tightest upper bound on the number of multiplications required to compute $b^n \bmod m, 0 \leq b, n \leq m$?

- A. $O(\log n)$
- B. $O(\sqrt{n})$
- C. $O\left(\frac{n}{\log n}\right)$
- D. $O(n)$

[gate2007-it](#) [algorithms](#) [time-complexity](#) [normal](#)

[Answer](#)

1.15.19 Time Complexity: GATE2007-IT-81 [top](#)

<http://gateoverflow.in/3533>

Let P_1, P_2, \dots, P_n be n points in the xy -plane such that no three of them are collinear. For every pair of points P_i and P_j , let L_{ij} be the line passing through them. Let L_{ab} be the line with the steepest gradient among all $n(n-1)/2$ lines.

The time complexity of the best algorithm for finding P_a and P_b is

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

[gate2007-it](#) [algorithms](#) [time-complexity](#) [normal](#)

[Answer](#)

1.15.20 Time Complexity: GATE2008-40 [top](#)

<http://gateoverflow.in/452>

The minimum number of comparisons required to determine if an integer appears more than $\frac{n}{2}$ times in a sorted array of n integers is

- A. $\Theta(n)$
- B. $\Theta(\log n)$
- C. $\Theta(\log^* n)$
- D. $\Theta(1)$

[gate2008](#) [normal](#) [algorithms](#) [time-complexity](#)

Answer**1.15.21 Time Complexity: GATE2008-47** [top](#)<http://gateoverflow.in/459>

We have a binary heap on n elements and wish to insert n more elements (not necessarily one after another) into this heap. The total time required for this is

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

[gate2008](#) [algorithms](#) [time-complexity](#) [normal](#)
Answer**1.15.22 Time Complexity: GATE2008-74** [top](#)<http://gateoverflow.in/495>

Consider the following C functions:

```
int f1 (int n)
{
    if(n == 0 || n == 1)
        return n;
    else
        return (2 * f1(n-1) + 3 * f1(n-2));
}
int f2(int n)
{
    int i;
    int X[N], Y[N], Z[N];
    X[0] = Y[0] = Z[0] = 0;
    X[1] = 1; Y[1] = 2; Z[1] = 3;
    for(i = 2; i <= n; i++){
        X[i] = Y[i-1] + Z[i-2];
        Y[i] = 2 * X[i];
        Z[i] = 3 * X[i];
    }
    return X[n];
}
```

The running time of $f1(n)$ and $f2(n)$ are

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

[gate2008](#) [algorithms](#) [time-complexity](#) [normal](#)
Answer**1.15.23 Time Complexity: GATE2008-75** [top](#)<http://gateoverflow.in/43489>

Consider the following C functions:

```
int f1 (int n)
{
    if(n == 0 || n == 1)
        return n;
    else
        return (2 * f1(n-1) + 3 * f1(n-2));
}
int f2(int n)
{
    int i;
    int X[N], Y[N], Z[N];
    X[0] = Y[0] = Z[0] = 0;
    X[1] = 1; Y[1] = 2; Z[1] = 3;
    for(i = 2; i <= n; i++){
        X[i] = Y[i-1] + Z[i-2];
        Y[i] = 2 * X[i];
        Z[i] = 3 * X[i];
    }
}
```

```

    }
    return X[n];
}

```

f1(8) and f2(8) return the values

- A. 1661 and 1640
- B. 59 and 59
- C. 1640 and 1640
- D. 1640 and 1661

[gate2008](#) [normal](#) [algorithms](#) [time-complexity](#)

[Answer](#)

1.15.24 Time Complexity: GATE2010-12 [top](#)

<http://gateoverflow.in/2185>

Two alternative packages A and B are available for processing a database having 10^k records. Package A requires $0.0001n^2$ time units and package B requires $10n \log_{10} n$ time units to process n records. What is the smallest value of k for which package B will be preferred over A ?

- A. 12
- B. 10
- C. 6
- D. 5

[gate2010](#) [algorithms](#) [time-complexity](#) [easy](#)

[Answer](#)

1.15.25 Time Complexity: GATE2014-1-42 [top](#)

<http://gateoverflow.in/1920>

Consider the following pseudo code. What is the total number of multiplications to be performed?

```

D = 2
for i = 1 to n do
    for j = i to n do
        for k = j + 1 to n do
            D = D * 3

```

- A. Half of the product of the 3 consecutive integers.
- B. One-third of the product of the 3 consecutive integers.
- C. One-sixth of the product of the 3 consecutive integers.
- D. None of the above.

[gate2014-1](#) [algorithms](#) [time-complexity](#) [normal](#)

[Answer](#)

1.15.26 Time Complexity: GATE2015-1_40 [top](#)

<http://gateoverflow.in/8299>

An algorithm performs $(\log N)^{1/2}$ find operations, N insert operations, $(\log N)^{1/2}$ delete operations, and $(\log N)^{1/2}$ decrease-key operations on a set of data items with keys drawn from a linearly ordered set. For a delete operation, a pointer is provided to the record that must be deleted. For the decrease-key operation, a pointer is provided to the record that has its key decreased. Which one of the following data structures is the most suited for the algorithm to use, if the goal is to achieve the best total asymptotic complexity considering all the operations?

- A. Unsorted array
- B. Min - heap
- C. Sorted array
- D. Sorted doubly linked list

[gate2015-1](#) [algorithms](#) [data-structure](#) [normal](#) [time-complexity](#)

[Answer](#)

1.15.27 Time Complexity: GATE2015-2_22 [top](#)

<http://gateoverflow.in/8113>

An unordered list contains n distinct elements. The number of comparisons to find an element in this list that is neither maximum nor minimum is

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

gate2015-2 | algorithms | time-complexity | easy

Answer

1.15.28 Time Complexity: GATE2017-2-03 [top](#)

<http://gateoverflow.in/118158>

Match the algorithms with their time complexities:

Algorithms	Time Complexity
P. TOwers of honoi with n disks	i. $\Theta(n^2)$
Q. Binary Search given n numbers n sorted numbers	ii. $\Theta(n \log n)$
R. Heap sort given n numbers at the worst case	iii. $\Theta(2^n)$
S. Addition of two nxn matrices	iv. $\Theta(\log n)$

- A. P → (iii) Q → (iv) R → (i) S → (ii)
 B. P → (iv) Q → (iii) R → (i) S → (ii)
 C. P → (iii) Q → (iv) R → (ii) S → (i)
 D. P → (iv) Q → (iii) R → (ii) S → (i)

gate2017-2 | algorithms | time-complexity

Answer

1.15.29 Time Complexity: GATE2017-2-38 [top](#)

<http://gateoverflow.in/118283>

Consider the following C function

```
int fun(int n) {
    int I, j;
    for(i=1; i<=n; i++) {
        for (j=1; j<n; j+=i) {
            printf("%d %d", I, j);
        }
    }
}
```

Time complexity of *fun* in terms of θ notation is

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

gate2017-2 | algorithms | time-complexity

Answer

1.15.30 Time Complexity: TIFR2013-B-12 [top](#)

<http://gateoverflow.in/25774>

It takes $O(n)$ time to find the median in a list of n elements, which are not necessarily in sorted order while it takes only $O(1)$ time to find the median in a list of n sorted elements. How much time does it take to find the median of $2n$ elements which are given as two lists of n sorted elements each?

- a. $O(1)$
 b. $O(\log n)$ but not $O(1)$
 c. $O(\sqrt{n})$ but not $O(\log n)$
 d. $O(n)$ but not $O(\sqrt{n})$

- e. $O(n \log n)$ but not $O(n)$

tifr2013 | algorithms | time-complexity

Answer

1.15.31 Time Complexity: TIFR2014-B-7 [top](#)

<http://gateoverflow.in/27189>

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

- a. $(\log n)^{\log \log n} < 2^{\sqrt{\log n}} < n^{1/4}$
- b. $2^{\sqrt{\log n}} < n^{1/4} < (\log n)^{\log \log n}$
- c. $n^{1/4} < (\log n)^{\log \log n} < 2^{\sqrt{\log n}}$
- d. $(\log n)^{\log \log n} < n^{1/4} < 2^{\sqrt{\log n}}$
- e. $2^{\sqrt{\log n}} < (\log n)^{\log \log n} < n^{1/4}$

tifr2014 | algorithms | time-complexity

Answer

1.15.32 Time Complexity: TIFR2015-B-3 [top](#)

<http://gateoverflow.in/29846>

Consider the following code fragment in the C programming language when run on a non-negative integer n .

```
int f ( int n )
{
    if ( n==0 || n==1 )
        return 1;
    else
        return f ( n - 1 ) + f ( n - 2 );
}
```

Assuming a typical implementation of the language, what is the running time of this algorithm and how does it compare to the optimal running time for this problem?

- a. This algorithm runs in polynomial time in n but the optimal running time is exponential in n .
- b. This algorithm runs in exponential time in n and the optimal running time is exponential in n .
- c. This algorithm runs in exponential time in n but the optimal running time is polynomial in n .
- d. This algorithm runs in polynomial time in n and the optimal running time is polynomial in n .
- e. The algorithm does not terminate.

tifr2015 | time-complexity

Answer

Answers: Time Complexity

1.15.1 Time Complexity: CMI2013-A-10 [top](#)

<http://gateoverflow.in/46602>



Selected Answer

Ans- $5050 (100 + 99 + 98 + \dots + 1) = (100 * 101)/2$

2 votes

-- Dhananjay Kumar Sharma (24.9k points)

1.15.2 Time Complexity: CMI2015-A-08 [top](#)

<http://gateoverflow.in/47045>



Selected Answer

Correct answer is C) 42

It will start comparison from (85, 5), (84, 6), (83, 7) (45, 45), (44, 46)

Hence total number of comparison will be **(85-44) + 1 = 41 + 1 = 42**

8 votes

-- Muktinath Vishwakarma (33.9k points)

1.15.3 Time Complexity: GATE1989-2-iii [top](#)

<http://gateoverflow.in/87080>



Selected Answer

(A) O(logn)	R.Binary Search
(B) O(n)	S.Selection of the k^{th} smallest element in a set of n elements. (Worst case)
(C) O(nlogn)	P.Heapsort
(D) O(n^2)	Q.Depth-first search(worst case) If we consider Adjacency matrix

3 votes

-- Prajwal Bhat (11.6k points)

1.15.4 Time Complexity: GATE1993_8.7 [top](#)

<http://gateoverflow.in/2305>



Selected Answer

This is N added itself N times. So it is N^2 . Even if you consider as sum of $O(1) + O(2) + \dots + O(n-1) + O(N)$, it will add up to N^2

So answer is

A) $O(N)$ this is false.

B,C,D,E) All of this are true. We have N^2 here, so all options apart from A are correct.

In fact B = D = E these three options are same. and N^3 is always upper bound of N^2 . So $O(N^3)$ is also true.

8 votes

-- Akash (41.5k points)

1.15.5 Time Complexity: GATE1999-1.13 [top](#)

<http://gateoverflow.in/1466>



Selected Answer

Answer is (d) None of these

We just require $n/2$ swaps in the worst case. The algorithm is as given below:

Find positive number from left side and negative number from right side and do exchange. Since, at least one of them must be less than or equal to $n/2$, there cannot be more than $n/2$ exchanges. An implementation is given below:

http://gatcse.in/wiki/Moving_Negative_Numbers_to_the_Beginning_of_Array

14 votes

-- Arjun Suresh (287k points)

1.15.6 Time Complexity: GATE1999-1.16 [top](#)

<http://gateoverflow.in/1469>



Selected Answer

a. $\log n$

$$a^n = (a^2)^{\frac{n}{2}}.$$

One multiplication and recurrence on $\frac{n}{2}$. So, we get the recurrence relation for the number of multiplications as

$$T(n) = T(n/2) + 1.$$

This gives $T(n) = \log_2 n$

For $n = 8$, we can do

$$b = a \times a$$

$$b = b \times b$$

$$b = b \times b \text{ and we get } b = a^8$$

11 votes

-- Arjun Suresh (287k points)

1.15.7 Time Complexity: GATE1999_11a [top](#)



Selected Answer

The recurrence relation for time complexity is

$$T(n) = T(n - 1) + \frac{1}{n} + c \quad (O(1/n) \text{ replaced with } 1/n \text{ and so our answer will also be in } O \text{ only and not } \Theta).$$

$$T(n) = T(n - 2) + \frac{1}{n-1} + \frac{1}{n} + 2c = T(1) + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} + (n - 1)c = A(1) + \frac{1}{2} + \frac{1}{3} \dots + \frac{1}{n} + nc = 1 + \frac{1}{2} + \frac{1}{3} \dots$$

(Sum of the first n terms in harmonic series is $\Theta(\log n)$)

So, our time complexity will be $O(n)$.

9 votes

-- Arjun Suresh (287k points)

1.15.8 Time Complexity: GATE2000-1.15 [top](#)



Selected Answer

Because array is always sorted just check the 1st two elements. option a.

13 votes

-- anshu (3k points)

1.15.9 Time Complexity: GATE2003-66 [top](#)



Selected Answer

We can simply do a binary search in the array of natural numbers from $1..n$ and check if the cube of the number matches n (i.e., check if $a[i] * a[i] * a[i] == n$). This check takes $O(\log n)$ time and in the worst case we need to do the search $O(\log n)$ times. So, in this way we can find the cube root in $O(\log^2 n)$. So, options (A) and (B) are wrong.

Now, a number is represented in binary using $\log n$ bit. Since each bit is important in finding the cube root, any cube root finding algorithm must examine each bit at least once. This ensures that complexity of cube root finding algorithm cannot be lower than $\log n$. (It must be $\Omega(\log n)$). So, (D) is also false and (C) is the correct answer.

28 votes

-- gatecse (13.1k points)

1.15.10 Time Complexity: GATE2004-39 [top](#)



Selected Answer

D is correct

Here time complexity is asked, for each access of array element it will be constant,

So the time complexity will not depend upon storage. If at all program execution time is asked

a is true

20 votes

-- Anurag Semwal (7.6k points)

1.15.11 Time Complexity: GATE2004-82 [top](#)

<http://gateoverflow.in/1076>



Selected Answer

The key part in the code is "counter = 0" in the else part as we can see below.

Lets take the best case. This happens when $a[i] = 1$ for all i , and then the loop executes with time complexity $\Theta(1)$ for each iteration and hence overall time complexity of $\Theta(n)$ and we can say time complexity of the code fragment is $\Omega(n)$ and hence options a and b are false.

Now, consider the worst case. This happens when $a[i] = 0$ or when else part is executed. Here, the time complexity of each iteration will be $\Theta(counter)$ and after each else, counter is reset to 0. Let k iterations go to the else part during the worst case. Then the worst case time complexity will be $\Theta(x_1) + \Theta(x_2) + \dots + \Theta(x_k) + \Theta(n - k)$, where x_i is the value of the counter when, $A[i] = 0$ and $f(counter)$ is called. But due to $counter = 0$ after each call to $f()$, we have, $x_1 + x_2 + \dots + x_k = n$. So, $\Theta(x_1) + \Theta(x_2) + \dots + \Theta(x_k) + \Theta(n - k) = \Theta(n) + \Theta(n - k) = \Theta(n)$.

Since the time complexity is $\Omega(n)$ and $\Theta(n)$ we can say it is $\Theta(n)$ - Option (C). (Option D is false because the small o needs the growth rate to be **STRICTLY lower** and not equal to or lower as the case for big O)

If $counter = 0$ was not there in else part, then time complexity would be $\Omega(n)$ and $O(n^2)$ as in worst case we can have equal number of 0's and 1's in array a giving time complexity $\Theta(1) + \Theta(2) + \dots + \Theta(n/2) + \Theta(n/2)$ would give $O(n^2)$.

11 votes

-- Arjun Suresh (287k points)

1.15.12 Time Complexity: GATE2006-15 [top](#)

<http://gateoverflow.in/976>



Selected Answer

Answer will be $\Theta(n)$

$$j = n/2 + n/4 + n/8 + \dots + 1$$

number of iteration will be $2^k = n$ or $k = \log n$

this is in GP find sum till $\log n = \Theta(n)$

15 votes

-- rahulkrr (665 points)

1.15.13 Time Complexity: GATE2007-15,ISRO2016-26 [top](#)

<http://gateoverflow.in/56129>

n	no. of Comparision	Ceil(log ₂ n)
1	+ 1	
2 (j = 1,2)	1	
3 (j = 1,2,3)	2	
4 (j = 1,2,3,4)	2	
8 (j = 1,2,3,4,8)	3	
15 (j = 1,2,3,4,8)	4	

may be we have to count those comparisons which results in the execution of loop.

Answer should be $\text{Ceil}(\log_2 n) + 1$

EDIT: but answer could be: $\text{floor}(\log_2 n) + 2$

14 votes

-- Vikrant Singh (13.4k points)

1.15.14 Time Complexity: GATE2007-44 [top](#)

<http://gateoverflow.in/124>



Worst case will arise when both n and m are consecutive Fibonacci numbers.

$$\gcd(F_n, F_{n-1}) = \gcd(F_{n-1}, F_{n-2}) = \dots = \gcd(F_1, F_0) = 1$$

and n^{th} Fibonacci number is 1.618^n , where 1.618 is the [Golden ratio](#).

So, to find $\gcd(n, m)$, number of recursive calls will be $\Theta(\log n)$.

14 votes

-- Vikrant Singh (13.4k points)

1.15.15 Time Complexity: GATE2007-45 [top](#)

<http://gateoverflow.in/124>



We are asked the time complexity which will be the number of recursive calls in the function as in each call we perform a constant no. of operations and a recursive call. The recurrence relation for this is (considering constant time "c" as 1)

$$T(n) = T(\sqrt{n}) + 1 = T(n^{1/4}) + 2 = T(n^{1/8}) + 3$$

Going like this we will eventually reach $T(3)$ or $T(2)$. For asymptotic case this doesn't matter and we can assume we reach $T(2)$ and in next step reach $T(1)$. So, all we want to know is how many steps it takes to reach $T(1)$ which will be $1 + \text{no. of steps to reach } T(2)$.

From the recurrence relation we know that $T(2)$ happens when $n^{\left(\frac{1}{2^k}\right)} = 2$.

Taking log and equating,

$$\frac{1}{2^k} \log n = 12^k = \log n k = \log \log n .$$

So, $T(1)$ happens in $\log \log n + 1$ calls, but for asymptotic complexity we can write as $\Theta(\log \log n)$

Alternatively,

Substituting values

$$T(1) = 1$$

$$T(2) = 1$$

$$T(3) = T(1) + 1 = 2$$

$$\dots$$

$$T(8) = T(2) + 1 = 2$$

$$T(9) = T(3) + 1 = 3$$

...

$$T\left(\left((2^2)^2\right)^2\right) = T\left((2^2)^2\right) + 1 = T(2^2) + 2 = T(2) + 3 = 1 + 3 = 4, \log \log n = 3 \text{ as } n = 256 .$$

$$T\left(\left(\left((2^2)^2\right)^2\right)^2\right) = 6, \log \log n = 5 \text{ as } n = 65536 \times 65536 = 2^{32}$$

$$T(2^{2^{10}}) = T(2^{512}) + 1 = T(2^{256}) + 2 = T(2^{128}) + 3 = T(2^{64}) + 4 = T(2^{32}) + 5 = T(2^{16}) + 6 = T(2^8) + 7 = T(2^4) + 8$$

So, answer is D

<http://stackoverflow.com/questions/16472012/what-would-cause-an-algorithm-to-have-olog-log-n-complexity>

20 votes

-- Arjun Suresh (287k points)

1.15.16 Time Complexity: GATE2007-50 [top](#)

<http://gateoverflow.in/1248>



Selected Answer

[need @arjun sir to verify it]

We are requested to find the MAX and MIN of the array of n elements . This can be done as follows

Non Divide And Conquer

```
Max_Min(A, n, max, min)
max=min=a[1]
for i-> 2 to n
{
    if A[i] > max    // First Comparision
        max=A[i]
    else if A[i] < min  // Seond Comparision
        min = A[i]
}
```

Analysis

Best Case

When input is in increasing order

- First Comparision : $n - 1$ time
- Second Comparision: 1 time

Worst Case

When input is in Decreasing order

- First Comparision : $n - 1$ time
- Second Comparision: $n - 1$ time

Average Case

When First Comparision fails for half of the input

- First Comparision : $n - 1$ time
- Second Comparision: $\frac{n}{2}$ time

Divide And Conquer

Given a function to compute on n inputs the [divide-and-conquer](#) strategy suggest splitting the inputs into k distinct subsets, $1 < K \leq n$, yielding k sub problems. These Sub problems must be solved, and then a method must be found to combine sub solutions into a solution of the whole.

Defining the Termination condition 'SMALL' of the problem , When $n \leq 2$. In this case, the maximum and minimum are $a[i]$ if $n = 1$. If $n = 2$, the problem can be solved by making one comparison.

How can we Combine The Subproblem ? If $\text{MAX}(A)$ and $\text{MIN}(P)$ are the maximum and minimum of the elements of A, then $\text{MAX}(A)$ is the larger of $\text{MAX}(A_1)$ and $\text{MAX}(A_2)$ Similarly for $\text{MIN}(A)$

```
MaxMin(i, j, max, min)
// a[1:n] is a global array. Parameters i and j are integers,
// 1 ≤ i ≤ j ≤ n. The effect is to set max and min to the largest and
// smallest values in a[i:j].
{
    if (i=j) then max := min := a[i]; //Small(P)
    else if (i=j-1) then // Another case of Small(P)
    {
        if (a[i] < a[j]) then max := a[j]; min := a[i];
        else max := a[i]; min := a[j];
    }
    else
```

```

{
    // if P is not small, divide P into sub-problems.
    // Find where to split the set.
    mid := ( i + j ) /2;
    // Solve the sub-problems.
    MaxMin( i, mid, max, min );
    MaxMin( mid+1, j, max1, min1 );
    // Combine the solutions.
    if (max < max1) then max := max1;
    if (min > min1) then min := min1;
}
}

```

Analysis

Time Complexity

$$T(n) = \begin{cases} 0 & n=1 \\ 1 & n=2 \\ 2T\left(\frac{n}{2}\right) + 2 & n>2 \end{cases}$$

Using [Master Theorem](#) Case 1 follows . Hence $T(n)$ is $\Theta(n)$

Solution

$$\begin{aligned}
T(n) &= 2T\left(\frac{n}{2}\right) + 2 \\
&= 4T\left(\frac{n}{4}\right) + 2^2 + 2 \\
&\quad \dots \\
&= 2^k T\left(\frac{n}{2^k}\right) + \sum_{i=1}^k 2^i \quad (\text{where } \sum_{i=1}^k 2^i = 2^{k+1} - 2) \\
&= 2^k T\left(\frac{n}{2^k}\right) + 2^{k+1} - 2 \quad (\text{where, } k = (\log n) - 1) \\
&= 2^{(\log n)-1} T(1) + 2^{\log n} - 2 \\
&= \frac{n}{2} + n - 2 \\
&= \frac{3n}{2} - 2
\end{aligned}$$

For Input Class : Numbers in Increasing Order Non-Divide And Conquer Strategy Perform Better than Divide And Conquer

Space Complexity

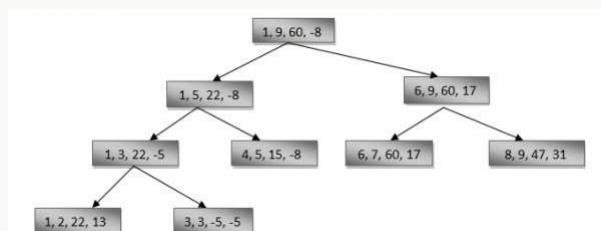
$S(n) = n + \log n + c$ Where n is the input size and $\log n$ is the size of the stack . Hence the space complexity is $O(\log n)$

Example

Consider the array

$$A[] = \{ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \}$$

$$A[] = \{ 22 \ 13 \ -5 \ -8 \ 15 \ 60 \ 17 \ 31 \ 47 \}$$



Video Example : <https://www.youtube.com/watch?v=EHRL2LbS5LU>

References

- <https://www.youtube.com/watch?v=lEvzwEcjQ54&feature=youtu.be&t=1823>

2. [Cormen](#)
 3. <http://somnathkayal.blogspot.in/2012/08/finding-maximum-and-minimum-using.html>

4 votes

-- pC (20.7k points)

1.15.17 Time Complexity: GATE2007-51 [top](#)

<http://gateoverflow.in/1249>



Selected Answer

Answer = **option B**

Worst Case : $T(n) = \mathcal{O}(\sqrt{n})$

Best Case : When n is an even number body of *for* loop is executed only 1 time (due to "return 0" inside if) which is irrespective of n . $\therefore T(n) = \Omega(1)$

14 votes

-- Gate Keeda (18.8k points)

1.15.18 Time Complexity: GATE2007-IT-17 [top](#)

<http://gateoverflow.in/3450>



Selected Answer

Answer is (A)

We need to divide

n recursively and compute like following:

$C_1 = b^{\frac{n}{2}} \times b^{\frac{n}{2}}$. In this, we need to calculate $b^{\frac{n}{2}}$ only once.

$C_2 = b^{\frac{n}{4}} \times b^{\frac{n}{4}}$

:

$$C_k = b^2 \times b^2 \quad \left\{ \begin{array}{l} k = \log n \end{array} \right.$$

Recurrence relation: $T(n) = T\left(\frac{n}{2}\right) + O(1)$

$T(n) = O(\log n)$

18 votes

-- Sandeep_Uniyal (7.3k points)

1.15.19 Time Complexity: GATE2007-IT-81 [top](#)

<http://gateoverflow.in/3533>



Selected Answer

Answer: B

Gradient = $y_2 - y_1 / x_2 - x_1$

For gradient to be maximum $x_2 - x_1$ should be minimum. So, sort the points (in $\Theta(n \log n)$ time) according to x coordinate and find the minimum difference between them (in $\Theta(n)$ time).

Best complexity: $\Theta(n \log n + n)$ which leads to B.

6 votes

-- Rajarshi Sarkar (34.4k points)

1.15.20 Time Complexity: GATE2008-40 [top](#)

<http://gateoverflow.in/452>



answer = **option B**

whenever there exists an element which is present in the array : more than $\frac{n}{2}$ times, then definitely it will be present at the middle index position; in addition to that it will also be present at anyone of the neighbourhood indices namely $i - 1$ and $i + 1$

No matter how we push that stream of **More than** $\frac{n}{2}$ times of elements of same value around the Sorted Array, it is bound to be present at the middle index + atleast anyone of its neighbourhood

once we got the element which should have occurred more than $n/2$ times we [count its total occurrences](#) in $\mathcal{O}(\log n)$ time.

1 21 votes

-- Amar Vashishth (27.9k points)

To check whether a given number is repeated $n/2$ times in the array can be done in $\mathcal{O}(\log n)$ time.

Algo

1. find the **first** occurrence (index i) of x(given number) in the array which can be done in $\mathcal{O}(\log n)$ time (a variant of binary search).

2. check if $A[i] == A[n/2+i]$

return true

3. else return false

1 16 votes

-- Vikrant Singh (13.4k points)

1.15.21 Time Complexity: GATE2008-47 [top](#)

<http://gateoverflow.in/459>



an insert operation on a binary heap takes $\mathcal{O}(\log n)$ time, but an alternative approach we can use. which requires us to insert n elements in heap without any computation i.e. in constant time. after which we can apply Heapify operation(this operation creates heap in linear time) on the array of those element and Hence obtain a Heap in $\mathcal{O}(n)$ time.

Here "not necessairly one after another" should mean that we can insert n elements at once and not necesairly have to wait for first insert to be completed before doing second.

1 16 votes

-- Amar Vashishth (27.9k points)

1.15.22 Time Complexity: GATE2008-74 [top](#)

<http://gateoverflow.in/495>



Q.74 = option B
Q.75 = option C

Time complexity of f1 is given by

$T(n) = T(n-1) + T(n-2)$, (multiplication by 2 and 3 won't affect complexity as it is a constant time operation)
 $T(0) = T(1) = 1$

The solution to this (fibonacci series) is given by Golden ratio. https://en.wikipedia.org/wiki/Golden_ratio which is $O(2^n)$. (Using theta in question must be a mistake)

Time complexity of f_2 is $\Theta(n)$ as here all recursive calls are avoided by saving the results in an array (dynamic programming).

So, answer to 74 is (B).

75. Both f_1 and f_2 are calculating the same function. So,

$$\begin{aligned}f_1(2) &= 2f_1(1) + 3f_1(0) = 2 \\f_1(3) &= 2f_1(2) + 3f_1(1) = 7 \\f_1(4) &= 20 \\f_1(5) &= 61 \\f_1(6) &= 182 \\f_1(7) &= 547 \\f_1(8) &= 1640 = f_2(8)\end{aligned}$$

16 votes

-- Arjun Suresh (287k points)

1.15.23 Time Complexity: GATE2008-75 [top](#)

<http://gateoverflow.in/43489>



Selected Answer

Here Ans C. 1640 and 1640

$f_1(8)$

$$\begin{array}{r} f_1(8) \\ \hline 2 + f_1(7) + 3 * f_1(6) = 1640 \\ \hline 2 + f_1(6) + 3 * f_1(5) = 547 \\ \hline 2 + f_1(5) + 3 * f_1(4) = 182 \\ \hline 2 + f_1(4) + 3 * f_1(3) = 61 \\ \hline 2 + f_1(3) + 3 * f_1(2) = 20 \\ \hline 2 + f_1(2) + 3 * f_1(1) = 7 \\ \hline 2 + f_1(1) + 3 * f_1(0) = 2 \end{array}$$

$$\begin{aligned}f_1(1) &= 1 \\f_1(0) &= 0\end{aligned}$$

$f_2(8)$ will return

X	[0	1	2	7	20	61	182	547	1640]
Y	[0	2	4	14	40	120	364	1094	3280]
Z	[0	3	6	21	60	183	546	1641	4920]
		6	1	3	3	4	5	6	7	8	

return X[8] = 1640

7 votes

-- Manoj Kumar (37k points)

1.15.24 Time Complexity: GATE2010-12 [top](#)

<http://gateoverflow.in/2185>



Selected Answer

$$10n \log_{10} n \leq 0.0001n^2$$

$$\Rightarrow 10 \times 10^k \log_{10} 10^k \leq 0.0001(10^k)^2$$

$$\Rightarrow 10^{k+1} k \leq 0.0001 \times 10^{2k}$$

$$\Rightarrow k \leq 10^{2k-k-4}$$

$$\Rightarrow k \leq 10^{k-5}$$

Trying the values, 5 doesn't satisfy this but 6 satisfies.

18 votes

-- Arjun Suresh (287k points)

1.15.25 Time Complexity: GATE2014-1-42 [top](#)

<http://gateoverflow.in/1920>



Selected Answer

Total number of multiplications

$$= \sum_{i=1}^n \sum_{j=i}^n \sum_{k=j+1}^n 1 = \sum_{i=1}^n \sum_{j=i}^n (n-j) = \sum_{i=1}^n (n-i) + (n-(i+1)) + \dots + (n-n) = \frac{1}{2} \sum_{i=1}^n (n-i)(n-i+1)$$

Therefore, correct answer would be (C).

25 votes

-- suraj (5k points)

1.15.26 Time Complexity: GATE2015-1_40 [top](#)

<http://gateoverflow.in/829>



Selected Answer

	$(\log N)^{1/2}$ find	N insert	$(\log N)^{1/2}$ delete	$(\log N)^{1/2}$ decrease-key
Unsorted Array	$O(N (\log N)^{1/2})$	$O(N)$	$O(\log N)^{1/2})$	$O(\log N)^{1/2})$
Min-heap	$O(N (\log N)^{1/2})$	$O(N \log N)$	$O((\log N)^{3/2})$	$O((\log N)^{3/2})$
Sorted Array	$O((\log N)^{3/2})$	$O(N^2)$	$O(N (\log N)^{1/2})$	$O(N (\log N)^{1/2})$
Sorted doubly linked-list	$O(N (\log N)^{1/2})$	$O(N^2)$	$O((\log N)^{1/2})$	$O(N (\log N)^{1/2})$

So, Unsorted array is the answer.

The operations given can be performed in any order. So, for Min-heap we cannot do the usual BuildHeap method.

Delete in unsorted array is $O(1)$ as we can just swap the deleted element with the last element in the array and delete the last element.

For sorted-doubly linked-list we cannot do binary search as this would require another array to maintain the pointers to the nodes.

32 votes

-- Arjun Suresh (287k points)

1.15.27 Time Complexity: GATE2015-2_22 [top](#)

<http://gateoverflow.in/8113>



Selected Answer

Ans O(1), because all elements are distinct, select any three numbers and output 2nd largest from them.

38 votes

-- Vikrant Singh (13.4k points)

1.15.28 Time Complexity: GATE2017-2-03 [top](#)

<http://gateoverflow.in/118156>



Selected Answer

According to the recurrence relation

$$T(n) = 2 T(n-1) + 1$$

Tower of hanoi we get it is $\Theta(2^n)$

now heap sort worst case $\Theta(n \log n)$

Binary Search given n numbers n sorted numbers $\Theta(\log n)$

Addition of two nxn matrices $\Theta(n^2)$

so C is correct answer here

11 votes

-- Aboveallplayer (18.5k points)

1.15.29 Time Complexity: GATE2017-2-38 [top](#)

<http://gateoverflow.in/118283>



Selected Answer

inner for loop is dependent on i, so for each i we have to check no of times inner loop operating..

it'll be something like

$$\frac{n-1}{1} + \frac{n-1}{2} + \frac{n-1}{3} + \dots + \frac{n-1}{n-1} + 1$$

$$\frac{n}{1} + \frac{n}{2} + \frac{n}{3} + \dots + \frac{n}{n-1} - \log(n-1)$$

$$n\left\{\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n-1}\right\} - \log(n-1)$$

$$n\log(n-1) - \log(n-1)$$

$$n\log(n-1)$$

nlogn

12 votes

-- 2018 (4.9k points)

1.15.30 Time Complexity: TIFR2013-B-12 [top](#)

<http://gateoverflow.in/25774>



Selected Answer

- 1) Calculate the medians m1 and m2 of the input arrays ar1[] and ar2[] respectively.
- 2) If m1 and m2 both are equal.
return m1 (or m2)
- 3) If m1 is greater than m2, then median is present in one of the below two subarrays.
 - a) From first element of ar1 to m1 (ar1[0 to n/2])
 - b) From m2 to last element of ar2 (ar2[n/2 to n-1])
- 4) If m2 is greater than m1, then median is present in one of the below two subarrays.
 - a) From m1 to last element of ar1 (ar1[n/2 to n-1])
 - b) From first element of ar2 to m2 (ar2[0 to n/2])
- 5) Repeat the above process until size of both the subarrays becomes 2.

6) If size of the two arrays is 2 then
the median.

Median = $(\max(ar1[0], ar2[0]) + \min(ar1[1], ar2[1]))/2$
Time complexity $O(\log n)$

<http://www.geeksforgeeks.org/median-of-two-sorted-arrays/>

8 votes

-- Umang Raman (14.5k points)

1.15.31 Time Complexity: TIFR2014-B-7 [top](#)

<http://gateoverflow.in/27189>



Selected Answer

Let us take \log for each function.

1. $\log(\log n)^{\log \log n} = (\log \log n)^2$
2. $\log 2^{\sqrt{\log n}} = \sqrt{\log n} = (\log n)^{0.5}$
3. $\log n^{1/4} = 1/4 \log n$

Here, If we consider $\log n$ as a term (which is common in all 3), first 1 is a log function, second one is sqrt function and third one is linear function of $\log n$. Order of growth of these functions are [well known](#) and \log is the slowest growing followed by sqrt and then linear. So, option A is the correct answer here.

PS: After taking \log we arrive at functions distinguished by some constant terms only, then we can not conclude the order of growth of the original functions using the \log function. Examples are $f(n) = 2^n, g(n) = 3^n$.

5 votes

-- srestha (55.2k points)

1.15.32 Time Complexity: TIFR2015-B-3 [top](#)

<http://gateoverflow.in/29846>



Selected Answer

Ans : C

it is fibanacci series generation. it takes exponential time if we won't use dynamic programming.

if we use dynamic programming then it takes $O(n)$

7 votes

-- pramod (3.2k points)

2

Compiler Design (157) [top](#)

2.1

Abstract Syntax Tree(1) [top](#)

2.1.1 Abstract Syntax Tree: GATE2015-2-14 [top](#)

<http://gateoverflow.in/8084>

In the context of abstract-syntax-tree (AST) and control-flow-graph (CFG), which one of the following is TRUE?

- A. In both AST and CFG, let node N_2 be the successor of node N_1 . In the input program, the code corresponding to N_2 is present after the code corresponding to N_1
- B. For any input program, neither AST nor CFG will contain a cycle
- C. The maximum number of successors of a node in an AST and a CFG depends on the input program
- D. Each node in AST and CFG corresponds to at most one statement in the input program

[gate2015-2](#) [compiler-design](#) [easy](#) [abstract-syntax-tree](#)[Answer](#)

Answers: Abstract Syntax Tree

2.1.1 Abstract Syntax Tree: GATE2015-2-14 [top](#)

<http://gateoverflow.in/8084>

Selected Answer

Option (C) is Correct

- (A) is false , In CFG , code of N_2 may be present before N_1 when there is a loop or Goto.
- (B) is false , CFG contains cycle when input program has loop.
- (C) is true ,successors in AST and CFG depend on Input program.
- (D) is false, In CFG a single node may belong to a block of statements.

[12 votes](#)

-- Himanshu Agarwal (15.9k points)

2.2

Assembler(5) [top](#)

2.2.1 Assembler: GATE1992-01,viii [top](#)

<http://gateoverflow.in/553>

The purpose of instruction location counter in an assembler is _____

[gate1992](#) [compiler-design](#) [assembler](#) [normal](#)[Answer](#)

2.2.2 Assembler: GATE1992_03,ii [top](#)

<http://gateoverflow.in/579>

Mention the pass number for each of the following activities that occur in a two pass assembler:

- a. object code generation
- b. literals added to literal table
- c. listing printed
- d. address resolution of local symbols

[gate1992](#) [compiler-design](#) [assembler](#) [easy](#)[Answer](#)

2.2.3 Assembler: GATE1993-7.6 [top](#)

<http://gateoverflow.in/2294>

A simple two-pass assembler does the following in the first pass:

- A. It allocates space for the literals.
- B. It computes the total length of the program.
- C. It builds the symbol table for the symbols and their values.
- D. It generates code for all the load and store register instructions.
- E. None of the above.

[gate1993](#) [compiler-design](#) [assembler](#) [easy](#)

[Answer](#)

2.2.4 Assembler: GATE1994_18 [top](#)

<http://gateoverflow.in/2514>

State whether the following statements are True or False with reasons for your answer

- a. A subroutine cannot always be used to replace a macro in an assembly language program.
- b. A symbol declared as 'external' in an assembly language program is assigned an address outside the program by the assembler itself.

[gate1994](#) [compiler-design](#) [normal](#) [assembler](#) [true-false](#)

[Answer](#)

2.2.5 Assembler: GATE1996_1.17 [top](#)

<http://gateoverflow.in/2721>

The pass numbers for each of the following activities

- i. object code generation
- ii. literals added to literal table
- iii. listing printed
- iv. address resolution of local symbols that occur in a two pass assembler

respectively are

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

[gate1996](#) [compiler-design](#) [normal](#) [assembler](#)

[Answer](#)

Answers: Assembler

2.2.1 Assembler: GATE1992-01,viii [top](#)

<http://gateoverflow.in/553>

Each section of an assembler language program has a location counter used to assign storage addresses to your program's statements. As the instructions of a source module are being assembled, the location counter keeps track of the current location in storage.

2 votes

-- Rohan Ghosh (1 . 9k points)

2.2.2 Assembler: GATE1992_03,ii [top](#)

<http://gateoverflow.in/579>

Selected Answer

- a) 2
- b) 1
- c) 2
- d) 1

P.S. : In first pass, symbol table is created and In second pass, machine code is generated. Listing of final machine code is done after 2nd pass only.

7 votes

-- Aditya Gaurav (2 . 8k points)

2.2.3 Assembler: GATE1993-7.6 [top](#)

<http://gateoverflow.in/2294>

Selected Answer

a, b, c are TRUE.

http://gateoverflow.in/?qa=blob&qa_blobid=2337905098612945492

14 votes

-- Arjun Suresh (287k points)

2.2.4 Assembler: GATE1994_18 [top](#)

<http://gateoverflow.in/2514>

A) This is true. We can not replace macro entirely using subroutine. Ex -> Macro constant used for renaming.

B) This is false. This is job of Linker.

3 votes

-- Akash (41 . 5k points)

2.2.5 Assembler: GATE1996_1.17 [top](#)

<http://gateoverflow.in/2721>

Selected Answer

Ans should be (B)

the functions performed in pass 1 and pass 2 in 2 pass assembler are

Pass 1

1. Assign addresses to all statements in the program.
2. Save the values assigned to all labels for use in pass 2
3. Perform some processing of assembler directives.

Pass 2

1. Assemble instructions.
2. Generate data values defined by BYTE, WORD etc.
3. Perform processing of assembler directives not done during pass 1.
4. Write the program and the assembling listing

9 votes

-- sonam vyas (12 . 8k points)

2.3**Code Optimization(4)** [top](#)**2.3.1 Code Optimization: GATE2008-12** [top](#)<http://gateoverflow.in/410>

Some code optimizations are carried out on the intermediate code because

- A. They enhance the portability of the compiler to the target processor
- B. Program analysis is more accurate on intermediate code than on machine code
- C. The information from dataflow analysis cannot otherwise be used for optimization
- D. The information from the front end cannot otherwise be used for optimization

[gate2008](#) [normal](#) [code-optimization](#) [compiler-design](#)

[Answer](#)

2.3.2 Code Optimization: GATE2014-1-17 [top](#)<http://gateoverflow.in/1784>

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

- A. A basic block is a sequence of instructions where control enters the sequence at the beginning and exits at the end.
- B. Available expression analysis can be used for common subexpression elimination.
- C. Live variable analysis can be used for dead code elimination.
- D. $x = 4 * 5 \Rightarrow x = 20$ is an example of common subexpression elimination.

[gate2014-1](#) [compiler-design](#) [code-optimization](#) [normal](#)

[Answer](#)

2.3.3 Code Optimization: GATE2014-3-11 [top](#)<http://gateoverflow.in/2045>

The minimum number of arithmetic operations required to evaluate the polynomial $P(X) = X^5 + 4X^3 + 6X + 5$ for a given value of X , using only one temporary variable is _____.

[gate2014-3](#) [compiler-design](#) [numerical-answers](#) [normal](#) [code-optimization](#)

[Answer](#)

2.3.4 Code Optimization: GATE2014-3-34 [top](#)<http://gateoverflow.in/2068>

Consider the basic block given below.

```
a = b + c
c = a + d
d = b + c
e = d - b
a = e + b
```

The minimum number of nodes and edges present in the DAG representation of the above basic block respectively are

- A. 6 and 6
- B. 8 and 10
- C. 9 and 12
- D. 4 and 4

[gate2014-3](#) [compiler-design](#) [code-optimization](#) [normal](#)

[Answer](#)

Answers: Code Optimization**2.3.1 Code Optimization: GATE2008-12** [top](#)<http://gateoverflow.in/410>

Selected Answer

Ans is (A)

Intermediate codes are machine independent codes. So, intermediate code can be used for code optimization since a given source code can be converted to target machine code.

13 votes

-- Keith Kr (6.2k points)

2.3.2 Code Optimization: GATE2014-1-17 [top](#)

<http://gateoverflow.in/1784>



Selected Answer

A) A basic block is a sequence of instructions where control enters the sequence at the beginning and exits at the end is TRUE.

(B) Available expression analysis can be used for common subexpression elimination is TRUE. Available expressions is an analysis algorithm that determines for each point in the program the set of expressions that need not be recomputed. Available expression analysis is used to do global common subexpression elimination (CSE). If an expression is available at a point, there is no need to re-evaluate it.

(C) Live variable analysis can be used for dead code elimination is TRUE.

(D) $x = 4 * 5 \Rightarrow x = 20$ is an example of common subexpression elimination is FALSE. Common subexpression elimination (CSE) refers to compiler optimization replaces identical expressions (i.e., they all evaluate to the same value) with a single variable holding the computed value when it is worthwhile to do so Source: Geeksforgeeks

5 votes

-- Pyuri sahu (2k points)

2.3.3 Code Optimization: GATE2014-3-11 [top](#)

<http://gateoverflow.in/2045>



Selected Answer

$$\begin{aligned} P(X) &= x^5 + 4x^3 + 6x + 5 \\ &= x(x^4 + 4x^2 + 6) + 5 \\ &= x(x(x^3 + 4x) + 6) + 5 \\ &= x(x(x(x^2 + 4)) + 6) + 5 \\ &= x(x(x(x(x) + 4)) + 6) + 5 \end{aligned}$$

mul = pair of brackets 4

add = num of signs 3

total 7

27 votes

-- jayendra (7.7k points)

$P(x) = x^5 + 4x^3 + 6x + 5$ can be re written as follows

$$P(x) = x^3(x^2 + 4) + 6x + 5$$

Now using only one temporary variable t and any number of data transfer as well as memory related operation the polynomial can be evaluated as follows

1. $t = x * x$ [Evaluate x^2 and store in memory]
2. $t = t + 4$ [Evaluate $(x^2 + 4)$ and store in memory]
3. $t = x^2$ [Retrieve x^2 from memory]
4. $t = t * x$ [Evaluate x^3 and store in memory]
5. $t = t * (x^2 + 4)$ [Evaluate $x^3(x^2 + 4)$ and store in memory]
6. $t = 6 * x$ [Evaluate 6x and store in memory]
7. $t = t + 5$ [Evaluate $(6x + 5)$ and store in memory]
8. $t = t + x^3(x^2 + 4)$ [Retrieve $x^3(x^2 + 4)$ from memory and evaluate $\{x^3(x^2 + 4) + 6x + 5\}$]

In the above 8 steps of evaluation, the total number of arithmetic operations required are 7 [4 Multiplications, 3 Additions]
So answer is 7 arithmetic operations.

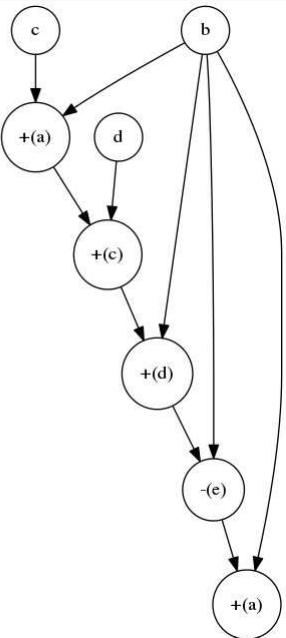
13 votes

-- Gate Keeda (18.8k points)

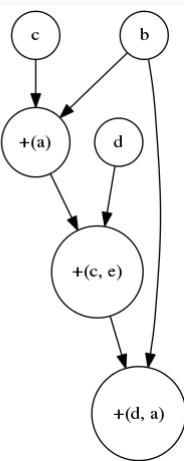
2.3.4 Code Optimization: GATE2014-3-34 [top](#)

<http://gateoverflow.in/2068>

A normal DAG construction will give 8 nodes and 10 edges as shown below.



Since, this question asks for minimum possible, we can assume algebraic simplification is allowed. So, $d = b + c$, $e = d - b$; can be simplified to $d = b + c$; $e = c$; Similarly, $e = d - b$; $a = e + b$; can be simplified to $a = d$. This gives the following DAG with 6 nodes and 6 edges.



<https://cs.nyu.edu/~gottlieb/courses/2000s/2006-07-fall/compilers/lectures/lecture-14.html> [working link]

21 votes

-- Arjun Suresh (287k points)

2.4

Compilation Phases(7) [top](#)

2.4.1 Compilation Phases: CMI2012-A-09 [top](#)

<http://gateoverflow.in/4653>

Consider the following programming errors:

- I. Type mismatch in an expression.
- II. Array index out of bounds.
- III. Use of an uninitialized variable in an expression.

Which of these errors will typically be caught at compile-time by a modern compiler.

- A. I, II and III
- B. I and II
- C. I and III
- D. None of them

[cmi2012](#) [compiler-design](#) [compilation-phases](#)

[Answer](#)

2.4.2 Compilation Phases: GATE 2016-2-19 [top](#)

<http://gateoverflow.in/39548>

Match the following:

- | | |
|-------------------------|---------------------------|
| (P) Lexical analysis | (i) Leftmost derivation |
| (Q) Top down parsing | (ii) Type checking |
| (R) Semantic analysis | (iii) Regular expressions |
| (S) Runtime environment | (iv) Activation records |
- A. $P \leftrightarrow i, Q \leftrightarrow ii, R \leftrightarrow iv, S \leftrightarrow iii$
 B. $P \leftrightarrow iii, Q \leftrightarrow i, R \leftrightarrow ii, S \leftrightarrow iv$
 C. $P \leftrightarrow ii, Q \leftrightarrow iii, R \leftrightarrow i, S \leftrightarrow iv$
 D. $P \leftrightarrow iv, Q \leftrightarrow i, R \leftrightarrow ii, S \leftrightarrow iii$

[gate2016-2](#) [compiler-design](#) [easy](#) [compilation-phases](#)

[Answer](#)

2.4.3 Compilation Phases: GATE1987-1-xi [top](#)

<http://gateoverflow.in/80282>

In a compiler the module which checks every character of the source text is called:

- A. The code generator.
- B. The code optimiser.
- C. The lexical analyser.
- D. The syntax analyser.

[gate1987](#) [compiler-design](#) [compilation-phases](#)

[Answer](#)

2.4.4 Compilation Phases: GATE1990-2-ix [top](#)

<http://gateoverflow.in/84033>

Match the pairs in the following questions:

- | | |
|-----------------------|-------------------------|
| (a) Lexical analysis | (p) DAG's |
| (b) Code optimization | (q) Syntax trees |
| (c) Code generation | (r) Push down automaton |
| (d) Abelian groups | (s) Finite automaton |

[gate1990](#) [match-the-following](#) [compiler-design](#) [compilation-phases](#)

[Answer](#)

2.4.5 Compilation Phases: GATE2009-17 [top](#)

<http://gateoverflow.in/1309>

Match all items in Group 1 with the correct options from those given in Group 2.

Group 1	Group 2
P. Regular Expression	1. Syntax analysis
Q. Pushdown automata	2. Code generation
R. Dataflow analysis	3. Lexical analysis
S. Register allocation	4. Code optimization

- A. P-4, Q-1, R-2, S-3
- B. P-3, Q-1, R-4, S-2
- C. P-3, Q-4, R-1, S-2
- D. P-2, Q-1, R-4, S-3

gate2009 compiler-design easy compilation-phases

Answer

2.4.6 Compilation Phases: GATE2015-2-19 [top](#)

<http://gateoverflow.in/8098>

Match the following:

- | | |
|--------------------------|-------------------------|
| P. Lexical analysis | 1. Graph coloring |
| Q. Parsing | 2. DFA minimization |
| R. Register allocation | 3. Post-order traversal |
| S. Expression evaluation | 4. Production tree |

- A. P-2, Q-3, R-1, S-4
- B. P-2, Q-1, R-4, S-3
- C. P-2, Q-4, R-1, S-3
- D. P-2, Q-3, R-4, S-1

gate2015-2 compiler-design normal compilation-phases

Answer

2.4.7 Compilation Phases: GATE2017-2-05 [top](#)

<http://gateoverflow.in/118592>

Match the following according to input (from the left column) to the compiler phase (in the right column) that processes it:

P. Syntax tree	i. Code generator
Q. Character stream	ii. Syntax analyser
R. Intermediate representation	iii. Semantic analyser
S. Token stream	iv. Lexical analyser

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

gate2017-2 compiler-design compilation-phases

Answer

Answers: Compilation Phases

2.4.1 Compilation Phases: CMI2012-A-09 [top](#)



Type mismatch gives compile time error for statically typed languages (type of variables are determined at compile time) like C.

Use of uninitialized variable can be detected by compiler? Not always- but possible in most cases. There can be cases where there are two branches and a variable is initialized in one but not in other. But, at compile time we can assume that we are only interested in those variables which are uninitialized in all the branches,

Array index out of bounds- can be detected by compiler in some cases. Consider

```
int a[100];
a[200] = 4;
```

A decent C compiler should catch this. But if the index is a variable it becomes more difficult to catch at compile time.

So, answer should be C.

1 votes

-- Arjun Suresh (287k points)

2.4.2 Compilation Phases: GATE 2016-2-19 [top](#)



Option B.

Lexical Analysis phase uses regular expressions.

LMD is involved in top down parsing.

Type checking is done in semantic analysis phase.

Activation records are related to Run Time Environments

13 votes

-- Sharathkumar Anbu (717 points)

2.4.3 Compilation Phases: GATE1987-1-xi [top](#)



lexical analyser phase checks every character of text to identify tokens..

4 votes

-- kirti singh (3.3k points)

2.4.4 Compilation Phases: GATE1990-2-ix [top](#)



- (a) Lexical analysis (s) Finite automaton (Dfa creation for finding token)
- (b) Code optimization (p) DAG's (common subtree minimization.)
- (c) Code generation (q) Syntax trees (one can construct a derivation and from it a parse tree that can be used for code generation)
- (d) Abelian groups (r) Push down automaton (left one)

4 votes

-- Prashant Singh (47.6k points)

2.4.5 Compilation Phases: GATE2009-17 [top](#)



Regular expressions are used in lexical analysis.

Pushdown automata is related to context free grammar which is related to syntax analysis.

Dataflow analysis is done in code optimization.

Register allocation is done in code generation.

Ans B

14 votes

-- Keith Kr (6.2k points)

2.4.6 Compilation Phases: GATE2015-2-19 [top](#)



Answer: C

13 votes

-- Rajarshi Sarkar (34.4k points)

1. Regular expression uses FA & Regular Sets.
2. Expression can be evaluated with postfix Traversals .
3. Register allocation can be modeled by graph coloring.
4. The parser constructs a production tree.

10 votes

-- shekhar chauhan (42.2k points)

2.4.7 Compilation Phases: GATE2017-2-05 [top](#)



Q - IV because Character stream is given as input to lexical analyser.

P - iii Syntax tree is given as input to semantic analyser

R - i Intermediate code given as input to code generator

S -ii Token stream given as input to syntax analyser.

Ans: C

9 votes

-- Arnabi (6k points)

2.5

Expression Evaluation(1) [top](#)

2.5.1 Expression Evaluation: GATE2002-2.19 [top](#)

<http://gateoverflow.in/849>

To evaluate an expression without any embedded function calls

- A. One stack is enough
- B. Two stacks are needed
- C. As many stacks as the height of the expression tree are needed
- D. A Turing machine is needed in the general case

[gate2002](#) [compiler-design](#) [expression-evaluation](#) [easy](#)

Answer

Answers: Expression Evaluation

2.5.1 Expression Evaluation: GATE2002-2.19 [top](#)

<http://gateoverflow.in/849>



Selected Answer

Expression without any calls in it => $1+2*3-4$

Expression with embedded calls => $1 + \text{fun1}(a,b,c) * \text{fun2}(3.4,58) - \text{fun3}(x,yz);$

First we can convert Infix to Postfix using single stack (Using it as operator stack)

Then we can evaluate that expression using Single stack.

6 votes

-- Akash (41.5k points)

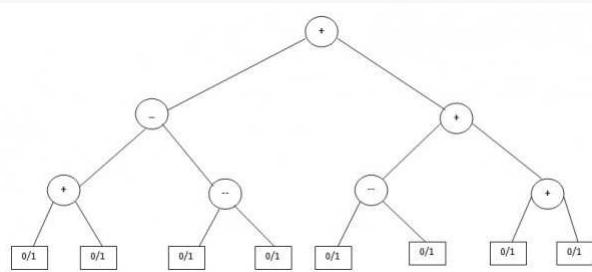
2.6

Expression Tree(1) top

2.6.1 Expression Tree: GATE2014-2-39 top

<http://gateoverflow.in/1999>

Consider the expression tree shown. Each leaf represents a numerical value, which can either be 0 or 1. Over all possible choices of the values at the leaves, the maximum possible value of the expression represented by the tree is ____.



[gate2014-2](#) [compiler-design](#) [normal](#) [expression-tree](#) [numerical-answers](#)

Answer

Answers: Expression Tree

2.6.1 Expression Tree: GATE2014-2-39 top

<http://gateoverflow.in/1999>



Selected Answer

ans is 6

at left leafs

$+ \rightarrow (1,1)=2$ intermediate $+ \rightarrow 2-(-1)=3$

$- \rightarrow (0,1)=-1$

at right leafs

$- \rightarrow (1,0)=1$ intermediate $+ \rightarrow 1+2=3$

$+ \rightarrow (1,1)=2$

at root $+ \rightarrow 3+3=6$

15 votes

-- GANNA (189 points)

2.7

Grammar(39) top

2.7.1 Grammar: GATE 2016-2-45 top

<http://gateoverflow.in/39594>

Which one of the following grammars is free from left recursion?

(A) $S \rightarrow AB$

$$A \rightarrow Aa|b$$

$$B \rightarrow c$$

(B) $S \rightarrow Ab|Bb|c$

$$A \rightarrow Bd|\epsilon$$

$$B \rightarrow e$$

(C) $S \rightarrow Aa|B$

$$A \rightarrow Bb|Sc|\epsilon$$

$$B \rightarrow d$$

(D) $S \rightarrow Aa|Bb|c$

$$A \rightarrow Bd|\epsilon$$

$$B \rightarrow Ae|\epsilon$$

gate2016-2 compiler-design grammar easy

[Answer](#)

2.7.2 Grammar: GATE 2016-2-46 [top](#)

<http://gateoverflow.in/39598>

A student wrote two context-free grammars G1 and G2 for generating a single C-like array declaration. The dimension of the array is at least one. For example,

```
int a[10] [3];
```

The grammars use D as the start symbol, and use six terminal symbols **int** ; **id** [] **num**.

Grammar **G1**

$$D \rightarrow \text{int } L;$$

$$L \rightarrow \text{id } [E]$$

$$E \rightarrow \text{num }$$

$$E \rightarrow \text{num }] [E]$$

Grammar**G2**

$$D \rightarrow \text{int } L;$$

$$L \rightarrow \text{id } E$$

$$E \rightarrow E \text{ [num]}$$

$$E \rightarrow [\text{num}]$$

Which of the grammars correctly generate the declaration mentioned above?

- A. Both **G1** and **G2**
- B. Only **G1**
- C. Only **G2**
- D. Neither **G1** nor **G2**

gate2016-2 compiler-design grammar normal

[Answer](#)

2.7.3 Grammar: GATE1990-16a [top](#)

<http://gateoverflow.in/86869>

Show that grammar **G1** is ambiguous using parse trees:

$G_1 : S \rightarrow \text{if } S \text{ then } S \text{ else } S$

$S \rightarrow \text{if } S \text{ then } S$

gate1990 descriptive compiler-design grammar

Answer

2.7.4 Grammar: GATE1991-10a [top](#)

<http://gateoverflow.in/537>

Consider the following grammar for arithmetic expressions using binary operators — and / which are not associative

$E \rightarrow E - T \mid T$

$T \rightarrow T/F \mid F$

$F \rightarrow (E) \mid id$

(E is the start symbol)

Is the grammar unambiguous? Is so, what is the relative precedence between — and /? If not, give an unambiguous grammar that gives / precedence over —.

gate1991 grammar compiler-design normal descriptive

Answer

2.7.5 Grammar: GATE1991-10b [top](#)

<http://gateoverflow.in/43604>

Consider the following grammar for arithmetic expressions using binary operators — and / which are not associative

$E \rightarrow E - T \mid T$

$T \rightarrow T/F \mid F$

$F \rightarrow (E) \mid id$

(E is the start symbol)

Does the grammar allow expressions with redundant parentheses as in (id/id) or in $id - (id/id)$? If so, convert the grammar into one which does not generate expressions with redundant parentheses. Do this with minimum number of changes to the given production rules and adding at most one more production rule.

gate1991 grammar compiler-design normal descriptive

Answer

2.7.6 Grammar: GATE1991-10c [top](#)

<http://gateoverflow.in/43605>

Consider the following grammar for arithmetic expressions using binary operators — and / which are not associative

$E \rightarrow E - T \mid T$

$T \rightarrow T/F \mid F$

$F \rightarrow (E) \mid id$

(E is the start symbol)

Does the grammar allow expressions with redundant parentheses as in (id/id) or in $id - (id/id)$? If so, convert the grammar into one which does not generate expressions with redundant parentheses. Do this with minimum number of changes to the given production rules and adding at most one more production rule.

Convert the grammar obtained above into one that is not left recursive.

gate1991 grammar compiler-design normal descriptive

Answer

2.7.7 Grammar: GATE1994_1.18 [top](#)

<http://gateoverflow.in/2461>

Which of the following features cannot be captured by context-free grammars?

- A. Syntax of if-then-else statements
- B. Syntax of recursive procedures
- C. Whether a variable has been declared before its use
- D. Variable names of arbitrary length

[gate1994](#) [compiler-design](#) [grammar](#) [normal](#)

[Answer](#)

2.7.8 Grammar: GATE1994_20 [top](#)

<http://gateoverflow.in/2516>

A grammar G is in Chomsky-Normal Form (CNF) if all its productions are of the form $A \rightarrow BC$ or $A \rightarrow a$, where A, B and C , are non-terminals and a is a terminal. Suppose G is a CFG in CNF and w is a string in $L(G)$ of length n , then how long is a derivation of w in G ?

[gate1994](#) [compiler-design](#) [grammar](#) [normal](#)

[Answer](#)

2.7.9 Grammar: GATE1994_3.5 [top](#)

<http://gateoverflow.in/2482>

Match the following items

(i) Backus-Naur form	(a) Regular expressions
(ii) Lexical analysis	(b) LALR(1) grammar
(iii) YACC	(c) LL(1) grammars
(iv) Recursive descent parsing	(d) General context-free grammars

[gate1994](#) [compiler-design](#) [grammar](#) [normal](#)

[Answer](#)

2.7.10 Grammar: GATE1995_1.10 [top](#)

<http://gateoverflow.in/2597>

Consider a grammar with the following productions

$$S \rightarrow aab \mid bac \mid aB$$

$$S \rightarrow \alpha S \mid b$$

$$S \rightarrow abb \mid ab$$

$$S\alpha \rightarrow bdb \mid b$$

The above grammar is:

- A. Context free
- B. Regular
- C. Context sensitive
- D. LR(k)

[gate1995](#) [compiler-design](#) [grammar](#) [normal](#)

[Answer](#)

2.7.11 Grammar: GATE1995_9 [top](#)

<http://gateoverflow.in/2644>

- a. Translate the arithmetic expression $a^* - (b + c)$ into syntax tree.

- b. A grammar is said to have cycles if it is the case that

$$A \Rightarrow^+ A$$

Show that no grammar that has cycles can be LL(1).

[gate1995](#) [compiler-design](#) [grammar](#) [normal](#)

[Answer](#)

2.7.12 Grammar: GATE1996_11 [top](#)

<http://gateoverflow.in/2763>

Let G be a context-free grammar where $G = (\{S, A, B, C\}, \{a, b, d\}, P, S)$ with the productions in P given below.

$$S \rightarrow ABAC$$

$$A \rightarrow aA \mid \epsilon$$

$$B \rightarrow bB \mid \epsilon$$

$$C \rightarrow d$$

(ϵ denotes the null string). Transform the grammar G to an equivalent context-free grammar G' that has no ϵ productions and no unit productions. (A unit production is of the form $x \rightarrow y$, and x and y are non terminals).

[gate1996](#) [compiler-design](#) [grammar](#) [normal](#)

[Answer](#)

2.7.13 Grammar: GATE1996_2.10 [top](#)

<http://gateoverflow.in/2739>

The grammar whose productions are

-> if id then <stmt>

-> if id then <stmt> else <stmt>

-> id := id

is ambiguous because

(a) the sentence

```
if a then if b then c:= d
```

has more than two parse trees

(b) the left most and right most derivations of the sentence

```
if a then if b then c:= d
```

give rise to different parse trees

(c) the sentence

```
if a then if b then c:= d else c:= f
```

has more than two parse trees

(d) the sentence

```
if a then if b then c:= d else c:= f
```

has two parse trees

[gate1996](#) [compiler-design](#) [grammar](#) [normal](#)

[Answer](#)

2.7.14 Grammar: GATE1997_1.6 [top](#)

<http://gateoverflow.in/2222>

In the following grammar

$$\begin{aligned} X &::= X \oplus Y \mid Y \\ Y &::= Z * Y \mid Z \\ Z &\quad ::= id \end{aligned}$$

Which of the following is true?

- a. ' \oplus ' is left associative while '*' is right associative
- b. Both ' \oplus ' and '*' are left associative
- c. ' \oplus ' is right associative while '*' is left associative
- d. None of the above

[gate1997](#) [compiler-design](#) [grammar](#) [normal](#)

[Answer](#)

2.7.15 Grammar: GATE1997_11 [top](#)

<http://gateoverflow.in/2271>

Consider the grammar

$$\begin{aligned} S &\rightarrow bSe \\ S &\rightarrow PQR \\ P &\rightarrow bPc \\ P &\rightarrow \epsilon \\ Q &\rightarrow cQd \\ Q &\rightarrow \epsilon \\ R &\rightarrow dRe \\ R &\rightarrow \epsilon \end{aligned}$$

where S, P, Q, R are non-terminal symbols with S being the start symbol; b, c, d, e are terminal symbols and ' ϵ ' is the empty string. This grammar generates strings of the form $b^i c^j d^k e^m$ for some $i, j, k, m \geq 0$.

- a. What is the condition on the values of i, j, k, m ?
- b. Find the smallest string that has two parse trees.

[gate1997](#) [compiler-design](#) [grammar](#) [normal](#) [theory-of-computation](#)

[Answer](#)

2.7.16 Grammar: GATE1998_6b [top](#)

<http://gateoverflow.in/1697>

Consider the grammar

- $S \rightarrow Aa \mid b$
- $A \rightarrow Ac \mid Sd \mid \epsilon$

Construct an equivalent grammar with no left recursion and with minimum number of production rules.

[gate1998](#) [compiler-design](#) [grammar](#) [descriptive](#)

[Answer](#)

2.7.17 Grammar: GATE1999_2.15 [top](#)

<http://gateoverflow.in/1493>

A grammar that is both left and right recursive for a non-terminal, is

- A. Ambiguous

- B. Unambiguous
- C. Information is not sufficient to decide whether it is ambiguous or unambiguous
- D. None of the above

[gate1999](#) [compiler-design](#) [grammar](#) [normal](#)

[Answer](#)

2.7.18 Grammar: GATE2000-2.21, ISRO2015-24 [top](#)

<http://gateoverflow.in/668>

Given the following expression grammar:

$$E \rightarrow E * F \mid F + E \mid F$$

$$F \rightarrow F - F \mid id$$

Which of the following is true?

- A. * has higher precedence than +
- B. — has higher precedence than *
- C. + and — have same precedence
- D. + has higher precedence than *

[gate2000](#) [grammar](#) [normal](#) [compiler-design](#) [isro2015](#)

[Answer](#)

2.7.19 Grammar: GATE2001-1.18 [top](#)

<http://gateoverflow.in/711>

Which of the following statements is false?

- A. An unambiguous grammar has same leftmost and rightmost derivation
- B. An LL(1) parser is a top-down parser
- C. LALR is more powerful than SLR
- D. An ambiguous grammar can never be LR(k) for any k

[gate2001](#) [compiler-design](#) [grammar](#) [normal](#)

[Answer](#)

2.7.20 Grammar: GATE2001-18 [top](#)

<http://gateoverflow.in/759>

- a. Remove left-recursion from the following grammar: $S \rightarrow Sa \mid Sb \mid a \mid b$
- b. Consider the following grammar:

$$S \rightarrow aSbS \mid bSaS \mid \epsilon$$

Construct all possible parse trees for the string abab. Is the grammar ambiguous?

[gate2001](#) [compiler-design](#) [grammar](#) [descriptive](#)

[Answer](#)

2.7.21 Grammar: GATE2003-56 [top](#)

<http://gateoverflow.in/944>

Consider the grammar shown below

$$S \rightarrow i E t S S' \mid a$$

$$S' \rightarrow e S \mid \epsilon$$

$$E \rightarrow b$$

In the predictive parse table, M, of this grammar, the entries $M[S', e]$ and $M[S', \$]$ respectively are

- A. $\{S' \rightarrow e S\}$ and $\{S' \rightarrow \epsilon\}$

- B. $\{S' \rightarrow e S\}$ and $\{ \}$
 C. $\{S' \rightarrow \epsilon\}$ and $\{S' \rightarrow \epsilon\}$
 D. $\{S' \rightarrow e S, S' \rightarrow \epsilon\}$ and $\{S' \rightarrow \epsilon\}$

gate2003 compiler-design grammar normal

Answer

2.7.22 Grammar: GATE2003-58 [top](#)

<http://gateoverflow.in/946>

Consider the translation scheme shown below.

$$S \rightarrow T R$$

$$R \rightarrow + T \{print('+');\} R | \epsilon$$

$$T \rightarrow \text{num} \{print(\text{num}.val);\}$$

Here **num** is a token that represents an integer and **num.val** represents the corresponding integer value. For an input string '9 + 5 + 2', this translation scheme will print

- A. 9 + 5 + 2
 B. 9 5 + 2 +
 C. 9 5 2 + +
 D. + + 9 5 2

gate2003 compiler-design grammar normal

Answer

2.7.23 Grammar: GATE2004-45 [top](#)

<http://gateoverflow.in/1042>

Consider the grammar with the following translation rules and E as the start symbol

$$\begin{aligned} & \{E.\text{value} = E_1 \\ & \quad .\text{value} * T.\text{value}\} \\ & \{E.\text{value} = T \\ & \quad .\text{value}\} \\ E \rightarrow & E_1 \# T \\ & | T \\ T \rightarrow & T_1 \& F \\ & | F \\ F \rightarrow & \text{num} \\ & .\text{value} \\ & \{T.\text{value} = T_1 \\ & \quad .\text{value} + F.\text{value}\} \\ & \{T.\text{value} = F \\ & \quad .\text{value}\} \\ & \{F.\text{value} = \text{num} \\ & \quad .\text{value}\} \end{aligned}$$

Compute E.value for the root of the parse tree for the expression: 2 # 3 & 5 # 6 & 4

- A. 200
 B. 180
 C. 160
 D. 40

gate2004 compiler-design grammar normal

Answer

2.7.24 Grammar: GATE2004-8 [top](#)

<http://gateoverflow.in/1005>

Which of the following grammar rules violate the requirements of an operator grammar? P, Q, R are nonterminals, and r, s, t are terminals.

- I. $P \rightarrow Q R$
 II. $P \rightarrow Q s R$
 III. $P \rightarrow \epsilon$
 IV. $P \rightarrow Q t R r$

- A. (I) only
 B. (I) and (III) only
 C. (II) and (III) only
 D. (III) and (IV) only

[gate2004](#) [compiler-design](#) [grammar](#) [normal](#)

[Answer](#)

2.7.25 Grammar: GATE2004-88 [top](#)

<http://gateoverflow.in/1082>

Consider the following grammar G:

$$S \rightarrow bS \mid aA \mid b$$

$$A \rightarrow bA \mid aB$$

$$B \rightarrow bB \mid aS \mid a$$

Let $N_a(w)$ and $N_b(w)$ denote the number of a's and b's in a string w respectively.

The language $L(G)$ over $\{a, b\}^+$ generated by G is

- A. $\{w \mid N_a(w) > 3N_b(w)\}$
 B. $\{w \mid N_b(w) > 3N_a(w)\}$
 C. $\{w \mid N_a(w) = 3k, k \in \{0, 1, 2, \dots\}\}$
 D. $\{w \mid N_b(w) = 3k, k \in \{0, 1, 2, \dots\}\}$

[gate2004](#) [compiler-design](#) [grammar](#) [normal](#)

[Answer](#)

2.7.26 Grammar: GATE2005-59 [top](#)

<http://gateoverflow.in/1382>

Consider the grammar:

$$E \rightarrow E + n \mid E \times n \mid n$$

For a sentence $n + n \times n$, the handles in the right-sentential form of the reduction are:

- A. $n, E + n$ and $E + n \times n$
 B. $n, E + n$ and $E + E \times n$
 C. $n, n + n$ and $n + n \times n$
 D. $n, E + n$ and $E \times n$

[gate2005](#) [compiler-design](#) [grammar](#) [normal](#)

[Answer](#)

2.7.27 Grammar: GATE2006-32, ISRO2016-35 [top](#)

<http://gateoverflow.in/995>

Consider the following statements about the context free grammar

$$G = \{S \rightarrow SS, S \rightarrow ab, S \rightarrow ba, S \rightarrow \epsilon\}$$

- I. G is ambiguous
 II. G produces all strings with equal number of a 's and b 's
 III. G can be accepted by a deterministic PDA.

Which combination below expresses all the true statements about G ?

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

[gate2006](#) [compiler-design](#) [grammar](#) [normal](#) [isro2016](#)
Answer**2.7.28 Grammar: GATE2006-59** [top](#)<http://gateoverflow.in/1837>

Consider the following translation scheme.

$$S \rightarrow ER$$

$$R \rightarrow^* E \{ \text{print}('*'); \} R \mid \epsilon$$

$$E \rightarrow F + E \{ \text{print}('+'); \} \mid F$$

$$F \rightarrow S \mid id \{ \text{print}(id.value); \}$$

Here id is a token that represents an integer and $id.value$ represents the corresponding integer value. For an input ' $2 * 3 + 4$ ', this translation scheme prints

- A. $2 * 3 + 4$
- B. $2 * +3 4$
- C. $2 3 * 4 +$
- D. $2 3 4 + *$

[gate2006](#) [compiler-design](#) [grammar](#) [normal](#)
Answer**2.7.29 Grammar: GATE2006-84** [top](#)<http://gateoverflow.in/1856>

Which one of the following grammars generates the language $L = \{a^i b^j \mid i \neq j\}$?

A. $S \rightarrow AC \mid CB$

$$C \rightarrow aCb \mid a \mid b$$

$$A \rightarrow aA \mid \epsilon$$

$$B \rightarrow Bb \mid \epsilon$$

B. $S \rightarrow aS \mid Sb \mid a \mid b$

C. $S \rightarrow AC \mid CB$

$$C \rightarrow aCb \mid \epsilon$$

$$A \rightarrow aA \mid \epsilon$$

$$B \rightarrow Bb \mid \epsilon$$

D. $S \rightarrow AC \mid CB$

$$C \rightarrow aCb \mid \epsilon$$

$$A \rightarrow aA \mid a$$

$$B \rightarrow Bb \mid b$$

[gate2006](#) [compiler-design](#) [grammar](#) [normal](#) [theory-of-computation](#)
Answer**2.7.30 Grammar: GATE2006-85** [top](#)<http://gateoverflow.in/79801>

The grammar

$$S \rightarrow AC \mid CB$$

$$C \rightarrow aCb \mid \epsilon$$

$A \rightarrow aA \mid a$

$B \rightarrow Bb \mid b$

generates the language $L = \{a^i b^j \mid i \neq j\}$. In that grammar what is the length of the derivation (number of steps starting from S) to generate the string $a^l b^m$ with $l \neq m$

- A. $\max(l, m) + 2$
- B. $l + m + 2$
- C. $l + m + 3$
- D. $\max(l, m) + 3$

gate2006 compiler-design grammar normal

Answer

2.7.31 Grammar: GATE2007-52 [top](#)

<http://gateoverflow.in/1250>

Consider the grammar with non-terminals $N = \{S, C, S_1\}$, terminals $T = \{a, b, i, t, e\}$, with S as the start symbol, and the following set of rules:

$S \rightarrow iCtSS_1 \mid a$

$S_1 \rightarrow eS \mid \epsilon$

$C \rightarrow b$

The grammar is NOT LL(1) because:

- A. it is left recursive
- B. it is right recursive
- C. it is ambiguous
- D. it is not context-free

gate2007 compiler-design grammar normal

Answer

2.7.32 Grammar: GATE2007-53 [top](#)

<http://gateoverflow.in/1251>

Consider the following two statements:

- P: Every regular grammar is LL(1)
- Q: Every regular set has a LR(1) grammar

Which of the following is **TRUE**?

- A. Both P and Q are true
- B. P is true and Q is false
- C. P is false and Q is true
- D. Both P and Q are false

gate2007 compiler-design grammar normal

Answer

2.7.33 Grammar: GATE2007-78 [top](#)

<http://gateoverflow.in/1272>

Consider the CFG with

$\{S, A, B\}$ as the non-terminal alphabet,

$\{a, b\}$ as the terminal alphabet, S as the start symbol and the following set of production rules:

$S \rightarrow aB \quad S \rightarrow bA$

$B \rightarrow b \quad A \rightarrow a$

$B \rightarrow bS \quad A \rightarrow aS$
 $B \rightarrow aBB \quad S \rightarrow bAA$

Which of the following strings is generated by the grammar?

- A. aaaabb
- B. aabbbb
- C. aabbab
- D. abbbba

gate2007 | compiler-design | grammar | normal

Answer

2.7.34 Grammar: GATE2007-79 [top](#)

<http://gateoverflow.in/43512>

Consider the CFG with $\{S, A, B\}$ as the non-terminal alphabet, $\{a, b\}$ as the terminal alphabet, S as the start symbol and the following set of production rules:

$S \rightarrow aB \quad S \rightarrow bA$
 $B \rightarrow b \quad A \rightarrow a$
 $B \rightarrow bS \quad A \rightarrow aS$
 $B \rightarrow aBB \quad S \rightarrow bAA$

For the string *aabbab*, how many derivation trees are there?

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

gate2007 | compiler-design | grammar | normal

Answer

2.7.35 Grammar: GATE2007-IT-9 [top](#)

<http://gateoverflow.in/3442>

Consider an ambiguous grammar G and its disambiguated version D . Let the language recognized by the two grammars be denoted by $L(G)$ and $L(D)$ respectively. Which one of the following is true?

- A. $L(D) \subset L(G)$
- B. $L(D) \supset L(G)$
- C. $L(D) = L(G)$
- D. $L(D)$ is empty

gate2007-it | compiler-design | grammar | normal

Answer

2.7.36 Grammar: GATE2008-50 [top](#)

<http://gateoverflow.in/395>

Which of the following statements are true?

- I. Every left-recursive grammar can be converted to a right-recursive grammar and vice-versa
 - II. All ϵ -productions can be removed from any context-free grammar by suitable transformations
 - III. The language generated by a context-free grammar all of whose productions are of the form $X \rightarrow w$ or $X \rightarrow wY$ (where, w is a string of terminals and Y is a non-terminal), is always regular
 - IV. The derivation trees of strings generated by a context-free grammar in Chomsky Normal Form are always binary trees
- A. I, II, III and IV
 - B. II, III and IV only
 - C. I, III and IV only
 - D. I, II and IV only

gate2008 | normal | compiler-design | grammar

Answer

2.7.37 Grammar: GATE2010-38 [top](#)

<http://gateoverflow.in/2339>

The grammar $S \rightarrow aSa \mid bS \mid c$ is

- A. LL(1) but not LR(1)
- B. LR(1) but not LL(1)
- C. Both LL(1) and LR(1)
- D. Neither LL(1) nor LR(1)

[gate2010](#) [compiler-design](#) [grammar](#) [normal](#)

[Answer](#)

2.7.38 Grammar: GATE2014-2-17 [top](#)

<http://gateoverflow.in/1973>

Consider the grammar defined by the following production rules, with two operators * and +

- $S \rightarrow T * P$
- $T \rightarrow U \mid T * U$
- $P \rightarrow Q + P \mid Q$
- $Q \rightarrow Id$
- $U \rightarrow Id$

Which one of the following is TRUE?

- A. + is left associative, while * is right associative
- B. + is right associative, while * is left associative
- C. Both + and * are right associative
- D. Both + and * are left associative

[gate2014-2](#) [compiler-design](#) [grammar](#) [normal](#)

[Answer](#)

2.7.39 Grammar: GateBook Mock Test_2(Compilers) [top](#)

<http://gateoverflow.in/116952>

Consider these three grammars.

Grammar G1:	Grammar G2:	Grammar G3:
$E \rightarrow E + T \mid T$	$E \rightarrow E + T \mid T$	$E \rightarrow T R \mid R$
$T \rightarrow T * v \mid v$	$T \rightarrow v R$	$R \rightarrow + T R \mid \epsilon$

Which of the following statements is not true?

- (A) If w can be generated by G1, then it can be generated by G2.
- (B) If w can be generated by G2, then it can be generated by G3.
- (C) If w can be generated by G3, then it can be generated by G1.
- (D) If w can be generated by G2, then it can be generated by G1.

[gatebook_mt2](#) [compiler-design](#) [grammar](#)

[Answer](#)

Answers: Grammar

2.7.1 Grammar: GATE 2016-2-45 [top](#)

<http://gateoverflow.in/39594>



Selected Answer

Option (a) has immediate left recursion. " $A \rightarrow Aa$ "

Option (c) has indirect left recursion " $S \rightarrow Aa \xrightarrow{A \rightarrow Sc} Sca$ "

Option (d) has indirect left recursion " $A \rightarrow Bd \xrightarrow{B \rightarrow Ae} Aed$ "

Option (b) is free from left recursion. No direct left recursion. No indirect left recursion.

option B is correct

15 votes

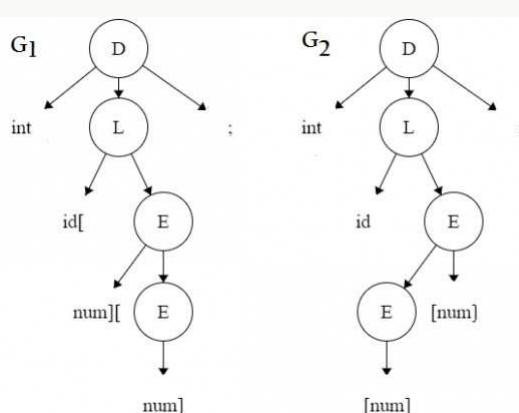
-- Ashish Deshmukh (1.5k points)

2.7.2 Grammar: GATE 2016-2-46 [top](#)

<http://gateoverflow.in/39598>



Option A Both G1 and G2



20 votes

-- Shashank Chavan (3.3k points)

2.7.3 Grammar: GATE1990-16a [top](#)

<http://gateoverflow.in/86869>

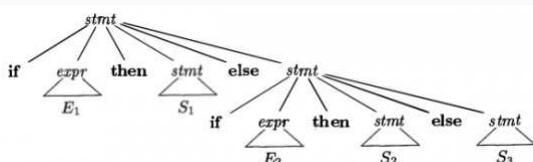


The given grammar is well known as "Dangling Else" problem . The given grammar is ambiguous and ambiguity can be resolved .

stmt → **if expr then stmt**
 | **if expr then stmt else stmt**
 | **other**

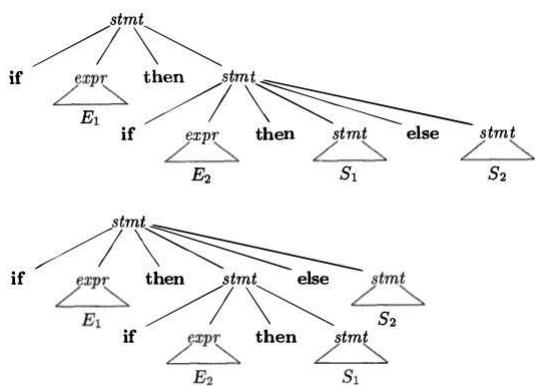
Consider the compound conditional statement for the above grammar

if E1 then S1 else if E2 then S2 else S3 has the following parse tree



Well this is ambiguous due to the statement
if E1 then if E2 then S1 else S2

The two parse trees are



Ambiguity can be resolved by parsers as follows

In all programming languages with conditional statements of this form, the first parse tree is preferred.
The general rule is, "Match each else with the closest unmatched then"

In practice it is rarely built into the productions .

```

stmt → matched_stmt
      | open_stmt
matched_stmt → if expr then matched_stmt else matched_stmt
      | other
open_stmt → if expr then stmt
           | if expr then matched_stmt else open_stmt

```

However this grammar is also ambiguous .

! Exercise 4.3.3: The following grammar is proposed to remove the “dangling-else ambiguity” discussed in Section 4.3.2:

```

stmt      → if expr then stmt
          | matchedStmt
matchedStmt → if expr then matchedStmt else stmt
          | other

```

Show that this grammar is still ambiguous.

Solution is Given [Here](#) . Moreover there is no grammar exists that will **eliminate** ambiguity in Dangling Else Grammar . Hence we can say that Language produced by dangling else grammar is " **inherently ambiguous** " .

Try running this code in any compilers . No doubt that all compilers will successfully parse and produce output

```
#include <stdio.h>

int main(void) {
    if(1<3)
        if(1<2)
            printf("1 is the smallest");
        else
            printf("2 is the smallest");
    return 0;
}
```

Thus we can say that ambiguity in Dangling Else Problem can be resolved but Cannot be Eliminated from the grammar .

Source

- [Compilers: Principles, Techniques, & Tools -Aho & Ullman](#)

Good Read

- <http://cs.stackexchange.com/questions/32475/resolving-ambiguity-in-dangling-else>
- <http://gateoverflow.in/54718/inherently-ambiguous-languages-deterministic-context-grammars>
- <http://gateoverflow.in/84451/dangling-else-problem-and-ambiguity-elimination>

6 votes

-- pC (20.7k points)

2.7.4 Grammar: GATE1991-10a [top](#)

<http://gateoverflow.in/537>



Selected Answer

Yes. It is unambiguous grammar since for any string no more than 1 parse tree is possible.

For precedence draw the parse tree and find the depth of operator- and / .

Here "/" having more depth than " - " operator so precedence of" /" is higher than "-".

3 votes

-- papesh (22.8k points)

2.7.5 Grammar: GATE1991-10b [top](#)

<http://gateoverflow.in/43604>

If the expression with redundant parentheses is (id/id) or id-(id/id) then it can be generated by the given grammar.

To generate expression (id/id) we can go through following steps-

1. E-> T
2. E-> F (T-> F)
3. E-> (E) (F-> (E))
4. E-> (T) (E->T)
5. E-> (T/F) (T-> T/F)
6. E-> (F/F) (T->F)
7. E-> (id/id) (F->id)

Now to generate expression id- (id/id) we can go through following steps-

1. E-> E - T
2. E-> E - F (T-> F)
3. E-> E - (E) (F-> (E))
4. E-> E - (T) (E->T)
5. E-> E - (T/F) (T-> T/F)
6. E-> E - (F/F) (T->F)
7. E-> T - (F/F) (E-> T)
8. E-> F - (F/F) (T->F)
9. E-> id - (id/id) (F->id)

3 votes

-- Dhananjay Kumar Sharma (24.9k points)

2.7.6 Grammar: GATE1991-10c [top](#)

<http://gateoverflow.in/43605>



Selected Answer

Here we have to convert the grammar into one which does not generate expressions with redundant parentheses. So the grammar which does not generate expressions with redundant parentheses

$$E \rightarrow E - T \mid T$$

$$E' \rightarrow E - T$$

$$T \rightarrow id \mid T'/F$$

$$T' \rightarrow F \mid T'/F$$

$$F \rightarrow id \mid (E')$$

Now equivalent grammar after removing left recursion

$$E \rightarrow TX'$$

$$X' \rightarrow -TX' \mid \epsilon$$

$$E' \rightarrow E - T$$

$$T \rightarrow id \mid T'/F$$

$$T' \rightarrow FY'$$

$$Y' \rightarrow /FY' \mid \epsilon$$

$$F \rightarrow id \mid (E')$$

7 votes

-- Manoj Kumar (37k points)

2.7.7 Grammar: GATE1994_1.18 [top](#)

<http://gateoverflow.in/2461>



Selected Answer

It will be C.

Since CFG's are used to show syntactic rules while designing compiler, and syntactic rules don't check for meaningful things such as if a variable has been declared before its use or not. Such things are meant to be handled by Semantic Analysis phase (requires power of a context sensitive grammar).

For D, a regular expression does not restrict the string length. Languages have restriction for variable name length for storing purpose like in symbol table.

For A, if then else is inherently ambiguous. But CFG can represent inherently ambiguous languages just that there are more than one parse trees possible for some strings.

16 votes

-- Gate Keeda (18.8k points)

2.7.8 Grammar: GATE1994_20 [top](#)

<http://gateoverflow.in/2516>



Selected Answer

its answer is $2n-1$ for n length string, because in CNF at every step only 1 terminal can replace a variable, for example
 S-AB
 A-a
 B-c

for generating string 'ac' 3 production will be used.

14 votes

-- Manu Thakur (5.8k points)

2.7.9 Grammar: GATE1994_3.5 [top](#)

<http://gateoverflow.in/2482>



Selected Answer

answer -

i - d

ii - a

iii - b

iv - c

5 votes

-- ankitrokdeonsns (9.1k points)

2.7.10 Grammar: GATE1995_1.10 [top](#)

<http://gateoverflow.in/2597>



Selected Answer

$S \propto \rightarrow$

This violates the conditions of context-free and hence the grammar becomes context-sensitive.

13 votes

-- Arjun Suresh (287k points)

2.7.11 Grammar: GATE1995_9 [top](#)

<http://gateoverflow.in/2644>

A grammar having left recursion generates a cycle.

And no left recursive grammar is LL(1) grammar.

5 votes

-- simran (87 points)

2.7.12 Grammar: GATE1996_11 [top](#)

<http://gateoverflow.in/2763>



final grammar is

$S \rightarrow ABAC / ABC / BAC / BC / AC / AAC / d$

$A \rightarrow aA / a$

$B \rightarrow bB / b$

$C \rightarrow d$

correct me

15 votes

-- Mithlesh Upadhyay (5.3k points)

2.7.13 Grammar: GATE1996_2.10 [top](#)

<http://gateoverflow.in/2739>



(d) the sentence

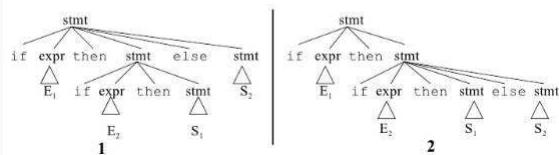
- if a then if b then c:= d else c:= f has two parse trees as follows:

- if a then (if b then c:= d) else c:= f
- and if a then (if b then c:=d else c:= f)

Ambiguity – “dangling else ”

$\text{stmt} \rightarrow \text{if expr then stmt} \mid \text{if expr then stmt else stmt} \mid \text{other stmts}$

if E_1 then if E_2 then S_1 else S_2



12 votes

-- Arjun Suresh (287k points)

2.7.14 Grammar: GATE1997_1.6 [top](#)

<http://gateoverflow.in/2222>



Selected Answer

It will be A. For multiple ' \oplus ', the derivation is possible only via ' X ' which is on left side of ' \oplus ' in the production. Hence it is left associative.

For multiple '*', the derivation is possible only via ' Y ' which is on the right side of '*' in the production. Hence it is right associative.

If both left and right derivations were possible, the grammar would have been ambiguous and we couldn't have given associativity.

16 votes

-- Gate Keeda (18.8k points)

2.7.15 Grammar: GATE1997_11 top

<http://gateoverflow.in/2271>



- A) $i+k=j+m$
where $i,j,k,m \geq 0$
B) bcde

6 votes

-- Danish (3.5k points)

2.7.16 Grammar: GATE1998_6b top

<http://gateoverflow.in/169>

(b) As it is the case of indirect recursion so let first make it as direct recursion then apply rules of removal of left recursion.

to make it as direct recursion first production remain unchanged while in second production substitute the right hand side of first production wherever it comes. In the question S comes in middle of A so substitute the right hand side of production S. Now after substituting it looks like..

$$A \rightarrow Ac|Aad|bd|\epsilon$$

Now remove direct recursion from it

For removal of direct recursion rule--

$$A \rightarrow A\alpha_1 | \dots | A\alpha_n | \beta_1 | \dots | \beta_m$$

Replace these with two sets of productions, one set for A :

$$A \rightarrow \beta_1 A' | \dots | \beta_m A'$$

and another set for the fresh nonterminal A'

$$A' \rightarrow \alpha_1 A' | \dots | \alpha_n A' | \epsilon$$

After applying these rule we'll get..

$$A \rightarrow bdA' | A'$$

$$A' \rightarrow cA' | adA' | \epsilon$$

Now complete production without left recursion is...

$$S \rightarrow Aa | b$$

$$A \rightarrow bdA' | A'$$

$$A' \rightarrow cA' | adA' | \epsilon$$

4 votes

-- shashi shekhar (505 points)

2.7.17 Grammar: GATE1999_2.15 top

<http://gateoverflow.in/1493>



Selected Answer

Let grammar is like this :

$$\begin{aligned} S &\rightarrow a \\ A &\rightarrow AbA \end{aligned}$$

This grammar is left as well as right recursive but still unambiguous.. **A** is useless production but still part of grammar.. A grammar having both left as well as right recursion may or may not be ambiguous ..

Let's take a grammar say

$$S \rightarrow SS$$

Now, according to the link https://en.wikipedia.org/wiki/Formal_grammar

For a grammar G, if we have L(G) then all the strings/sentences in this language can be produced after some finite number of steps .

But, for the grammar

$$S \rightarrow SS$$

Can we produce any string after a finite number of steps ? NO, and hence language of this grammar is empty set {} .

Hence, If a grammar is having both left and right recursion, then grammar may or may not be ambiguous .

15 votes

-- Digvijay (46k points)

2.7.18 Grammar: GATE2000-2.21, ISRO2015-24 [top](#)

<http://gateoverflow.in/668>



Selected Answer

I guess its B.

Operator which is at lower level in the grammar is termed to have higher precedence.

18 votes

-- Gate Keeda (18.8k points)

2.7.19 Grammar: GATE2001-1.18 [top](#)

<http://gateoverflow.in/711>



Selected Answer

(A) is the answer.

(A) We can not have different Left Most Derivation and Right Most Derivation parse trees BUT we can certainly have different LMD and RMD for a given string.(LMD and RMD here refers to the order of various productions used for derivation which could be different.)

(D) is wrong w.r.t. question because IT IS TRUE that any LR(k) IS NEVER AMBIGUOUS and so an ambiguous can never be an LR(K) for any k, no matter how large k becomes.

(B) and (C) can not be the answer because LL(1) is TOP-DOWN parser and LALR is powerful than SLR. So both are TRUE.

14 votes

-- Sandeep_Uniyal (7.3k points)

2.7.20 Grammar: GATE2001-18 [top](#)

<http://gateoverflow.in/759>



Selected Answer

a) $S \rightarrow aS'/bS'$

$S' \rightarrow aS'/bS'/e$

b) $S \rightarrow aSbS \rightarrow abS \rightarrow abaSbS \rightarrow ababS \rightarrow abab$

$S \rightarrow aSbS \rightarrow abSaSbS \rightarrow abaSbS \rightarrow ababS \rightarrow abab$

Two derivation so it is ambiguous

11 votes

-- Pooja Palod (31.2k points)

2.7.21 Grammar: GATE2003-56 [top](#)

<http://gateoverflow.in/944>

Selected Answer

First (S) = {i, a}

First(S') = {e, epsilon}

First(E) = {b}

Follow(S') = {e, \$}

Only when first contains epsilon we need to consider follow

M[S', e] = {S' -> eS(first), S' -> epsilon(considering follow)}

M[S', \$] = {S -> epsilon}

	a	i	b	e	t	\$
S	S -> a	S -> ietSS'				
S'				S' -> eS S' -> epsilon		S' -> epsilon
E			E -> b			

answer is d

 17 votes

-- Pooja Palod (31.2k points)

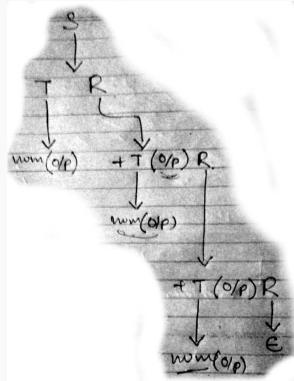
2.7.22 Grammar: GATE2003-58 [top](#)

<http://gateoverflow.in/946>

Selected Answer

answer = option B

9 5 + 2 +

 17 votes

-- Amar Vashishth (27.9k points)

2.7.23 Grammar: GATE2004-45 [top](#)

<http://gateoverflow.in/1042>



Selected Answer

we can do in simple manner ...

here # is multiplication and & is addition by semantics rules given in the question ...

and by observation of productions ..

1. here & (+) is higher precedence than #(*). bcoz & is far from starting symbol ...

2. and both &, # are left associative so we can solve the express in this way ...

$((2*(3+5))*(6+4)) = 160$ so ans should be (C).

17 votes

-- sonam vyas (12.8k points)

2.7.24 Grammar: GATE2004-8 [top](#)



Selected Answer

answer is B .because operator grammar does not contain 1) nullable variable 2) 2 adjacent non-terminal on rhs of production

13 votes

-- koushiksng264 (233 points)

2.7.25 Grammar: GATE2004-88 [top](#)



Selected Answer

above CFG generate string b, aaa..

b will eliminate options A and D

aaa eliminate options B.

C is answer i.e. number of a = 3k, k = 0,1,2....

12 votes

-- Digvijay (46k points)

2.7.26 Grammar: GATE2005-59 [top](#)



Selected Answer

$n+n^*n$

$E+n^*n$

E^*n

E

string in red indicates handle here

so ans is d

12 votes

-- Pooja Palod (31.2k points)

2.7.27 Grammar: GATE2006-32, ISRO2016-35 [top](#)



Selected Answer

I. True. G is ambiguous. E.g. the string ab has multiple derivation trees like $S \rightarrow SS \rightarrow abS \rightarrow ab$, and $S \rightarrow ab$.

II. False. G does not produce all strings with equal no. of a 's and b 's. ($aabb$ cannot be generated).

III. True. The given grammar G generates the language $(ab + ba)^*$, which is Regular and therefore also DCFL. So, a D-PDA can be designed for G .

Hence, the answer is option B.

136 votes

-- Pooja Palod (31.2k points)

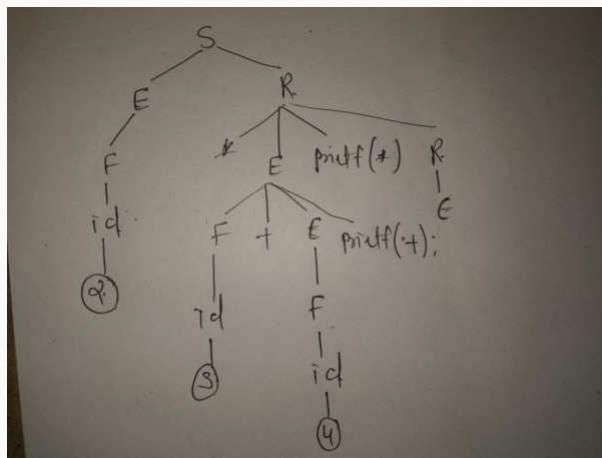
2.7.28 Grammar: GATE2006-59 [top](#)



Selected Answer

<http://gateoverflow.in/1837>

It will be D. make a tree and perform post order evaluation.



13 votes

-- Gate Keeda (18.8k points)

2.7.29 Grammar: GATE2006-84 [top](#)



Selected Answer

<http://gateoverflow.in/1856>

84, answer = option D 85 answer = option A

Q 84 .

option A

$C \Rightarrow a$

or, $C \Rightarrow b$

or, $C \Rightarrow aC \Rightarrow aaC \Rightarrow aab$.. soon

at last you have to put either $C \rightarrow a$ or $C \rightarrow b$

so production C is used to derive $a^{n+1}b^n$ or $a^n b^{n+1}$ $n >= 0$

$S \rightarrow AC$ [$Aa^n b^{n+1}$] can make $a^{n+1}b^{n+1}$ as single a can be derived from A [$A \Rightarrow aA \Rightarrow a$ as $A \rightarrow \epsilon$] , similarly $S \rightarrow CB$

simple way, ab can be derived from grammar as $S \Rightarrow AC \Rightarrow aAC \Rightarrow aC \Rightarrow ab$

option A is wrong

option B , language is used to drive a^+b^* or a^*b^+ , ab will be derived as $S \Rightarrow aS \Rightarrow ab$

option B is wrong

Option C

$C \Rightarrow \epsilon$

or $C \Rightarrow aCb \Rightarrow aaCbb \Rightarrow aaaCbbb \dots$ soon at last need to put $C \rightarrow \epsilon$

Production C will generate $a^n b^n$ $n \geq 0$

$S \rightarrow AC$ can generate $a^n b^n$ as A can be ϵ , similarly $S \rightarrow CB$

option C is wrong

Option D .

production C is used for generate $a^n b^n$ as in option C

$S \rightarrow AC$ will increase no of a's before $a^n b^n$

as A will generate a or aa or aaa... i.e a^+ , so $S \rightarrow AC$ will generate $a^+ a^n b^n$, i.e $a^i b^j$ $i > j$

$S \rightarrow CB$ will generate $a^n b^n b^+$ i.e $a^i b^j$ $i < j$

option D is right .

Q 85

$L = a^l b^m$ $l \neq m$ means either $l > m$ or $l < m$

Case I [$l > m$]

if $l > m$, $a^l b^m$ can be written as $a^{l-m} a^m b^m$ [$l-m$ cannot be 0 as l should be $> m$]

$S \rightarrow AC$, one step

a^{l-m} use $l-m$ steps using productions of A

[as $l-m = 1$, one step $A \rightarrow a$

$l-m = 2$, two steps $A \rightarrow aA \rightarrow aa$

$l-m = 3$, three steps , $A \rightarrow aA \rightarrow aaA \rightarrow aaa\dots$ so on]

$a^m b^m$ will be generate in $m + 1$ steps using production C

[as $m = 0$ one step $C \rightarrow \epsilon$

$m = 1$, two steps $C \rightarrow aCb \rightarrow ab$

$m = 2$, three steps $C \rightarrow aCb \rightarrow aaCbb \rightarrow aabb\dots$ so on]

so if $l > m$ total steps = $1+l-m+m+1 = l+2$

Case II [$l < m$]

similar if $l < m$, $a^l b^m$ can be written as $a^l b^l b^{m-l}$ [$m-l$ cannot be 0 as m should be $> l$]

$S \rightarrow CB$ one step

$a^l b^l$ will be derived using $l+1$ steps

b^{m-l} will be derived using $m-l$ steps

total = $1+l+1+m-l = m+2$

so $L = a^l b^m$ $l \neq m$ will take $\max(l,m)+2$ steps

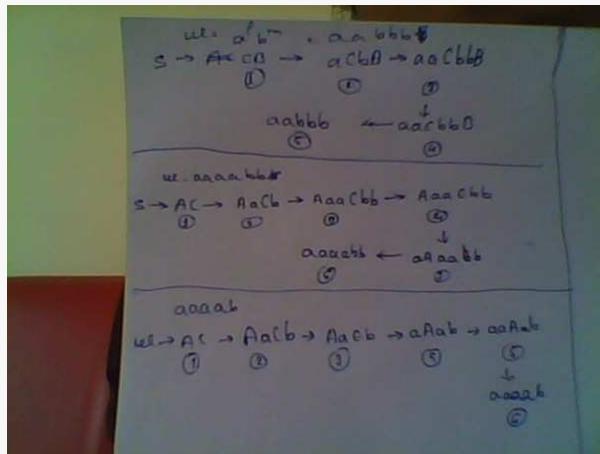
16 votes

-- Praveen Saini (53k points)

2.7.30 Grammar: GATE2006-85

<http://gateoverflow.in/79801>

I think option A seems correct, please see the image for explanation



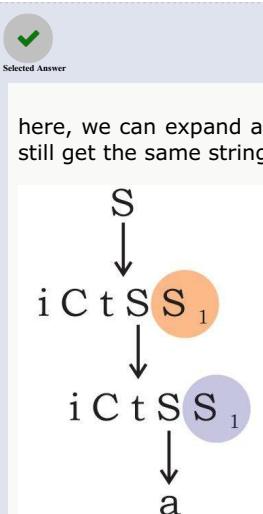
in case 2 and 3 reducing number of b's has no affect on number of productions required



-- rameshbabu (3.2k points)

2.7.31 Grammar: GATE2007-52 top

<http://gateoverflow.in/1250>



this means that the Grammar is Ambiguous. LL(1) cannot be ambiguous. [Here's a Proof for that.](#)

LL(1) Derivations

Left to Right Scan of input

Leftmost Derivation

(1) look ahead 1 token at each step

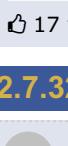
Alternative characterization of LL(1) Grammars:

Whenever $A \rightarrow \alpha \mid \beta \in G$

1. $FIRST(\alpha) \cap FIRST(\beta) = \{ \}$, and
 2. if $\alpha \xrightarrow{*} \epsilon$ then $FIRST(\beta) \cap FOLLOW(A) = \{ \}$.

Corollary: No Ambiguous Grammar is LL(1).

answer = **option C**



-- Amar Vashishth (27.9k points)

2.7.32 Grammar: GATE2007-53 [top](#)

www.scholarone.com

P:- This is false.

Every regular language is LL(1) meaning we have a LL(1) grammar for it. But we can not say same about every Regular Grammar. For example, every regular language can be represented by Left & Right Linear Grammar, where Left Linear Grammar is not LL(1), Right linear is.

Example aa^* we can represent this as $S \rightarrow Sa|a$ which is not LL(1) ,but $S \rightarrow a|aS$ is LL(1).

Q:- This is true because of every LL(1) is LR(1).

All regular sets have Right recursive grammar, which is LL(1) & Every LL(1) is LR(1).

We can also say that LR(1) accepts DCFL & Regular languages are subset of DCFL.

So Answer is C.

1 27 votes

-- Aakash Kansse (361 points)

2.7.33 Grammar: GATE2007-78 [top](#)

<http://gateoverflow.in/1272>



Selected Answer

$S \rightarrow aB$
 $\rightarrow aaBB$
 $\rightarrow aabB$
 $\rightarrow aabbS$
 $\rightarrow aabbAB$
 $\rightarrow aabbab$

1 12 votes

-- Arjun Suresh (287k points)

2.7.34 Grammar: GATE2007-79 [top](#)

<http://gateoverflow.in/43512>



Selected Answer

$S \rightarrow aB$
 $\rightarrow aaBB$
 $\rightarrow aabB$
 $\rightarrow aabbS$
 $\rightarrow aabbAB$
 $\rightarrow aabbab$

$S \rightarrow aB$
 $\rightarrow aaBB$ (till now, only 1 choice possible)
 $\rightarrow aabSB$ (last time we took $B \rightarrow b$, now taking $B \rightarrow bS$)
 $\rightarrow aabbAB$
 $\rightarrow aabbab$
 $\rightarrow aabbab$

So, totally 2 possible derivation trees.

1 8 votes

-- Arjun Suresh (287k points)

2.7.35 Grammar: GATE2007-IT-9 [top](#)

<http://gateoverflow.in/3442>



Selected Answer

c) grammar may change but language remain the same.

1 24 votes

-- Arpit Dhuriya (3k points)

c)L (D) = L (G) , **Both must represent same language** .Also if we are converting a grammar from Ambiguous to UnAmbiguous form , we must ensure that our new grammar represents the same language as previous grammar.

For ex G1: $S \rightarrow Sa|aS|a$; {Ambiguous (2 parse trees for string 'aa')}

G1':S->aS/a; {Unambiguous}

Both represents language represented by regular expression: a^+

11 votes

-- Anurag Semwal (7.6k points)

2.7.36 Grammar: GATE2008-50 [top](#)

<http://gateoverflow.in/395>



Selected Answer

Answer is C:

Statement 1 is **true**: Using GNF we can convert Left recursive grammar to right recursive and by using reversal of CFG and GNF we can convert right recursive to left recursive.

Statement 2 is **false**: because if ϵ is in the language then we can't remove ϵ production from Start symbol. (For example $L = a^*$)

Statement 3 is **true** because right linear grammar generates regular set

Statement 4 is **true**, only two non-terminals are there in each production in CNF. So it always form a binary tree.

22 votes

-- Vikrant Singh (13.4k points)

2.7.37 Grammar: GATE2010-38 [top](#)

<http://gateoverflow.in/2339>



Selected Answer

It will be C.

For LL(1) take First(S). and do intersection between the result. if intersection is Phi then LL(1) else not.

Making a parsing table and checking if there are two or more entries under any terminal. If yes then neither LL(1) nor LR(1).

18 votes

-- Gate Keeda (18.8k points)

2.7.38 Grammar: GATE2014-2-17 [top](#)

<http://gateoverflow.in/1973>



Selected Answer

P->Q+P here P is to right of +

so + is right associative

Similarly for T->T *U * is left associative as T is to left of *

so ans is b

12 votes

-- Pooja Palod (31.2k points)

2.7.39 Grammar: GateBook Mock Test_2(Compilers) [top](#)

<http://gateoverflow.in/116952>



Selected Answer

In G1, T can take the form of $v^* v^* v^* v^* v^* \dots$

In G2, R can take the form $* v^* v^* v^* v^* \dots$,

so T can take the form of $v^* v^* v^* v^* v^* \dots$

Thus, in both G1 and G2, E can take the form $v * v * v + v * v * v * v + v * v \dots$

These are the algebraic expressions involving only addition and multiplication.

In G3, T can take the form $v * v * v * v$, so R can take the form $+ v * v + v * v$. Thus, through the production $E \rightarrow T R$, E can take the form of any arithmetic expression.

However, E can also take the form of non-arithmetic expressions,

as with $E \rightarrow R \rightarrow + T R \rightarrow + v R \rightarrow + v + T R \rightarrow + v + v R \rightarrow + v + v$.

In addition, G3 can generate ϵ , which G1 and G2 cannot.

Thus, while G1 and G2 generate the same language,

G3 generates a somewhat larger language. Incidentally, G1 demonstrates left-recursion because it contains productions of the form $X \rightarrow X \alpha$ (in addition to non-problematic productions of the form $X \rightarrow \beta$).

Left recursion can be eliminated by introducing a helper non-terminal R and changing the grammar to read

$X \rightarrow \beta R$
 $R \rightarrow \alpha R \mid \epsilon$

hence C is the answer .

1 3 votes

-- Bikram (28.3k points)

2.8

Infix Postfix(2) top

2.8.1 Infix Postfix: GATE1997_1.7 top

<http://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 compiler-design normal infix-postfix

Answer

2.8.2 Infix Postfix: GATE1998_19b top

<http://gateoverflow.in/15708>

Compute the post fix equivalent of the following expression $3^* \log(x+1) - \frac{a}{2}$

gate1998 compiler-design infix-postfix

Answer

Answers: Infix Postfix

2.8.1 Infix Postfix: GATE1997_1.7 top

<http://gateoverflow.in/2223>



Selected Answer

A. An operator stack // Infix to (Postfix or Prefix)

B. An operand stack //Postfix or Prefix Evaluation

C. An operand stack and an operator stack //we never use two stacks

But for Prefix to (Infix or postfix) OR Postfix to (Infix or prefix) We can use a stack where both operator and operand can present simultaneously

D. A parse tree // Not relevant to this question

Hence, Option A is Ans

<http://condor.depaul.edu/ichu/csc415/notes/notes9/Infix.htm> is a good read.

1 3 votes

-- Rajesh Pradhan (16.9k points)

A.

we use operator stack (only operators are pushed as +, *, (,), /) for converting infix to postfix. And we use operand stack(operands such as 5,4,17 etc) for postfix evaluation.

1 15 votes

-- Gate Keeda (18.8k points)

2.8.2 Infix Postfix: GATE1998_19b [top](#)

<http://gateoverflow.in/15708>



Selected Answer

According to <http://faculty.washington.edu/jstraub/dsa/aexp/>

The priority of the operators follows the usual conventions:

- The highest priority is assigned to unary operators (note that, in this context, a function such `assIn` is considered a unary operator). All unary operators have the same priority.
- Exponentiation has the second highest priority.
- The third highest priority is assigned to the multiplication and division operators.
- The lowest priority is given to the addition and subtraction operators.

Example:->

Infix expression: $3 * \log(10)$

Postfix expression:

= $3 * (10 \log)$ // (Priority of unary operator `log` forces $\log(10)$ to evaluate first.)

= $3 10 \log *$

Now For our case

$3 * \log(x + 1) - a / 2$

first content inside parenthesis will evaluate

so $x+1$ will become $x1+$

now **among (*,/,log,+,-)operators ..log has highest priority due to unary opeartor**

So, $\log(x1+)$ will become **$x1+log$**

now **suppose z= x1+log**

so convert this $3 * z - a / 2$ to postfix

$3z * a2 / -$

now substitue $z = x1+log$

$3x1+log * a2 / -$ is Ans.

1 11 votes

-- Rajesh Pradhan (16.9k points)

$3x1 + log * a2 / -$

1 12 votes

-- Happy Mittal (10.7k points)

2.9

Intermediate Code(7) [top](#)

2.9.1 Intermediate Code: GATE1992-11a [top](#)

<http://gateoverflow.in/590>

Write syntax directed definitions (semantic rules) for the following grammar to add the type of each identifier to its entry in the symbol table during semantic analysis. Rewriting the grammar is not permitted and semantic rules are to be added to the ends of productions only.

$$D \rightarrow TL;$$

$$T \rightarrow \text{int}$$

$$T \rightarrow \text{real}$$

$$L \rightarrow L, id$$

$$L \rightarrow id$$
[gate1992](#) [compiler-design](#) [syntax-directed-translation](#) [intermediate-code](#) [normal](#)
[Answer](#)

2.9.2 Intermediate Code: GATE1992-11b [top](#)

<http://gateoverflow.in/4358>

Write 3 address intermediate code (quadruples) for the following boolean expression in the sequence as it would be generated by a compiler. Partial evaluation of boolean expressions is not permitted. Assume the usual rules of precedence of the operators.

$$(a + b) > (c + d) \text{ or } a > c \text{ and } b < d$$

[gate1992](#) [compiler-design](#) [syntax-directed-translation](#) [intermediate-code](#) [descriptive](#)
[Answer](#)

2.9.3 Intermediate Code: GATE1994_1.12 [top](#)

<http://gateoverflow.in/2453>

Generation of intermediate code based on an abstract machine model is useful in compilers because

- A. it makes implementation of lexical analysis and syntax analysis easier
- B. syntax-directed translations can be written for intermediate code generation
- C. it enhances the portability of the front end of the compiler
- D. it is not possible to generate code for real machines directly from high level language programs

[gate1994](#) [compiler-design](#) [intermediate-code](#) [easy](#)
[Answer](#)

2.9.4 Intermediate Code: GATE2014-2-34 [top](#)

<http://gateoverflow.in/1993>

For a C program accessing **X[i] [j] [k]**, the following intermediate code is generated by a compiler. Assume that the size of an **integer** is 32 bits and the size of a **character** is 8 bits.

```
t0 = i * 1024
t1 = j * 32
t2 = k * 4
t3 = t1 + t0
t4 = t3 + t2
t5 = X[t4]
```

Which one of the following statements about the source code for the C program is CORRECT?

- A. **X** is declared as "int **X[32] [32] [8]**".
- B. **X** is declared as "int **X[4] [1024] [32]**".
- C. **X** is declared as "char **X[4] [32] [8]**".
- D. **X** is declared as "char **X[32] [16] [2]**".

[gate2014-2](#) [compiler-design](#) [intermediate-code](#) [programming-in-c](#) [normal](#)
[Answer](#)

2.9.5 Intermediate Code: GATE2014-3-17 [top](#)

<http://gateoverflow.in/2051>

One of the purposes of using intermediate code in compilers is to

- A. make parsing and semantic analysis simpler.
- B. improve error recovery and error reporting.
- C. increase the chances of reusing the machine-independent code optimizer in other compilers.
- D. improve the register allocation.

[gate2014-3](#) [compiler-design](#) [intermediate-code](#) [easy](#)

[Answer](#)

2.9.6 Intermediate Code: GATE2015-1_55 [top](#)

<http://gateoverflow.in/8365>

The least number of temporary variables required to create a three-address code in static single assignment form for the expression $q + r / 3 + s - t * 5 + u * v/w$ is _____.

[gate2015-1](#) [compiler-design](#) [intermediate-code](#) [normal](#) [numerical-answers](#)

[Answer](#)

2.9.7 Intermediate Code: GATE2015-1_8 [top](#)

<http://gateoverflow.in/8096>

For computer based on three-address instruction formats, each address field can be used to specify which of the following:

- (S1) A memory operand
 - (S2) A processor register
 - (S3) An implied accumulator register
- A. Either S1 or S2
 - B. Either S2 or S3
 - C. Only S2 and S3
 - D. All of S1, S2 and S3

[gate2015-1](#) [compiler-design](#) [intermediate-code](#) [normal](#)

[Answer](#)

Answers: Intermediate Code

2.9.1 Intermediate Code: GATE1992-11a [top](#)

<http://gateoverflow.in/590>

PRODUCTION RULE

D->TL

T->int

T->real

L->L,id

L->id

SEMANTIC ACTIONS

L.in:=T.type

T.type:=integer

T.type:=real

L1.in=L.in

Enter_type(id.entry, L.in)

Enter_type(id.entry, L.in)

2 votes

-- Anupoju Satish Kumar (291 points)

2.9.2 Intermediate Code: GATE1992-11b [top](#)

<http://gateoverflow.in/43583>



Selected Answer

Each instruction in quadruples presentation is divided into four fields: operator, arg1, arg2, and result. The above example is represented below in quadruples format:

$(a+b) > (c+d)$ OR $a > c$ AND $b < d$

$(t1 > t2)$ OR $a > c$ AND $b < d$

$t3 \text{ OR } t4 \text{ AND } t5$

$t3 \text{ OR } t6$

$t1 = a + b$

$t2 = c + d$

$t3 = t1 > t2$

$t4 = a > c$

$t5 = b < d$

$t6 = t4 \text{ AND } t5$

$t7 = t3 \text{ OR } t6$

Op	arg1	arg2	Result
+	a	b	t1
+	c	d	t2
>	t1	t2	t3
.>	a	c	t4
<	b	d	t5
AND	t4	t5	t6
OR	t3	t6	t7

15 votes

-- shekhar chauhan (42.2k points)

2.9.3 Intermediate Code: GATE1994_1.12 [top](#)

<http://gateoverflow.in/245>



Selected Answer

C. stating the actual use of the Intermediate Code.

Also optimizations can be done on intermediate code enhancing the portability of the optimizer.

16 votes

-- Gate Keeda (18.8k points)

2.9.4 Intermediate Code: GATE2014-2-34 [top](#)

<http://gateoverflow.in/1993>



Selected Answer

k is multiplied by 4, means sizeof(datatype) is int.

j is multiplied by 32, means the inner most dimension of array is $32/4 = 8$ (we have to divide by the size of the inner dimension- which here is a simple integer)

i is multiplied by 1024, means the second dimension of array is $1024/32 = 32$ ($32 = 8*4$ is the size of the inner dimension here)

So, (A) is correct. The first dimension is not needed for code generation and that is why in C language while passing an array to a function, we can omit the value of the first dimension but not any others.

We can also do as follows:

$X[i][j][k] = *(*(*X + i) + j) + k$

In Integer arithmetic, this equals

$*(*(*X + i * sizeof(*X)) + j * sizeof(**X)) + k * sizeof(***)$

as for every add to a pointer we have to multiply the size of the pointed value (to get a valid address)

So, from the given code we get

`sizeof(**X) = 4, - int`

`sizeof(**X) = 32 - int array of size 8`

`sizeof(*X) = 1024 - 2 D int array of size [32] havinf size of inner 1D array 32.`

So, the inner dimensions must be 32 and 8 and type must be integer. So, only option A matches.

1 23 votes

-- Arjun Suresh (287k points)

X is of type integer and dimensions of X are $r_1 \times r_2 \times r_3$.

value of $r_2 = 32$

$r_3 = 8$

1 14 votes

-- Vikrant Singh (13.4k points)

2.9.5 Intermediate Code: GATE2014-3-17 [top](#)

<http://gateoverflow.in/2051>



C.

that is the actual use of intermediate code generator in a compiler.

1 16 votes

-- Gate Keeda (18.8k points)

2.9.6 Intermediate Code: GATE2015-1_55 [top](#)

<http://gateoverflow.in/8365>



Answer is 8.

In **compiler design**, **static single assignment form** (often abbreviated as **SSA form** or simply **SSA**) is a property of an **intermediate representation (IR)**, which requires that each variable is assigned exactly once, and every variable is defined before it is used. Existing variables in the original IR are split into **versions**, new variables.

We will need a temporary variable for storing the result of each binary operation as **SSA (Static Single Assignment)** implies the variable cannot be repeated on **LHS of assignment**.

`q + r / 3 + s - t * 5 + u * v/w`

```
t1 = r/3;
t2 = t*5;
t3 = u*v;
t4 = t3/w;
t5 = q + t1;
t6 = t5 + s;
t7 = t6 - t2;
t8 = t7 + t4
```

<http://web.stanford.edu/class/archive/cs/cs143/cs143.1128/handouts/240%20TAC%20Examples.pdf>

36 votes

-- Arjun Suresh (287k points)

2.9.7 Intermediate Code: GATE2015-1_8 [top](#)

<http://gateoverflow.in/8096>



Selected Answer

Three address Instruction

Computer with three addresses instruction format can use each address field to specify either processor register or memory operand.

Ex. $X = (A + B) * (C + A)$

Equivalent Three address Instructions

```
ADD R1, A, B    R1 @ M [A] + M [B]
ADD R2, C, D    R2 @ M [C] + M [D]
MUL X, R1, R2   M [X] @ R1 * R2
```

The advantage of the three address formats is that it results in short program when evaluating arithmetic expression. The disadvantage is that the binary-coded instructions require too many bits to specify three addresses.

http://www.laureateiit.com/projects/bacii2014/projects/coa_anil/instruction_formate.html#sthash.7y6Hcwvd.dpuf

17 votes

-- Prasanna Ranganathan (3.9k points)

option A either A memory operand or A processor register.

15 votes

-- Anoop Sonkar (4.8k points)

2.10

Left Recursion(1) [top](#)

2.10.1 Left Recursion: GATE2017-2-32 [top](#)

<http://gateoverflow.in/118374>

Consider the following expression grammar G :

- $E \rightarrow E - T \mid T$
- $T \rightarrow T + F \mid F$
- $F \rightarrow (E) \mid id$

Which of the following grammars is not left recursive, but is equivalent to G ?

- $E \rightarrow E - T \mid T$
 $T \rightarrow T + F \mid F$
 $F \rightarrow (E) \mid id$
- $E \rightarrow TE'$

$E' \rightarrow -TE' \mid \epsilon$
 $T \rightarrow T + F \mid F$
 $F \rightarrow (E) \mid id$
 C. $E \rightarrow TX \mid (TX)$
 $X \rightarrow -TX \mid +TX \mid \epsilon$
 $T \rightarrow id$
 $F \rightarrow (E) \mid id$
 D. $E \rightarrow TX$
 $X \rightarrow -TX \mid \epsilon$
 $T \rightarrow +FY \mid \epsilon$
 $F \rightarrow (E) \mid id$

gate2017-2 | grammar | left-recursion

Answer

Answers: Left Recursion

2.10.1 Left Recursion: GATE2017-2-32 [top](#)

<http://gateoverflow.in/118374>



Selected Answer

Since the grammar given in the question is left recursive , we need to remove left recursion ,

If Grammar is of form

$A \rightarrow A\alpha \mid \beta$

then after removal of left recursion it should be written as

$A \rightarrow \beta A'$

$A' \rightarrow \alpha A' \mid \epsilon$

Since the grammar is :

$E \rightarrow E - T \mid T$ (Here α is ' $-T$ ' and β is T)

$T \rightarrow T + F \mid F$ (Here α is ' $+F$ ' and β is F)

$F \rightarrow (E) \mid id$ (It is not having left recursion)

Rewriting after removing left recursion :

$E \rightarrow TE'$

$E' \rightarrow -TE' \mid \epsilon$

$T \rightarrow FT'$

$T' \rightarrow +FT' \mid \epsilon$

$F \rightarrow (E) \mid id$

Now replace E' with X and T' with Y to match with Option C.

$E \rightarrow TX$

```
X---> -TX | ε
T ---> FY
Y---> +FY | ε
F ---> (E) | id
Hence C is correct.
```

8 votes

-- Indranil Maji (203 points)

2.11**Lexical Analysis(4)** top**2.11.1 Lexical Analysis: GATE2000-1.18, ISRO2015-25** top<http://gateoverflow.in/641>

The number of tokens in the following C statement is

```
printf("i=%d, &i=%x", i, &i);
```

- A. 3
- B. 26
- C. 10
- D. 21

gate2000 compiler-design lexical-analysis easy isro2015
Answer**2.11.2 Lexical Analysis: GATE2010-13** top<http://gateoverflow.in/2186>

Which data structure in a compiler is used for managing information about variables and their attributes?

- A. Abstract syntax tree
- B. Symbol table
- C. Semantic stack
- D. Parse table

gate2010 compiler-design lexical-analysis easy
Answer**2.11.3 Lexical Analysis: GATE2011_1** top<http://gateoverflow.in/2103>

In a compiler, keywords of a language are recognized during

- (A) parsing of the program
- (B) the code generation
- (C) the lexical analysis of the program
- (D) dataflow analysis

gate2011 compiler-design lexical-analysis easy
Answer**2.11.4 Lexical Analysis: GATE2011_19** top<http://gateoverflow.in/2121>

The lexical analysis for a modern computer language such as Java needs the power of which one of the following machine models in a necessary and sufficient sense?

- (A) Finite state automata
- (B) Deterministic pushdown automata
- (C) Non-deterministic pushdown automata
- (D) Turing machine

[gate2011](#) [compiler-design](#) [lexical-analysis](#) [easy](#)
Answer

Answers: Lexical Analysis

2.11.1 Lexical Analysis: GATE2000-1.18, ISRO2015-25 [top](#)

<http://gateoverflow.in/641>

Selected Answer

answer - C

Tokens are:

1. printf
2. (
3. "i=%d, &i=%x"
4. ,
5. i
6. ,
7. &
8. i
9.)
10. ;

19 votes

-- ankitrokdeonsns (9.1k points)

2.11.2 Lexical Analysis: GATE2010-13 [top](#)

<http://gateoverflow.in/218>

Selected Answer

Symbol table is answer . It can be implemented by using array , hash table , tree and eve some time with the help of !Inked list !

15 votes

-- Dexter (9.6k points)

B. It uses array to implement.

13 votes

-- Gate Keeda (18.8k points)

2.11.3 Lexical Analysis: GATE2011_1 [top](#)

<http://gateoverflow.in/213>

Selected Answer

Typically, the lexical analysis phase of compilation breaks the input text up into sequences of lexemes that each belongs to some particular token type that's useful in later analysis. Consequently, keywords are usually first recognized during lexical analysis in order to make parsing easier. Since parsers tend to be implemented by writing context-free grammars of tokens rather than of lexemes (that is, the *category* of the lexeme rather than the *contents* of the lexeme), it is significantly easier to build a parser when keywords are marked during lexing. Any identifier is also a token so it is recognized in lexical Analysis .

Hence option C is True.

ref@ <http://stackoverflow.com/questions/5202709/phases-of-a-compiler>

18 votes

-- Mithlesh Upadhyay (5.3k points)

2.11.4 Lexical Analysis: GATE2011_19 [top](#)

<http://gateoverflow.in/212>

Selected Answer

Answer - A

In compiler lexical analyzer categorizes character sequence into lexemes and produces tokens as output for parser. And

tokens are expressed in regular expressions so a simple Finite Automata is sufficient for it.

17 votes

-- ankitrokdeonsns (9.1k points)

2.12

Linking(3) top

2.12.1 Linking: GATE1991_03,ix top

<http://gateoverflow.in/519>

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

A "link editor" is a program that:

- A. matches the parameters of the macro-definition with locations of the parameters of the macro call
- B. matches external names of one program with their location in other programs
- C. matches the parameters of subroutine definition with the location of parameters of subroutine call.
- D. acts as a link between text editor and the user
- E. acts as a link between compiler and the user program

gate1991 compiler-design normal linking

Answer

2.12.2 Linking: GATE2003-76 top

<http://gateoverflow.in/962>

Which of the following is NOT an advantage of using shared, dynamically linked libraries as opposed to using statically linked libraries?

- A. Smaller sizes of executable files
- B. Lesser overall page fault rate in the system
- C. Faster program startup
- D. Existing programs need not be re-linked to take advantage of newer versions of libraries

gate2003 compiler-design runtime-environments linking easy

Answer

2.12.3 Linking: GATE2004-9 top

<http://gateoverflow.in/1006>

Consider a program P that consists of two source modules M_1 and M_2 contained in two different files. If M_1 contains a reference to a function defined in M_2 the reference will be resolved at

- A. Edit time
- B. Compile time
- C. Link time
- D. Load time

gate2004 compiler-design easy linking

Answer

Answers: Linking

2.12.1 Linking: GATE1991_03,ix top

<http://gateoverflow.in/519>



Selected Answer

Link editor or (linker) performs

1. external symbol resolution
2. relocation.

ANS: B

Matches external names of one program with their location in other programs

5 votes

-- pramod (3.2k points)

2.12.2 Linking: GATE2003-76 [top](#)

<http://gateoverflow.in/962>



Selected Answer

option c: DLL takes more time in program setup (in loading and linking phase to set up the global offset table and load and link the required libraries)

- Since DLLs are separated from executable, the size of executable becomes smaller. Since DLLs are shared among multiple executables, the total memory usage of the system goes down and hence overall page fault rate decreases.
- Dynamic linking takes place during program runtime. So, if a DLL is replaced to a new version, it will automatically get linked during runtime. There is no explicit relinking required as in the case of static linking. (This works by linking the DLL calls to Global Offset Table and the contents of this table is filled during program run. A simple jump in static linking becomes an indirect jump in dynamic linking).

Reffer :From galvin

8.1.5 Dynamic Linking and Shared Libraries

Figure 8.3 also shows **dynamically linked libraries**. Some operating systems support only **static linking**, in which system language libraries are treated like any other object module and are combined by the loader into the binary program image. The concept of dynamic linking is similar to that of dynamic loading. Here, though, linking, rather than loading, is postponed until execution time. This feature is usually used with system libraries, such as language subroutine libraries. Without this facility, each program on a system must include a copy of its language library (or at least the routines referenced by the program) in the executable image. This requirement wastes both disk space and main memory.

With dynamic linking, a *stub* is included in the image for each library-routine reference. The stub is a small piece of code that indicates how to locate the appropriate memory-resident library routine or how to load the library if the routine is not already present. When the stub is executed, it checks to see whether the needed routine is already in memory. If not, the program loads the routine into memory. Either way, the stub replaces itself with the address of the routine and executes the routine. Thus, the next time that particular code segment is reached, the library routine is executed directly, incurring no cost for dynamic linking. Under this scheme, all processes that use a language library execute only one copy of the library code.

This feature can be extended to library updates (such as bug fixes). A library may be replaced by a new version, and all programs that reference the library will automatically use the new version. Without dynamic linking, all such programs would need to be relinked to gain access to the new library. So that programs will not accidentally execute new, incompatible versions of libraries, version information is included in both the program and the library. More than one version of a library may be loaded into memory, and each program uses its version information to decide which copy of the library to use. Minor changes retain the same version number, whereas major changes increment the version number. Thus, only programs that are compiled with the new library version are affected by the incompatible changes incorporated in it. Other programs linked before the new library was installed will continue using the older library. This system is also known as **shared libraries**.

22 votes

-- GateMaster Prime (1.6k points)

2.12.3 Linking: GATE2004-9 [top](#)

<http://gateoverflow.in/1006>



Selected Answer

answer - C. Each module is compiled separately and then linked together to make the executable. The below commands shows how to do this for two modules c1.c and c2.c using gcc.

```
gcc -c c1.c -o c1.o
gcc -c c2.c -o c2.o
gcc c1.o c2.o -o C.exe
```

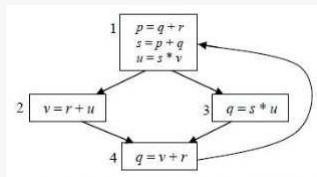
10 votes

-- ankitrokdeonsns (9.1k points)

2.13**Live Variable(1)** [top](#)**2.13.1 Live Variable: GATE2015-1_50** [top](#)<http://gateoverflow.in/8356>

A variable x is said to be live at a statement S_i in a program if the following three conditions hold simultaneously:

- There exists a statement S_j that uses x
- There is a path from S_i to S_j in the flow graph corresponding to the program
- The path has no intervening assignment to x including at S_i and S_j



The variables which are live both at the statement in basic block 2 and at the statement in basic block 3 of the above control flow graph are

- A. p, s, u
- B. r, s, u
- C. r, u
- D. q, v

[gate2015-1](#) [compiler-design](#) [live-variable](#) [normal](#)

[Answer](#)

Answers: Live Variable**2.13.1 Live Variable: GATE2015-1_50** [top](#)<http://gateoverflow.in/8356>

Selected Answer

r, u.

p, and s are assigned to in 1 and there is no intermediate use of them before that. Hence p, and s are not live in both 2 and 3.

q is assigned to in 4 and hence is not live in both 2 and 3.

v is live at 3 but not at 2.

u is live at 3 and also at 2 if we consider a path of length 0 from 2 - 2.

So, r, u is the answer.

6 votes

-- Arjun Suresh (287k points)

2.14**Macros(3)** [top](#)**2.14.1 Macros: GATE1995_1.11** [top](#)<http://gateoverflow.in/2598>

What are x and y in the following macro definition?

```
macro Add x, y
    Load y
    Mul x
    Store y
end macro
```

- A. Variables

- B. Identifiers
 C. Actual parameters
 D. Formal parameters

gate1995 compiler-design macros easy

Answer

2.14.2 Macros: GATE1996_2.16 [top](#)

<http://gateoverflow.in/2745>

Which of the following macros can put a macro assembler into an infinite loop?

(i)

i.

```
.MACRO M1, X
.IF EQ, X ;if X=0 then
M1 X + 1
.ENDC
.IF NE, X ;if X ≠ 0 then
.WORD X ;address (X) is stored here
.ENDC
.ENDM
```

(ii)

ii.

```
.MACRO M2, X
.IF EQ, X
M2 X
.ENDC
.IF NE, X
.WORD X + 1
.ENDC
.ENDM
```

- A. (ii) only
 B. (i) only
 C. both (i) and (ii)
 D. None of the above

gate1996 compiler-design macros normal

Answer

2.14.3 Macros: GATE1997_1.9 [top](#)

<http://gateoverflow.in/2225>

The conditional expansion facility of macro processor is provided to

- test a condition during the execution of the expanded program
- to expand certain model statements depending upon the value of a condition during the execution of the expanded program
- to implement recursion
- to expand certain model statements depending upon the value of a condition during the process of macro expansion

gate1997 compiler-design macros easy

Answer

Answers: Macros

2.14.1 Macros: GATE1995_1.11 [top](#)

<http://gateoverflow.in/2598>



Selected Answer

ans is D

5 votes

-- jayendra (7.7k points)

2.14.2 Macros: GATE1996_2.16 [top](#)<http://gateoverflow.in/2745>

Selected Answer

if M2 macro is called with x=0, then it'll go into an infinite loop.
Hence correct option would be A.

1 votes

-- suraj (5k points)

2.14.3 Macros: GATE1997_1.9 [top](#)<http://gateoverflow.in/2225>

Selected Answer

Macro is expanded during the process of macro expansion. Hence, correct answer would be (d).

6 votes

-- suraj (5k points)

2.15**Parameter Passing(8)** [top](#)**2.15.1 Parameter Passing: CMI2010-A-08** [top](#)<http://gateoverflow.in/46140>

In programming language terminology, call by value refers to the fact that:

- A. A function call can return a value.
- B. When a function is called, arguments are copied into local storage.
- C. Functions can indirectly modify the value of external variables.
- D. Every argument passed to a function must have a value.

[cmi2010](#) [compiler-design](#) [runtime-environments](#) [parameter-passing](#)**Answer****2.15.2 Parameter Passing: GATE 2016-1-36** [top](#)<http://gateoverflow.in/39701>

What will be the output of the following pseudo-code when parameters are passed by reference and dynamic scoping is assumed?

```
a = 3;
void n(x) { x = x * a; print (x); }
void m(y) { a = 1 ; a = y - a; n(a); print (a); }
void main () { m(a); }
```

- A. 6,2
- B. 6,6
- C. 4,2
- D. 4,4

[gate2016-1](#) [parameter-passing](#) [normal](#)**Answer****2.15.3 Parameter Passing: GATE1988-2xv** [top](#)<http://gateoverflow.in/94333>

What is printed by following program, assuming call-by reference method of passing parameters for all variables in the parameter list of procedure P?

```
program Main(inout, output);
var
  a, b:integer;
  procedure P(x, y, z:integer);
begin
  y:=y+1
  z:=x+x
end P;
begin
  a:=2; b:=3;
  p(a+b, a, a);
```

```
    Write(a)
end.
```

gate1988 descriptive compiler-design runtime-environments parameter-passing numerical-answers

Answer

2.15.4 Parameter Passing: GATE1991-09a [top](#)

<http://gateoverflow.in/536>

Consider the following pseudo-code (all data items are of type integer):

```
procedure P(a, b, c);
  a := 2;
  c := a + b;
end {P}

begin
  x := 1;
  y := 5;
  z := 100;
  P(x, x+y, z);
  Write ('x = ', x, 'z = ', z);
end
```

Determine its output, if the parameters are passed to the Procedure P by

- i. value
- ii. reference
- iii. name

gate1991 compiler-design parameter-passing normal runtime-environments

Answer

2.15.5 Parameter Passing: GATE1991_03,x [top](#)

<http://gateoverflow.in/524>

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

Indicate all the true statements from the following:

- (a). Recursive descent parsing cannot be used for grammar with left recursion.
- (b). The intermediate form for representing expressions which is best suited for code optimization is the postfix form.
- (c). A programming language not supporting either recursion or pointer type does not need the support of dynamic memory allocation.
- (d). Although C does not support call-by-name parameter passing, the effect can be correctly simulated in C
- (e). No feature of Pascal typing violates strong typing in Pascal.

gate1991 compiler-design parameter-passing programming difficult

Answer

2.15.6 Parameter Passing: GATE2004-2,ISRO2017-54 [top](#)

<http://gateoverflow.in/899>

Consider the following function

```
void swap(int a, int b)
{
  int temp;
  temp = a;
  a = b;
  b = temp;
}
```

In order to exchange the values of two variables x and y.

- A. call swap(x, y)
- B. call swap(&x, &y)
- C. swap (x, y) cannot be used as it does not return any value

- D. swap (x, y) cannot be used as the parameters are passed by value

[gate2004](#) [compiler-design](#) [programming-in-c](#) [parameter-passing](#) [easy](#) [isro2017](#) [runtime-environments](#)

[Answer](#)

2.15.7 Parameter Passing: GATE2007-IT-33 [top](#)

<http://gateoverflow.in/3468>

Consider the program below in a hypothetical language which allows global variable and a choice of call by reference or call by value methods of parameter passing.

```
int i ;
program main ()
{
    int j = 60;
    i = 50;
    call f (i, j);
    print i, j;
}
procedure f (x, y)
{
    i = 100;
    x = 10;
    y = y + i ;
}
```

Which one of the following options represents the correct output of the program for the two parameter passing mechanisms?

- A. Call by value : i = 70, j = 10; Call by reference : i = 60, j = 70
- B. Call by value : i = 50, j = 60; Call by reference : i = 50, j = 70
- C. Call by value : i = 10, j = 70; Call by reference : i = 100, j = 60
- D. Call by value : i = 100, j = 60; Call by reference : i = 10, j = 70

[gate2007-it](#) [programming](#) [parameter-passing](#) [normal](#) [compiler-design](#) [runtime-environments](#)

[Answer](#)

2.15.8 Parameter Passing: GATE2013-42 [top](#)

<http://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](#)

[Answer](#)

Answers: Parameter Passing

2.15.1 Parameter Passing: CMI2010-A-08 [top](#)

<http://gateoverflow.in/46140>



Selected Answer

B) When a function is called, arguments are copied into local storage.

In call by value , a copy of variable send from where it is called and those are local to the calling function

1 votes

-- srestha (55.2k points)

2.15.2 Parameter Passing: GATE 2016-1-36 [top](#)

<http://gateoverflow.in/39701>



Selected Answer

It is a bit confusing as variable declaration is not explicit. But we can see that "a=3" and "a=1" are declaring new variables, one in global and other in local space.

Main is calling m(a). Since there is no local 'a', 'a' here is the global one.

In m, we have "a = 1" which declares a local "a" and gives 1 to it. "a = y-a" assigns $3-1 = 2$ to 'a'.

Now, in n(x), 'a' is used and as per dynamic scoping this 'a' comes from 'm()' and not the global one. So, "x=x*a" assigns " $2*2 = 4$ " to "x" and 4 is printed. Being passed by reference, "a" in m() also get updated to 4. So, D is the answer here.

1 36 votes

-- Arjun Suresh (287k points)

2.15.3 Parameter Passing: GATE1988-2xv [top](#)

<http://gateoverflow.in/94333>

let variable "a" has address 100 and "b" has 200 .

and avariable in which "a+b" is stored has address 300.

now p(300,100,100) which represent x,y,z

y:=y+1 // it makes a=3;

z:=x+x // x means the value contained at address 300 i.e. 5

$5+5 = 10$ hence value at address 100 i.e. variable "a" will get the value 10 .

hence value of a i.e. 10 will be printed.

1 2 votes

-- psb (309 points)

2.15.4 Parameter Passing: GATE1991-09a [top](#)

<http://gateoverflow.in/536>



Selected Answer

1. Pass by value: Function cannot modify a variable in the calling function. So,

$x = 1, z = 100$

2. Pass by reference: An alias of the variable (a different name but having same memory location) is used to pass the variable to a function. So, whatever change occurs for the variable in the called function is reflected in the calling function.

$x = 2, z = 7 (2 + 5)$

3. Pass by name: The expression used to call a function is copy pasted for each formal parameter. So, the body of P becomes,

```
x := 2;
z := x + x*y;
```

So, printed value will be

$x = 2, z = 12$

1 10 votes

-- Arjun Suresh (287k points)

2.15.5 Parameter Passing: GATE1991_03,x [top](#)

<http://gateoverflow.in/524>



Selected Answer

A is TRUE. Left recursive grammars if used directly in recursive descent parsing causes an infinite loop. So, left recursion must be removed before giving to a recursive descent parser.

B is a strong statement- but I do not have any proof or reference for this- so for now I consider this FALSE.

C is false. The language can have dynamic datatypes which requires dynamically growing memory when data type size increases.

D is true and using macro we can do this.

E - out of syllabus now,

4 votes

-- Arjun Suresh (287k points)

2.15.6 Parameter Passing: GATE2004-2,ISRO2017-54 [top](#)

<http://gateoverflow.in/999>



Selected Answer

ans (d).

option a will not swap the values bcoz it is passed by value...

option b will not swap the value

```
void swap(int a, int b) in which arguments will not take
```

option c is false , given reason is wrong

and option d is correct .. we cant use swap(x,y) bcoz it is pass value function call which will not swap the values

10 votes

-- sonam vyas (12.8k points)

2.15.7 Parameter Passing: GATE2007-IT-33 [top](#)

<http://gateoverflow.in/3466>



Selected Answer

Correct answer is (d)

CALL BY VALUE :- i as global variable declared. Then in main() a local variable j as integer declared i.e j=60 And global variable i initialized to 50 by i=50. Now procedure f called and values of i and j are passed to it. i.e., in f(i,j) -> f(x, y) content of memory location of i (here 50) is copied to memory location of x (which is different from i) and content of memory location of j (here, 60) is copied to memory location of y. Then in f(x,y) i=100 changes the global i to 100, X= 10 changes the local X from 50 to 10 and Y= y+ i means y=60+100=160. Now when return back to main, i and j will be 100 and 60 respectively.

CALL BY REFERENCE:- Now procedure f called and passed reference of i and j to it. i.e., in f(i,j) -> f(x, y) x and y are new names (aliases) pointing to the same memory location of i and j respectively. So, i = 100 changes the global i to 100 and x= 10 means x as well as global i =10 (as the i being passed is the global variable and x and i share the same address).

y= y+ i means y = 60+10=70 and this changes the value of j also to 70 as j and y have the same address. Now when return back to main, i and j will be 10 and 70 respectively.

15 votes

-- Kalpana Bhargav (3.2k points)

2.15.8 Parameter Passing: GATE2013-42 [top](#)

<http://gateoverflow.in/60>



Selected Answer

In GATE 2013 marks were given to all as the same code in C/C++ produces undefined behavior. This is because * is not a

sequence point in C/C++. The correct code must replace
 return $f(x,c) * x;$
 with
 $res = f(x,c);$
 return $res * x;$

In this code, there will be 4 recursive calls with parameters (6,4), (7,3), (8,2) and (9,1). The last call returns 1. But due to pass by reference, x in all the previous functions is now 9. Hence, the value returned by $f(p,p)$ will be $9 * 9 * 9 * 9 * 1 = 6561$.

Good Read

- <http://stackoverflow.com/questions/41775973/is-this-undefined-behaviour-in-c-if-not-predict-the-output- logically>
- <http://gateoverflow.in/108445/c-programming?show=108582#a108582>

24 votes

-- Arjun Suresh (287k points)

2.16

Parsing(42) top

2.16.1 Parsing: GATE 2016-1-45 top

<http://gateoverflow.in/39697>

The attribute of three arithmetic operators in some programming language are given below.

OPERATOR	PRECEDENCE	ASSOCIATIVITY	ARITY
+	high	Left	Binary
-	Medium	Right	Binary
*	Low	Left	Binary

The value of the expression $2 - 5 + 1 - 7 * 3$ in this language is _____.

[gate2016-1](#) [compiler-design](#) [parsing](#) [normal](#) [numerical-answers](#)

Answer

2.16.2 Parsing: GATE1987-1-xiv top

<http://gateoverflow.in/80295>

An operator precedence parser is a

- Bottom-up parser.
- Top-down parser.
- Back tracking parser.
- None of the above.

[gate1987](#) [compiler-design](#) [parsing](#)

Answer

2.16.3 Parsing: GATE1989-1-iii top

<http://gateoverflow.in/87046>

Merging states with a common core may produce _____ conflicts and does not produce _____ conflicts in an LALR parser.

[gate1989](#) [descriptive](#) [compiler-design](#) [parsing](#)

Answer

2.16.4 Parsing: GATE1992_02,xiii top

<http://gateoverflow.in/570>

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

For a context free grammar, FOLLOW(A) is the set of terminals that can appear immediately to the right of non-terminal A in some "sentential" form. We define two sets LFOLLOW(A) and RFOLLOW(A) by replacing the word "sentential" by "left

sentential" and "right most sentential" respectively in the definition of FOLLOW (A).

- (a). FOLLOW(A) and LFOLLOW(A) may be different.
- (b). FOLLOW(A) and RFOLLOW(A) are always the same.
- (c). All the three sets are identical.
- (d). All the three sets are different.

[gate1992](#) [parsing](#) [compiler-design](#) [normal](#)

[Answer](#)

2.16.5 Parsing: GATE1992_02,xiv [top](#)

<http://gateoverflow.in/571>

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

Consider the *SLR(1)* and *LALR(1)* parsing tables for a context free grammar. Which of the following statement is/are true?

- (a). The *goto* part of both tables may be different.
- (b). The *shift* entries are identical in both the tables.
- (c). The *reduce* entries in the tables may be different.
- (d). The *error* entries in tables may be different

[gate1992](#) [compiler-design](#) [normal](#) [parsing](#)

[Answer](#)

2.16.6 Parsing: GATE1998-1.26 [top](#)

<http://gateoverflow.in/1663>

Which of the following statements is true?

- A. SLR parser is more powerful than LALR
- B. LALR parser is more powerful than Canonical LR parser
- C. Canonical LR parser is more powerful than LALR parser
- D. The parsers SLR, Canonical CR, and LALR have the same power

[gate1998](#) [compiler-design](#) [parsing](#) [normal](#)

[Answer](#)

2.16.7 Parsing: GATE1998_1.27 [top](#)

<http://gateoverflow.in/1664>

Type checking is normally done during

- (a) lexical analysis
- (b) syntax analysis
- (c) syntax directed translation
- (d) code optimization

[gate1998](#) [compiler-design](#) [parsing](#) [easy](#)

[Answer](#)

2.16.8 Parsing: GATE1998_22 [top](#)

<http://gateoverflow.in/1737>

- a. An identifier in a programming language consists of up to six letters and digits of which the first character must be a

letter. Derive a regular expression for the identifier.

- b. Build an LL(1) parsing table for the language defined by the LL(1) grammar with productions

$\text{Program} \rightarrow \text{begin } d \text{ semi } X \text{ end}$

$X \rightarrow d \text{ semi } X \mid sY$

$Y \rightarrow \text{semi } sY \mid \epsilon$

gate1998 | compiler-design | parsing | descriptive

Answer

2.16.9 Parsing: GATE1999_1.17 [top](#)

<http://gateoverflow.in/1470>

Which of the following is the most powerful parsing method?

- A. LL (1)
- B. Canonical LR
- C. SLR
- D. LALR

gate1999 | compiler-design | parsing | easy

Answer

2.16.10 Parsing: GATE2000-1.19, UGCNET-Dec2013-II-30 [top](#)

<http://gateoverflow.in/642>

Which of the following derivations does a top-down parser use while parsing an input string? The input is assumed to be scanned in left to right order.

- A. Leftmost derivation
- B. Leftmost derivation traced out in reverse
- C. Rightmost derivation
- D. Rightmost derivation traced out in reverse

gate2000 | compiler-design | parsing | normal | ugcnetdec2013ii

Answer

2.16.11 Parsing: GATE2001-16 [top](#)

<http://gateoverflow.in/757>

Consider the following grammar with terminal alphabet $\Sigma\{a, (,), +, *\}$ and start symbol E . The production rules of the grammar are:

$$E \rightarrow aA$$

$$E \rightarrow (E)$$

$$A \rightarrow +E$$

$$A \rightarrow * E$$

$$A \rightarrow \epsilon$$

- a. Compute the FIRST and FOLLOW sets for E and A .
- b. Complete the LL(1) parse table for the grammar.

gate2001 | compiler-design | parsing | normal

Answer

2.16.12 Parsing: GATE2002-22 [top](#)

<http://gateoverflow.in/875>

- a. Construct all the parse trees corresponding to $i + j * k$ for the grammar

$E \rightarrow E+E$
 $E \rightarrow E*E$
 $E \rightarrow id$

- b. In this grammar, what is the precedence of the two operators * and +?
 c. If only one parse tree is desired for any string in the same language, what changes are to be made so that the resulting LALR(1) grammar is unambiguous?

gate2002 | compiler-design | parsing | normal | descriptive

Answer

2.16.13 Parsing: GATE2003-16 [top](#)

<http://gateoverflow.in/906>

Which of the following suffices to convert an arbitrary CFG to an LL(1) grammar?

- A. Removing left recursion alone
- B. Factoring the grammar alone
- C. Removing left recursion and factoring the grammar
- D. None of the above

gate2003 | compiler-design | parsing | easy

Answer

2.16.14 Parsing: GATE2003-17 [top](#)

<http://gateoverflow.in/907>

Assume that the SLR parser for a grammar G has n_1 states and the LALR parser for G has n_2 states. The relationship between n_1 and n_2 is

- A. n_1 is necessarily less than n_2
- B. n_1 is necessarily equal to n_2
- C. n_1 is necessarily greater than n_2
- D. None of the above

gate2003 | compiler-design | parsing | easy

Answer

2.16.15 Parsing: GATE2003-57 [top](#)

<http://gateoverflow.in/945>

Consider the grammar shown below.

$S \rightarrow C\ C$

$C \rightarrow c\ C \mid d$

This grammar is

- A. LL(1)
- B. SLR(1) but not LL(1)
- C. LALR(1) but not SLR(1)
- D. LR(1) but not LALR(1)

gate2003 | compiler-design | grammar | parsing | normal

Answer

2.16.16 Parsing: GATE2005-14 [top](#)

<http://gateoverflow.in/1350>

The grammar $A \rightarrow AA \mid (A) \mid \epsilon$ is not suitable for predictive-parsing because the grammar is:

- A. ambiguous
- B. left-recursive
- C. right-recursive

- D. an operator-grammar

[gate2005](#) [compiler-design](#) [parsing](#) [grammar](#) [easy](#)

[Answer](#)

2.16.17 Parsing: GATE2005-60 [top](#)

<http://gateoverflow.in/1383>

Consider the grammar:

$$S \rightarrow (S) \mid a$$

Let the number of states in SLR(1), LR(1) and LALR(1) parsers for the grammar be n_1, n_2 and n_3 respectively. The following relationship holds good:

- A. $n_1 < n_2 < n_3$
- B. $n_1 = n_3 < n_2$
- C. $n_1 = n_2 = n_3$
- D. $n_1 \geq n_3 \geq n_2$

[gate2005](#) [compiler-design](#) [parsing](#) [normal](#)

[Answer](#)

2.16.18 Parsing: GATE2005-83a [top](#)

<http://gateoverflow.in/1405>

Statement for Linked Answer Questions 83a & 83b:

Consider the following expression grammar. The semantic rules for expression evaluation are stated next to each grammar production.

$$\begin{array}{ll} E \rightarrow \text{number} & E.\text{val} = \text{number.val} \\ | E + E & E^{(1)}.\text{val} = E^{(2)}.\text{val} \\ | E \times E & \quad \quad \quad + E^{(3)}.\text{val} \\ & E^{(1)}.\text{val} = E^{(2)}.\text{val} \\ & \quad \quad \quad \times E^{(3)}.\text{val} \end{array}$$

The above grammar and the semantic rules are fed to a *yaac* tool (which is an LALR(1) parser generator) for parsing and evaluating arithmetic expressions. Which one of the following is true about the action of *yaac* for the given grammar?

- A. It detects *recursion* and eliminates recursion
- B. It detects *reduce-reduce* conflict, and resolves
- C. It detects *shift-reduce* conflict, and resolves the conflict in favor of a *shift* over a *reduce* action
- D. It detects *shift-reduce* conflict, and resolves the conflict in favor of a *reduce* over a *shift* action

[gate2005](#) [compiler-design](#) [parsing](#) [normal](#)

[Answer](#)

2.16.19 Parsing: GATE2005-83b [top](#)

<http://gateoverflow.in/8703>

Consider the following expression grammar. The semantic rules for expression evaluation are stated next to each grammar production.

$$\begin{array}{ll} E \rightarrow \text{number} & E.\text{val} = \text{number.val} \\ | E E^{(1)}\text{Eval} = E^{(2)}.\text{val} + E^{(3)}.\text{val} \\ | E E^{(4)}\text{Eval} = E^{(2)}.\text{val} \times E^{(3)}.\text{val} \end{array}$$

Assume the conflicts of this question are resolved using yacc tool and an LALR(1) parser is generated for parsing arithmetic expressions as per the given grammar. Consider an expression $3 \times 2 + 1$. What precedence and associativity properties does the generated parser realize?

- A. Equal precedence and left associativity; expression is evaluated to 7
- B. Equal precedence and right associativity; expression is evaluated to 9

- C. Precedence of ' \times ' is higher than that of '+', and both operators are left associative; expression is evaluated to 7
 D. Precedence of '+' is higher than that of ' \times ', and both operators are left associative; expression is evaluated to 9

gate2005 compiler-design parsing normal

[Answer](#)

2.16.20 Parsing: GATE2005-IT-83a [top](#)

<http://gateoverflow.in/3849>

Consider the context-free grammar

$$\begin{aligned} E &\rightarrow E + E \\ E &\rightarrow (E * E) \\ E &\rightarrow id \end{aligned}$$

where E is the starting symbol, the set of terminals is {id, (, +,), *}, and the set of nonterminals is {E}.

Which of the following terminal strings has more than one parse tree when parsed according to the above grammar?

- A. id + id + id + id
 B. id + (id* (id * id))
 C. (id* (id * id)) + id
 D. ((id * id + id) * id)

gate2005-it compiler-design grammar parsing easy

[Answer](#)

2.16.21 Parsing: GATE2005-IT-83b [top](#)

<http://gateoverflow.in/3850>

Consider the context-free grammar

$$\begin{aligned} E &\rightarrow E + E \\ E &\rightarrow (E * E) \\ E &\rightarrow id \end{aligned}$$

where E is the starting symbol, the set of terminals is {id, (, +,), *}, and the set of non-terminals is {E}.

For the terminal string id + id + id + id, how many parse trees are possible?

- A. 5
 B. 4
 C. 3
 D. 2

gate2005-it compiler-design parsing normal

[Answer](#)

2.16.22 Parsing: GATE2006-58 [top](#)

<http://gateoverflow.in/1836>

Consider the following grammar:

$$S \rightarrow FR$$

$$R \rightarrow *S \mid \epsilon$$

$$F \rightarrow id$$

In the predictive parser table, M, of the grammar the entries M[S,id] and M[R,\$] respectively are

- A. $\{S \rightarrow FR\}$ and $\{R \rightarrow \epsilon\}$
 B. $\{S \rightarrow FR\}$ and $\{\}$
 C. $\{S \rightarrow FR\}$ and $\{R \rightarrow *S\}$
 D. $\{F \rightarrow id\}$ and $\{R \rightarrow \epsilon\}$

gate2006 compiler-design parsing normal

[Answer](#)

2.16.23 Parsing: GATE2006-7 [top](#)

<http://gateoverflow.in/888>

Consider the following grammar

S → S * E
 S → E
 E → F + E
 E → F
 F → id

Consider the following LR(0) items corresponding to the grammar above

- i. S → S * .E
- ii. E → F. + E
- iii. E → F + .E

Given the items above, which two of them will appear in the same set in the canonical sets-of-items for the grammar?

- A. i and ii
- B. ii and iii
- C. i and iii
- D. None of the above

[gate2006](#) [compiler-design](#) [parsing](#) [normal](#)

[Answer](#)

2.16.24 Parsing: GATE2007-18 [top](#)

<http://gateoverflow.in/1216>

Which one of the following is a top-down parser?

- A. Recursive descent parser.
- B. Operator precedence parser.
- C. An LR(k) parser.
- D. An LALR(k) parser.

[gate2007](#) [compiler-design](#) [parsing](#) [normal](#)

[Answer](#)

2.16.25 Parsing: GATE2008-11 [top](#)

<http://gateoverflow.in/409>

Which of the following describes a handle (as applicable to LR-parsing) appropriately?

- A. It is the position in a sentential form where the next shift or reduce operation will occur
- B. It is non-terminal whose production will be used for reduction in the next step
- C. It is a production that may be used for reduction in a future step along with a position in the sentential form where the next shift or reduce operation will occur
- D. It is the production p that will be used for reduction in the next step along with a position in the sentential form where the right hand side of the production may be found

[gate2008](#) [compiler-design](#) [parsing](#) [normal](#)

[Answer](#)

2.16.26 Parsing: GATE2008-55 [top](#)

<http://gateoverflow.in/478>

An LALR(1) parser for a grammar G can have shift-reduce (S-R) conflicts if and only if

- A. The SLR(1) parser for G has S-R conflicts
- B. The LR(1) parser for G has S-R conflicts
- C. The LR(0) parser for G has S-R conflicts
- D. The LALR(1) parser for G has reduce-reduce conflicts

[gate2008](#) [compiler-design](#) [parsing](#) [normal](#)
Answer

2.16.27 Parsing: GATE2008-IT-79 [top](#)

<http://gateoverflow.in/3393>

A CFG G is given with the following productions where S is the start symbol, A is a non-terminal and a and b are terminals.

$$S \rightarrow aS \mid AA \rightarrow aAb \mid bAa \mid \epsilon$$

For the string "aabbaab" how many steps are required to derive the string and how many parse trees are there?

- A. 6 and 1
- B. 6 and 2
- C. 7 and 2
- D. 4 and 2

[gate2008-it](#) [compiler-design](#) [context-free-language](#) [parsing](#) [normal](#)
Answer

2.16.28 Parsing: GATE2009-42 [top](#)

<http://gateoverflow.in/1328>

Which of the following statements are TRUE?

- I. There exist parsing algorithms for some programming languages whose complexities are less than $\Theta(n^3)$
 - II. A programming language which allows recursion can be implemented with static storage allocation.
 - III. No L-attributed definition can be evaluated in the framework of bottom-up parsing.
 - IV. Code improving transformations can be performed at both source language and intermediate code level.
- A. I and II
 - B. I and IV
 - C. III and IV
 - D. I, III and IV

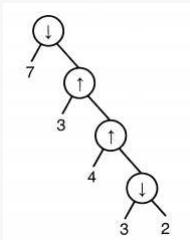
[gate2009](#) [compiler-design](#) [parsing](#) [normal](#)
Answer

2.16.29 Parsing: GATE2011_27 [top](#)

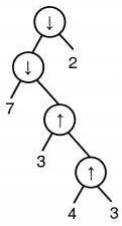
<http://gateoverflow.in/2129>

Consider two binary operators ' \uparrow ' and ' \downarrow ' with the precedence of operator \downarrow being lower than that of the operator \uparrow . Operator \uparrow is right associative while operator \downarrow is left associative. Which one of the following represents the parse tree for expression $(7 \downarrow 3 \uparrow 4 \uparrow 3 \downarrow 2)$

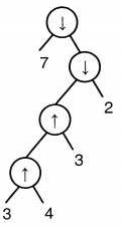
(A)



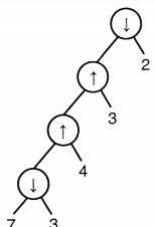
(B)



(C)



(D)



gate2011 compiler-design parsing normal

<http://gateoverflow.in/2181>

For the grammar below, a partial

LL(1) parsing table is also presented along with the grammar. Entries that need to be filled are indicated as **E1**, **E2**, and **E3**.

ε is the empty string, \$ indicates end of input, and, | separates alternate right hand sides of productions.

$$S \rightarrow aAbB \mid bAaB \mid \varepsilon$$

$$A \rightarrow S$$

B → S

	a	b	\$
S	E1	E2	$S \rightarrow \varepsilon$
A	$A \rightarrow S$	$A \rightarrow S$	error
B	$B \rightarrow S$	$B \rightarrow S$	E3

The FIRST and FOLLOW sets for the non-terminals A and B are

(A)

$$\text{FIRST}(A) = \{a, b, \epsilon\} = \text{FIRST}(B)$$

$$\text{FOLLOW}(A) = \{a, b\}$$

$$\text{FOLLOW}(B) = \{a, b, \$\}$$

(B)

$$\text{FIRST}(A) = \{a, b, \$\}$$

$$\text{FIRST}(B) = \{a, b, \epsilon\}$$

$$\text{FOLLOW}(A) = \{a, b\}$$

$$\text{FOLLOW}(B) = \{\$\}$$

(C)

$$\text{FIRST}(A) = \{a, b, \epsilon\} = \text{FIRST}(B)$$

$$\text{FOLLOW}(A) = \{a, b\}$$

$$\text{FOLLOW}(B) = \emptyset$$

(D)

$$\text{FIRST}(A) = \{a, b\} = \text{FIRST}(B)$$

$$\text{FOLLOW}(A) = \{a, b\}$$

$$\text{FOLLOW}(B) = \{a, b\}$$

[gate2012](#) [compiler-design](#) [parsing](#) [normal](#)
Answer

2.16.31 Parsing: GATE2012-53 [top](#)

<http://gateoverflow.in/43312>

For the grammar below, a partial $LL(1)$ parsing table is also presented along with the grammar. Entries that need to be filled are indicated as **E1**, **E2**, and **E3**. ϵ is the empty string, $\$$ indicates end of input, and, $|$ separates alternate right hand sides of productions.

$$S \rightarrow aAbB \mid bAaB \mid \epsilon$$

$$A \rightarrow S$$

$$B \rightarrow S$$

	a	b	\$
S	E1	E2	$S \rightarrow \epsilon$
A	$A \rightarrow S$	$A \rightarrow S$	error
B	$B \rightarrow S$	$B \rightarrow S$	E3

The appropriate entries for E1, E2, and E3 are

(A)

$$\text{E1 : } S \rightarrow aAbB, A \rightarrow S$$

E2 : $S \rightarrow bAaB, B \rightarrow S$

E1 : $B \rightarrow S$

(B)

E1 : $S \rightarrow aAbB, S \rightarrow \epsilon$

E2 : $S \rightarrow bAaB, S \rightarrow \epsilon$

E3 : $S \rightarrow \epsilon$

(C)

E1 : $S \rightarrow aAbB, S \rightarrow \epsilon$

E2 : $S \rightarrow bAaB, S \rightarrow \epsilon$

E3 : $B \rightarrow S$

(D)

E1 : $A \rightarrow S, S \rightarrow \epsilon$

E2 : $B \rightarrow S, S \rightarrow \epsilon$

E3 : $B \rightarrow S$

[normal](#) [gate2012](#) [compiler-design](#) [parsing](#)

[Answer](#)

2.16.32 Parsing: GATE2013_40 [top](#)

<http://gateoverflow.in/1551>

Consider the following two sets of LR(1) items of an LR(1) grammar.

$$\begin{array}{ll} X \rightarrow c.X, c/d & X \rightarrow c.X, \$ \\ X \rightarrow .cX, c/d & X \rightarrow .cX, \$ \\ X \rightarrow .d, c/d & X \rightarrow .d, \$ \end{array}$$

Which of the following statements related to merging of the two sets in the corresponding LALR parser is/are **FALSE**?

1. Cannot be merged since look aheads are different.
 2. Can be merged but will result in S-R conflict.
 3. Can be merged but will result in R-R conflict.
 4. Cannot be merged since goto on c will lead to two different sets.
- (A) 1 only (B) 2 only (C) 1 and 4 only (D) 1, 2, 3 and 4

[gate2013](#) [compiler-design](#) [parsing](#) [normal](#)

[Answer](#)

2.16.33 Parsing: GATE2013_9 [top](#)

<http://gateoverflow.in/1418>

What is the maximum number of reduce moves that can be taken by a bottom-up parser for a grammar with no epsilon and unit-production (i.e., of type $A \rightarrow \epsilon$ and $A \rightarrow a$) to parse a string with n tokens?

- (A) $n/2$
 (B) $n - 1$
 (C) $2n - 1$

(D) 2^n [gate2013](#) [compiler-design](#) [parsing](#) [normal](#)[Answer](#)**2.16.34 Parsing: GATE2014-1-34** [top](#)<http://gateoverflow.in/1807>

A canonical set of items is given below

$$S \rightarrow L. > R$$

$$Q \rightarrow R.$$

On input symbol < the set has

- A. a shift-reduce conflict and a reduce-reduce conflict.
- B. a shift-reduce conflict but not a reduce-reduce conflict.
- C. a reduce-reduce conflict but not a shift-reduce conflict.
- D. neither a shift-reduce nor a reduce-reduce conflict.

[gate2014-1](#) [compiler-design](#) [parsing](#) [normal](#)[Answer](#)**2.16.35 Parsing: GATE2015-3_16** [top](#)<http://gateoverflow.in/8413>

Among simple LR (SLR), canonical LR, and look-ahead LR (LALR), which of the following pairs identify the method that is very easy to implement and the method that is the most powerful, in that order?

- A. SLR, LALR
- B. Canonical LR, LALR
- C. SLR, canonical LR
- D. LALR, canonical LR

[gate2015-3](#) [compiler-design](#) [parsing](#) [normal](#)[Answer](#)**2.16.36 Parsing: GATE2015-3_31** [top](#)<http://gateoverflow.in/8488>

Consider the following grammar G

$$S \rightarrow F \mid H$$

$$F \rightarrow p \mid c$$

$$H \rightarrow d \mid c$$

Where S , F , and H are non-terminal symbols, p , d , and c are terminal symbols. Which of the following statement(s) is/are correct?

S1: LL(1) can parse all strings that are generated using grammar G

S2: LR(1) can parse all strings that are generated using grammar G

- A. Only S1
- B. Only S2
- C. Both S1 and S2
- D. Neither S1 and S2

[gate2015-3](#) [compiler-design](#) [parsing](#) [normal](#)[Answer](#)**2.16.37 Parsing: GATE2017-1-17** [top](#)<http://gateoverflow.in/118297>

Consider the following grammar:

$P \rightarrow xQRS$
 $Q \rightarrow yz|z$
 $R \rightarrow w|\epsilon$
 $S \rightarrow y$

What is FOLLOW(Q)?

- (A) $\{R\}$ (B) $\{w\}$ (C) $\{w, y\}$ (D) $\{w, \$\}$

[gate2017-1](#) [compiler-design](#) [parsing](#)
[Answer](#)

2.16.38 Parsing: GATE2017-1-43 [top](#)

<http://gateoverflow.in/118326>

Consider the following grammar:

- stmt \rightarrow if expr then expr else expr; stmt | 0
- expr \rightarrow term relop term | term
- term \rightarrow id | number
- id \rightarrow a | b | c
- number \rightarrow [0-9]

where **relop** is a relational operator (e.g.. < . > ...). 0 refers to the empty statement, and **if**, **then**, **else** are terminals. Consider a program P following the above grammar containing ten **if** terminals. The number of control flow paths in P is_____ . For example. the program

if e₁ **then** e₂ **else** e₃

 has 2 control flow paths. e₁ \rightarrow e₂ and e₁ \rightarrow e₃.

[gate2017-1](#) [compiler-design](#) [parsing](#) [normal](#) [numerical-answers](#)
[Answer](#)

2.16.39 Parsing: GATE2017-2-6 [top](#)

<http://gateoverflow.in/118343>

Which of the following statements about parser is/are CORRECT?

- I. Canonical LR is more powerful than SLR
- II. SLR is more powerful than LALR
- III. SLR is more powerful than Canonical LR

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

[gate2017-2](#) [compiler-design](#) [parsing](#)
[Answer](#)

2.16.40 Parsing: TIFR2012-B-17 [top](#)

<http://gateoverflow.in/25215>

 Which of the following correctly describes $LR(k)$ parsing?

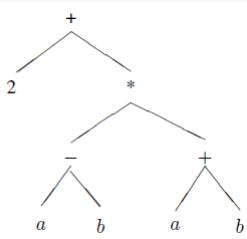
- a. The input string is alternately scanned left to right and right to left with k reversals.
- b. Input string is scanned once left to right with rightmost derivation and k symbol look-ahead.
- c. $LR(k)$ grammars are expressively as powerful as context-free grammars.
- d. Parser makes k left-to-right passes over input string.
- e. Input string is scanned from left to right once with k symbol to the right as look-ahead to give left-most derivation.

[tifr2012](#) [compiler-design](#) [parsing](#)
[Answer](#)

2.16.41 Parsing: TIFR2012-B-8 [top](#)

<http://gateoverflow.in/25108>

Consider the parse tree



Assume that $*$ has higher precedence than $+$, $-$ and operators associate right to left (i.e. $(a + b + c) = (a + (b + c))$). Consider

- (i) $2 + a - b$
- (ii) $2 + a - b * a + b$
- (iii) $2 + ((a - b) * (a + b))$
- (iv) $2 + (a - b) * (a + b)$

The parse tree corresponds to

- a. Expression (i)
- b. Expression (ii)
- c. Expression (iv) only
- d. Expression (ii), (iii), and (iv)
- e. Expression (iii) and (iv) only

tifr2012 | compiler-design | parsing

[Answer](#)

2.16.42 Parsing: TIFR2015-B-15 [top](#)

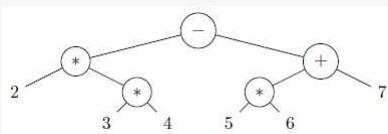
<http://gateoverflow.in/30079>

Consider the following grammar (the start symbol is E) for generating expressions.

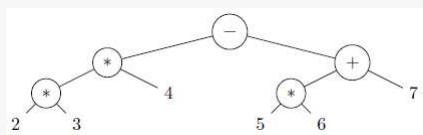
$$\begin{aligned} E &\rightarrow T - E | T + E | T \\ T &\rightarrow T * F | F \\ F &\rightarrow 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 \end{aligned}$$

With respect to this grammar, which of the following trees is the valid evaluation tree for the expression $2 * 3 * 4 - 5 * 6 + 7$?

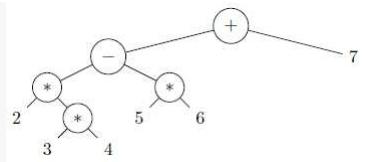
(a)



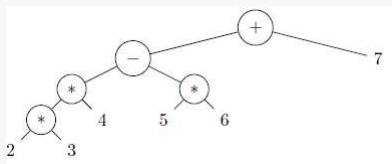
(b)



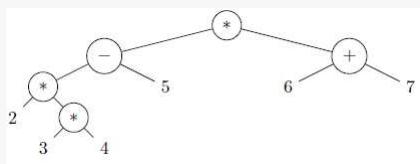
(c)



(d)



(e)



tifr2015 parsing

Answer

Answers: Parsing

2.16.1 Parsing: GATE 2016-1-45 [top](#)

<http://gateoverflow.in/39697>

Selected Answer

$2 - 5 + 1 - 7 * 3$ will be evaluated according to the precedence and associativity as given in the question as follows:
 $((2 - ((5 + 1) - 7)) * 3) \Rightarrow ((2 - (-1)) * 3) \Rightarrow 9$

1 31 votes

-- Monanshi Jain (8.1k points)

2.16.2 Parsing: GATE1987-1-xiv [top](#)

<http://gateoverflow.in/80295>

operator precedence is a bottom up parser based on shift reduce parsers...

1 3 votes

-- kirti singh (3.3k points)

2.16.3 Parsing: GATE1989-1-iii [top](#)

<http://gateoverflow.in/87046>

Merging states with a common core may produce **Reduce - Reduce** conflicts and does not produce **Shift- Reduce** conflicts in an LALR parser.

1 2 votes

-- Prashant Singh (47.6k points)

2.16.4 Parsing: GATE1992_02,xiii [top](#)

<http://gateoverflow.in/570>

Selected Answer

Ans - a,b, LFollow may be different but RFollow and Follow will be same

Consider a Grammar -

$S \rightarrow AB$

A \rightarrow a

B \rightarrow b

Now only string derivable is { ab }.

Let's find Follow(A) in all cases :

i) Follow(A) - set of terminals that can appear immediately to the right of non-terminal A in some "sentential" form

S \rightarrow AB \rightarrow Ab \rightarrow ab

Here, we notice only 'b' can appear to the right of A.

Follow(A) = { b }

ii) LFFollow(A) - set of terminals that can appear immediately to the right of non-terminal A in some "left sentential" form

S \rightarrow AB \rightarrow aB \rightarrow ab

Here, we notice no terminal can appear to the right of A.

LFFollow(A) = {}

iii) RFFollow(A) - set of terminals that can appear immediately to the right of non-terminal A in some "right most sentential" form

S \rightarrow AB \rightarrow Ab \rightarrow ab

Here, we notice only 'b' can appear to the right of A.

RFFollow(A) = { b }

13 votes

-- Himanshu Agarwal (15.9k points)

2.16.5 Parsing: GATE1992_02,xiv [top](#)

<http://gateoverflow.in/571>



Selected Answer

Goto part & shift entry must be same.

Reduce entry & error entry may be different due to conflicts.

11 votes

-- Digvijay (46k points)

2.16.6 Parsing: GATE1998-1.26 [top](#)

<http://gateoverflow.in/1663>



Selected Answer

1. SLR parser is more powerful than LALR. **False**.
2. LALR parser is more powerful than Canonical LR parser. **False**.
3. Canonical LR parser is more powerful than LALR parser. **True**.
4. The parsers SLR, Canonical CR, and LALR have the same power. **False**.

answer - C

8 votes

-- ankitrokdeonsns (9.1k points)

2.16.7 Parsing: GATE1998_1.27 [top](#)

<http://gateoverflow.in/1664>



Selected Answer

The answer is C .

The use of syntax analyser is used to create parse Tree. But along with Grammar as input to Syntax Analyser we add even semantic rules which form the basis of Syntax Directed Translation That help us in Evaluation of Expression .Remember that

Syntax Directed Translation is used in following cases

1. Conversion of infix to Postfix
2. Calculation of infix expression
3. For creating a Acyclic graph
4. Type Checking
5. Conversion of Binary number to Decimal
6. Counting the numbers of bits (0 or 1) in a binary number
7. Creation of syntax tree
8. To generate Intermediate code
9. Storing the data into Symbol table

15 votes

-- spriti1991 (2.1k points)

2.16.8 Parsing: GATE1998_22 [top](#)

<http://gateoverflow.in/173>

a.
(letter)(letter + digit + epsilon)⁵

b.
1. program ---> begin s end
2. X -----> d semi x
3. | sY
4. Y -----> semi sY
5. | epsilon

program	begin	d	semi	s	end	\$
	1					
X		2		3		
Y			4			5

5 votes

-- Digvijay (46k points)

2.16.9 Parsing: GATE1999_1.17 [top](#)

<http://gateoverflow.in/1470>



Canonical LR is most powerful method

LR>LALR>SLR

so ans is b

4 votes

-- Pooja Palod (31.2k points)

2.16.10 Parsing: GATE2000-1.19, UGCNET-Dec2013-II-30 [top](#)

<http://gateoverflow.in/642>



ans a)

6 votes

-- Aditi Dan (5.3k points)

2.16.11 Parsing: GATE2001-16 [top](#)

<http://gateoverflow.in/757>



Selected Answer

First (E) = { a,(}

First (A) = { +,*,ε }

Follow (E) = Follow (A) = { \$,) }

LL(1) Parsing Table :

	a	()	+	*	\$
E	E->aA	E->(E)				
A		A->ε	A->+E	A->*E	A->ε	

10 votes

-- Aditya Gaurav (2.8k points)

2.16.12 Parsing: GATE2002-22 [top](#)

<http://gateoverflow.in/875>



Selected Answer

- a. two parse tree for i+j*k.
- b. + and * having same precedence..
- c. to make grammer LALR compatible give priority to + over * or vice versa.
following grammar is LALR(1)

E -----> E + T
 | T
 T -----> T * F
 | F
 F -----> id

10 votes

-- Digvijay (46k points)

2.16.13 Parsing: GATE2003-16 [top](#)

<http://gateoverflow.in/906>



Selected Answer

LL(1) parser is top down parser.

For top down parsers, the grammar should be unambiguous, deterministic and should not be left recursive.

All the 3 conditions must be satisfied for LL(1) parsers.

Now, even if all 3 conditions are satisfied we cannot get an LL(1) or even LL(k) (for any k) grammar for even a DCFG. This is because there are DCFLs which does not have an LL(k) grammar (see ref below). On the other hand for any DCFL, we can always have an LR(1) grammar.

<http://mathoverflow.net/questions/31733/can-i-have-an-ll-grammar-for-every-deterministic-context-free-language>

So, option D is correct.

17 votes

-- Monanshi Jain (8.1k points)

2.16.14 Parsing: GATE2003-17 [top](#)

<http://gateoverflow.in/907>



Selected Answer

no of states in slr and lalr are equal

and no of states in slr and lalr are less than or equal to lr(1)

9 votes

-- Pooja Palod (31.2k points)

2.16.15 Parsing: GATE2003-57 [top](#)



Selected Answer

ans is a

 $\text{First}(S) = \text{First}(C) = \{c, d\}$

there are no multiple in single row of parsing table hence grammar is LL1

note : if we have $A \rightarrow B/C$ for grammar to be LL(1) $\text{first}(B) \cap \text{First}(C)$ should be null otherwise grammar is not LL1. If $\text{First}(B)$ contains epsilon then $\text{Follow}(A) \cap \text{First}(C)$ should be null. Using this we can say grammar is LL(1) or not without constructing parsing table.

An ϵ free LL(1) grammar is also SLR(1) and hence LALR(1) and LR(1) too.

11 votes

-- Pooja Palod (31.2k points)

2.16.16 Parsing: GATE2005-14 [top](#)



Selected Answer

both A and B can be answers but A is a better answer. Because we have standard procedure for removing left-recursion but ambiguity is not easy to remove. - checking if a given CFG is ambiguous is an undecidable problem.

15 votes

-- Vikrant Singh (13.4k points)

2.16.17 Parsing: GATE2005-60 [top](#)



Selected Answer

ans b)

Both in SLR(1) and LALR(1), states are the LR(0) items while in LR(1) the states are LR(1) set of items. Number of LR(0) items can never be greater than number of LR(1) items. So, $n_1 = n_3 \leq n_2$, B choice. If we construct the states for the grammar we can replace \leq with $<$.

11 votes

-- Aditi Dan (5.3k points)

2.16.18 Parsing: GATE2005-83a [top](#)



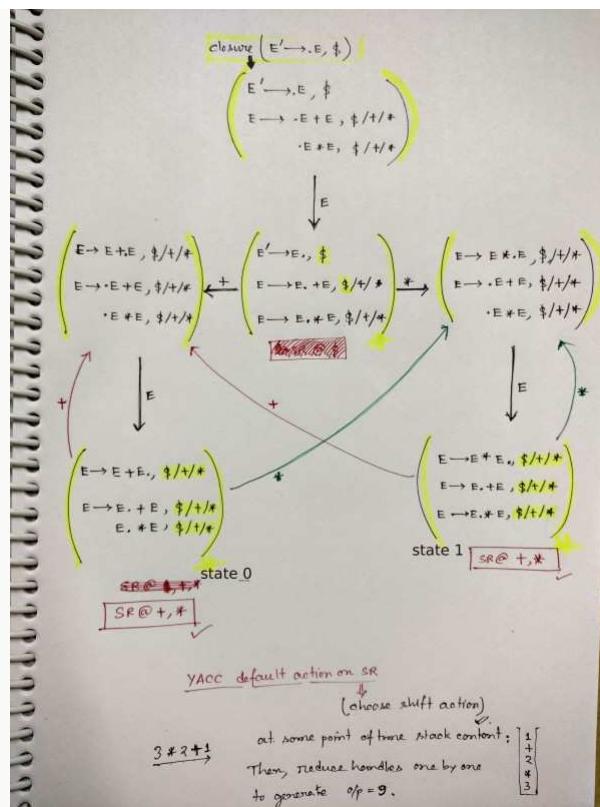
Selected Answer

Given grammar :

$$\begin{aligned} E &\rightarrow \text{num} \\ E &\rightarrow E + E \quad | \quad E * E \end{aligned}$$

First $LR(1)$ item

$$E' \rightarrow \bullet E, \$$$



- num does not create any conflict.
- Additionally here no states differ by lookahead symbols only.
- \Rightarrow LALR(1) and LR(1) tables are same.
- LR(1) table only for state0 and state1:

	+	*	\$
0	SR	SR	R
1	SR	SR	R

So total $2 + 2 = 4$ SR conflict originated in two states of the DFA.

- **Shift-reduce conflict:** Yacc's default action in the case of a shift-reduce conflict is to choose the shift action.
- **Reduce-reduce conflict :** Yacc's default action in the case of a reduce-reduce conflict is to reduce using the production that comes first, textually, in the input grammar specification.

and LEX-YACC-gcc output after implementing the given grammar :

```

exp : number      ($$ = $1;     printf("\t\tstep = %d : reduction,(E->num) %d->%d\n",++step,$$, $1);)
| exp '+' exp   ($$ = $1 + $3; printf("\t\tstep = %d : reduction,(E->E+E) %d->%d\n",++step,$$, $1, $3);)
| exp '*' exp   ($$ = $1 * $3; printf("\t\tstep = %d : reduction,(E->E*E) %d->%d\n",++step,$$, $1, $3);)
| identifier     ($$ = symbolVal($1);)
;

/* C code */

```

```

x - o dpD:~/Desktop/Gate_C_programs/yacc
d@D:~/Desktop/Gate_C_programs/yacc$ yacc -d calc.y
calc.y: warning: 4 shift/reduce conflicts [-Wconflicts-sr]
d@D:~/Desktop/Gate_C_programs/yacc$ lex calc.l
d@D:~/Desktop/Gate_C_programs/yacc$ gcc lex.yy.c y.tab.c -o calc
d@D:~/Desktop/Gate_C_programs/yacc$ ./calc

a = 3*2+1;
step = 1 : i/p scan: 3

step = 2 : reduction,(E->num) 3->3

step = 3 : i/p scan: *

step = 4 : i/p scan: 2

step = 5 : reduction,(E->num) 2->2

step = 6 : i/p scan: +

step = 7 : i/p scan: 1

step = 8 : reduction,(E->num) 1->1

step = 9 : i/p scan: ;

step = 10 : reduction,(E->E+E) 3->2+1

step = 11 : reduction,(E->E*E) 9->3*3

print a;
step = 12 : i/p scan: ;

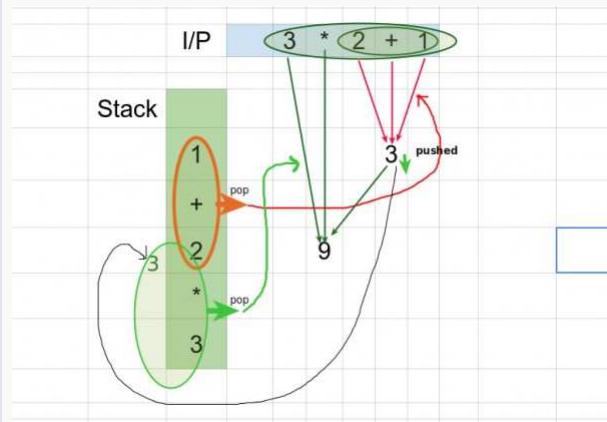
Printing 9

```

As we can see from the output reduction on $E \rightarrow \text{num}$ is carried out as soon as top of stack contains a num. So, no conflict related to $E \rightarrow \text{num}$.

one example : Because of YACC shift preference, even if $3 * 2$ ($E * E$) handle found on top of the stack at some point of time, it will shift on reading + instead of reducing with $E \rightarrow E * E$. In this way, the complete input will be pushed into the stack. After that only reduce work starts as shown below.

- **Equal precedence** because of the given grammar $E \rightarrow E + E \mid E * E$, (single level)
- and **Right associativity** :



How YACC handles conflicts

23 votes

-- Debashish Deka (49.1k points)

Adding Nandan Jha answer....

Date :

(A) Here, in LALR(1), we might end up with these productions

$E \rightarrow E + E$ (lookheads) We have
 $E \rightarrow E \cdot E$ (lookheads) SR conflict
 $E \rightarrow E \cdot * E$ (lookheads)

YACC favours shift move in case of SR conflict

option (C) ✓

(B) First of all, it is ambiguous grammar.
Hence, equal precedence and associativity
Now, as YACC resolved it with shift move
we will shift until the last operator
and then we will start reducing.

$3 \times 2 + 1$

↓
3
9 → Hence, it gets right associativity

option (b) ✓

A HCI

21 votes

-- Kathleen Bankson (59.4k points)

2.16.19 Parsing: GATE2005-83b [top](#)



Selected Answer

LALR Parser is type of **Bottom up Parser** which uses **Right most Derivation**

For $3 \times 2 + 1$

$E \rightarrow E * E$ (Both shift and reduce possible but yacc prefers shift)

$\rightarrow E * E + E$
 $\rightarrow E * E + 1$
 $\rightarrow E * 2 + 1$
 $\rightarrow E * 3$
 $\rightarrow 3 * 3$
 $\rightarrow 9$

All the productions are in same level therefore all have same precedence

Therefore Ans is **B. Equal precedence and right associativity; expression is evaluated to 9**

7 votes

-- Prajwal Bhat (11.6k points)

2.16.20 Parsing: GATE2005-IT-83a [top](#)



Selected Answer

Answer is A.

13 votes

-- Gate Keeda (18.8k points)

2.16.21 Parsing: GATE2005-IT-83b [top](#)

<http://gateoverflow.in/3850>



Selected Answer

5 parse trees are possible.

8 votes

-- ujjwal saini (287 points)

2.16.22 Parsing: GATE2006-58 [top](#)



Selected Answer

First $S = \{ id \}$
Follow $R = \{ \$ \}$

so $M[S,id] = S \rightarrow FR$
 $M[S,\$] = R \rightarrow \epsilon$

So ans is A

10 votes

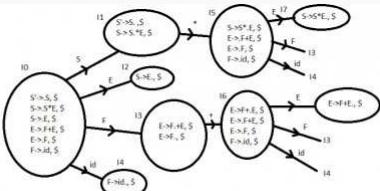
-- Pooja Palod (31.2k points)

2.16.23 Parsing: GATE2006-7 [top](#)



Selected Answer

ans is D.



11 votes

-- jayendra (7.7k points)

2.16.24 Parsing: GATE2007-18 [top](#)



Selected Answer

1. Recursive descent parser-TOP DOWN PARSER
2. Operator precedence parser-BOTTOM UP PARSER
3. An LR(k) parser.-BOTTOM UP PARSER
4. An LALR(k) parser-BOTTOM UP PARSER

3 votes

-- vnc (6.1k points)

2.16.25 Parsing: GATE2008-11 [top](#)



Selected Answer

A *sentential form* is the start symbol S of a grammar or any string in $(V \cup T)^*$ that can be derived from S .

Consider the linear grammar

$\{S, B\}, \{a, b\}, S, \{S \rightarrow aS, S \rightarrow B, B \rightarrow bB, B \rightarrow$

$\lambda\}$.

A derivation using this grammar might look like this:

$$S \Rightarrow aS \Rightarrow aB \Rightarrow abB \Rightarrow abbB \Rightarrow abb$$

Each of {S, aS, aB, abB, abb} is a sentential form.

Because this grammar is linear, each sentential form has at most one variable. Hence there is never any choice about which variable to expand next.

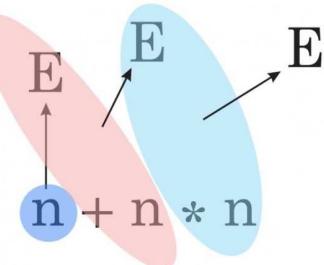
Here, in option D the sentential forms are same but generated differently coz we are using here Bottom Up production.

Handle:

for example the grammar is:

$$\begin{aligned} E &\rightarrow E + n \\ E &\rightarrow E * n \\ E &\rightarrow n \end{aligned}$$

then say to derive string
 $n + n * n$:



these are three different handles shown in 3 different colors =
 $\{n, E + n, E * n\}$

that's what **option D** says

17 votes

-- Amar Vashishth (27.9k points)

2.16.26 Parsing: GATE2008-55 [top](#)

<http://gateoverflow.in/478>



Selected Answer

Both LALR(1) and LR(1) parser uses LR(1) set of items to form their parsing tables. And LALR(1) states can be find by merging LR(1) states of LR(1) parser that have the same set of first components of their items.

i.e. if LR(1) parser has 2 states I and J with items A->a.bP,x and A->a.bP,y respectively, where x and y are look ahead symbols, then as these items are same with respect to their first component, they can be merged together and form one single state, let's say K. Here we have to take union of look ahead symbols. After merging, State K will have one single item as A->a.bP,x,y . This way LALR(1) states are formed (i.e. after merging the states of LR(1)).

Now, S-R conflict in LR(1) items can be there whenever a state has items of the form :

A-> a.bB , p
C-> d. , b

i.e. it is getting both shift and reduce at symbol b,
hence a conflict.

Now, as LALR(1) have items similar to LR(1) in terms of their first component, shift-reduce form will only take place if it is already there in LR(1) states. If there is no S-R conflict in LR(1) state it will never be reflected in the LALR(1) state obtained by combining LR(1) states.

2 votes

-- Madhab Paul Choudhury (4.8k points)

Answer is B.

11 votes

-- Gate Keeda (18.8k points)

2.16.27 Parsing: GATE2008-IT-79 [top](#)

<http://gateoverflow.in/3393>



Selected Answer

$S \rightarrow aS$	1
$S \rightarrow aA$	2
$S \rightarrow aaAb$	3
$S \rightarrow aabAab$	4
$S \rightarrow aabbAaab$	5
$S \rightarrow aabbaab$	6

Thus 6 steps are needed and only one way to derive the string so only one parse tree.

Upvote 20 votes

-- Shreyans Dhankhar (2.5k points)

2.16.28 Parsing: GATE2009-42 [top](#)



Answer is B.

A: Yes there does exist parsing algos less than $\Theta(n^3)$.

B: It cannot be implemented with static storage allocation. It needs dynamic memory allocation.

C: If the L-attributed definitions contain synthesized attributes then it can be evaluated.

D: True.

Upvote 12 votes

-- Gate Keeda (18.8k points)

2.16.29 Parsing: GATE2011_27 [top](#)



Answer is B.

To make the parse tree start compiling the identifiers into blocks based on associativity and precedence.

Grouping: $(7 \downarrow (3 \uparrow (4 \uparrow 3))) \downarrow 2$

Tree can be made by opening inner braces and move towards outer braces.

Upvote 17 votes

-- Sona Praneeth Akula (4k points)

2.16.30 Parsing: GATE2012-52 [top](#)



$\text{First}(S) = \text{First}(A) = \text{First}(B) = \{a, b, \epsilon\}$

$\text{Follow}(A) = \{a, b\}$

$\text{Follow}(B) = \text{Follow}(S) = \{a, b, \$\}$

So, the answer to question 52 is option A.

Upvote 7 votes

-- Pooja Palod (31.2k points)

2.16.31 Parsing: GATE2012-53 [top](#)

<http://gateoverflow.in/43312>



Selected Answer

Make your own parse table. Firstly calculate First and Follow from the given grammar.

$\text{First}(S) = \{a, b, \epsilon\}$

$\text{First}(A) = \{a, b, \epsilon\}$

$\text{First}(B) = \{a, b, \epsilon\}$

$\text{Follow}(S) = \{a, b, \$\}$

$\text{Follow}(A) = \{a, b\}$

$\text{Follow}(B) = \{a, b, \$\}$

Now make LL(1) parse table

Non Terminal	a	b	\$
S	$S \rightarrow aAbB$	$S \rightarrow bAbB$	$S \rightarrow \epsilon$
	$S \rightarrow \epsilon$	$S \rightarrow \epsilon$	
A	$A \rightarrow S$	$A \rightarrow S$	
B	$B \rightarrow S$	$B \rightarrow S$	$B \rightarrow S$

Here is the explanation of entries asked in question

1) For E1 and E2 Look into $\text{First}(S) = \{a, b, \epsilon\}$.

a is because of $S \rightarrow aAbB$ and b is because of $B \rightarrow bAbB$

So $M[S, a]$ and $M[S, b]$ will contain $S \rightarrow aAbB$ and $B \rightarrow bAbB$ respectively. For epsilon Look into $\text{Follow}(S) = \{a, b, \$\}$. So $S \rightarrow \epsilon$ will be in $M[S, a]$, $M[S, b]$ and $M[S, \$]$

2) Now for E2 look into $\text{First}(B) = \{a, b, \$\}$. a and b are because of $B \rightarrow S$.

So $M[B, a]$ and $M[B, b]$ will contain $B \rightarrow S$ and for epsilon look into $\text{Follow}(B) = \{a, b, \$\}$. Hence $M[B, \$]$ will contain $B \rightarrow S$

Now we got the answer **E1 is S**

$\rightarrow aAbB, S$

$\rightarrow \epsilon, S$

$\rightarrow bAbB, S$

$\rightarrow \epsilon, S$

$\rightarrow S$.

Hence **Option (C)** is correct.

7 votes

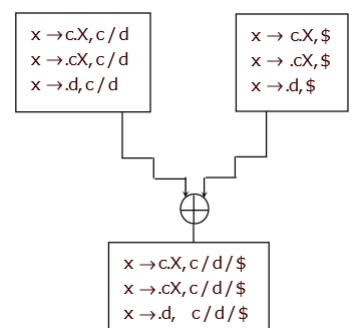
-- Ashwani Kumar (2.5k points)

2.16.32 Parsing: GATE2013_40 [top](#)

<http://gateoverflow.in/1551>



Selected Answer



The TRUE statements are about merging of two states for LALR(1) parser from RR(1) parser.i.e.

1. These can be merged because kernel of these are same, look ahead don't matter in merging
2. Two states are not containing reduces item , so after merging , the merged states can not be contain any S-R conflict.
3. There is no reduction possible so no R-R conflict
4. Merging of states does not depend on further GOTO part on any terminal.

Therefore , ALL given statement in question are FALSE , so option (d) is correct.

15 votes

-- Viral Kapoor (2k points)

2.16.33 Parsing: GATE2013_9 [top](#)



Selected Answer

Ans will be B

A->BC

B->aa

C->bb

now suppose string is aabb

then

A->BC(reduction 3)

->aaC(reduction 2)

->aabb (reduction 1)

n = 4

and number of reductions are 3 so n-1

12 votes

-- rahulkrr (665 points)

2.16.34 Parsing: GATE2014-1-34 [top](#)



Selected Answer

Ans : The given input symbol no where in the given grammar so with given symbol we have neither a shift-reduce nor a reduce-reduce conflict. So, correct answer is (D.) ...

17 votes

-- Jay (1.2k points)

2.16.35 Parsing: GATE2015-3_16 [top](#)



Selected Answer

Answer is C.

SLR is the simplest to implement and Canonical LR is the most powerful.

http://en.wikipedia.org/wiki/LALR_parser_generator

21 votes

-- Arjun Suresh (287k points)

2.16.36 Parsing: GATE2015-3_31 [top](#)

<http://gateoverflow.in/8488>



Selected Answer

A parser works on the basis of given grammar. It takes the grammar as it is. Parser does not work on the basis of the yield of the grammar. Also, while constructing the LL(1) parser table, that entry for terminal 'c' will contain multiple entries. SO LL(1) parser cannot be constructed for the given grammar.

$$S \rightarrow F \mid H$$

$$F \rightarrow p \mid c$$

$$H \rightarrow d \mid c$$

That $\{p, d, c\}$ are the strings generated by the grammar is absolutely correct. But LL(1) and LR(1) can parse these strings successfully only if the grammar is unambiguous and like given below...

$$S \rightarrow P \mid D \mid C$$

$$P \rightarrow p$$

$$D \rightarrow d$$

$$C \rightarrow c$$

Please note the difference between these two grammars. Both derive the same strings, but in different manner. With the grammar given in the question, both top-down and bottom-up parsers will get confused while deriving "c". Top-down parser will get confused between $F \rightarrow c$ and $H \rightarrow c$. Similarly, bottom-up parser will get confused while reducing "c". This confusion in case of bottom-up parsing is technically termed as "reduce-reduce" conflict.

While top-down parsing, first(F) and first(H) are not disjoint, so the grammar cannot be LL(1). Therefore, LL(1) parser cannot parse it.

Hence, the answer should be option (D). Neither S1 nor S2.

Upvote 25 votes

-- ashishacm (303 points)

2.16.37 Parsing: GATE2017-1-17 [top](#)



Selected Answer

Follow of Q is first of R so we get $\{w\}$

but since R can be Null so we have to check first of S which is $\{y\}$

so FOLLOW Q = $\{w, y\}$

Correct option (C)

Upvote 4 votes

-- sriv_shubham (2.4k points)

2.16.38 Parsing: GATE2017-1-43 [top](#)



Selected Answer

This question is picked from area of **Counting in Combinatorics**.

Given:

if e1 then e2 else e3 has 2 control flow paths e1->e2 and e1->e3.
(Meaning of "how many control flow" for if structure is clearly mentioned)

What is asked:

Number of control flow paths for 10 if terminals?

Solution:

To get 10 if's we need to use grammar to get,

```
if <expr> then <expr> else <expr> ; stmt
if <expr> then <expr> else <expr> ; if <expr> then <expr> else <expr> ; stmt
.....
```

.....
.....

(keep doing it 10 times to get 10 if's)

Observe that there is a **semi-colon after every if structure.**

We know that every if structure has 2 control flows as given in question. Hence,

We have 2 control flow choices for 1st if terminal.

We have 2 control flow choices for 2nd if terminal.

.....
.....
.....

We have 2 control flow choices for 10th if terminal.

By using multiplicative law of counting we get,

Total choices as $2^2 \cdot 2^2 \cdot 2^2 \cdots \cdot 2^2 \cdots \cdot 10 \text{ times} = 2^{10} = 1024$

Once again, **one need not know "what control flow" is**, but needs to know "**how many control flows**" are in if structure which is given in question.

5 votes

-- prs11 (165 points)

each if-else condition leads to two different paths, there are 10 if-else conditions one after the other, therefore totally $2^{10} = 1024$ paths possible

10 votes

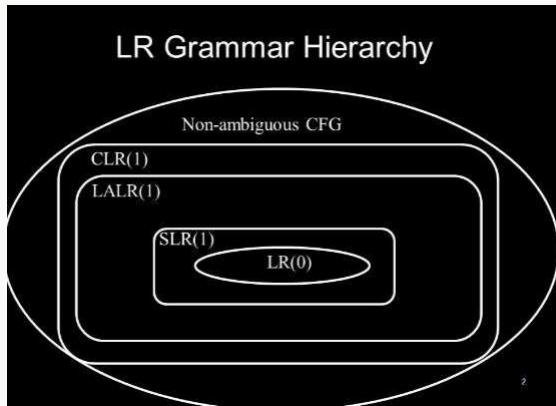
-- Vinay Rachapalli (1k points)

2.16.39 Parsing: GATE2017-2-6 [top](#)

<http://gateoverflow.in/118343>



Selected Answer



so only 1st one is correct ans

6 votes

-- 2018 (4.9k points)

2.16.40 Parsing: TIFR2012-B-17 [top](#)

<http://gateoverflow.in/25215>



Selected Answer

- A) Does not make any sense. false.
- B) This is definition of LR(K) Parser. True
- C) False. LR(K) is subset of CFL.
- D) False.
- E) LR(K) , bottom up parser . We have Right most derivation. This is False.

Answer :- B

7 votes

-- Akash (41.5k points)

2.16.41 Parsing: TIFR2012-B-8 [top](#)

<http://gateoverflow.in/25108>



Selected Answer

e is correct

Because as the expression evaluated right to left , so in (ii) $2+(a-(b*(a+b)))$ this evaluation performed, which is not a correct evaluation as the parse tree

4 votes

-- srestha (55.2k points)

2.16.42 Parsing: TIFR2015-B-15 [top](#)

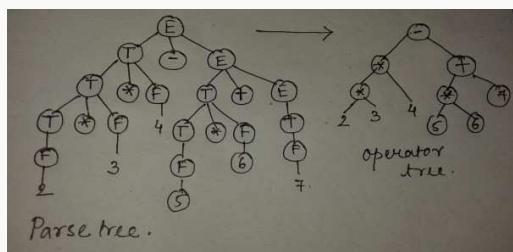
<http://gateoverflow.in/30079>



Selected Answer

Answer is option B.

The corresponding parse tree is drawn for the given expression according to the given grammar .



9 votes

-- Riya Roy(Arayana) (7k points)

2.17

Programming In C(1) [top](#)

2.17.1 Programming In C: GATE2005-61 [top](#)

<http://gateoverflow.in/4068>

Consider line number 3 of the following C-program.

```
int main() { /*Line 1 */
    int I, N; /*Line 2 */
    fro (I=0, I<N, I++); /*Line 3 */
}
```

Identify the compiler's response about this line while creating the object-module:

- A. No compilation error
- B. Only a lexical error
- C. Only syntactic errors
- D. Both lexical and syntactic errors

[gate2005](#) [compiler-design](#) [programming-in-c](#) [normal](#)

Answer

Answers: Programming In C

2.17.1 Programming In C: GATE2005-61 [top](#)

<http://gateoverflow.in/4068>



C language allows only certain words in it- these are called tokens. If we input any invalid tokens it causes lexical error.

eg:

44a44

causes lexical error as in C as an alphabet cannot come in between digits.

Syntactic error is caused by bad combination of tokens. For example, we cannot have a constant on the left hand side of an assignment statement, a for loop must have two expressions inside () separated by semi colon etc.

In the given question, line 3 won't cause lexical error or syntactic error. The statement will be treated as a function call with three arguments. Function definition being absent will cause link time error, but the question asks only for compile time errors. So, (a) must be the answer.

PS: Implicit function declaration was removed from C99 standard onwards. As per current standard we should not use a function without declaration. Still, we cannot guarantee "compilation error"- just expect compiler warnings in C. In C++ this should produce compilation (semantic) error.

<http://stackoverflow.com/questions/15570553/lexical-and-semantic-errors-in-c>

28 votes

-- Arjun Suresh (287k points)

2.18

Recursion(1) top

2.18.1 Recursion: GATE2014-3-18 top

<http://gateoverflow.in/2052>

Which of the following statements are CORRECT?

1. Static allocation of all data areas by a compiler makes it impossible to implement recursion.
 2. Automatic garbage collection is essential to implement recursion.
 3. Dynamic allocation of activation records is essential to implement recursion.
 4. Both heap and stack are essential to implement recursion.
- A. 1 and 2 only
 B. 2 and 3 only
 C. 3 and 4 only
 D. 1 and 3 only

[gate2014-3](#) [compiler-design](#) [recursion](#) [normal](#)

Answer

Answers: Recursion

2.18.1 Recursion: GATE2014-3-18 top

<http://gateoverflow.in/2052>



It will be D.

option 2 is wrong because it is not necessary to have automatic garbage collection to implement recursion.

option 4 is wrong because it says that both are required to implement recursion, which is wrong. Either of them will suffice.

13 votes

-- Gate Keeda (18.8k points)

2.19

Register Allocation(4) top

2.19.1 Register Allocation: GATE1997_4.9 top

<http://gateoverflow.in/2250>

The expression $(a * b) * c$ op...

where 'op' is one of '+', '*' and ' \uparrow ' (exponentiation) can be evaluated on a CPU with single register without storing the value of $(a * b)$ if

- A. 'op' is '+' or '*'
- B. 'op' is ' \uparrow ' or '*'
- C. 'op' is ' \uparrow ' or '+'
- D. not possible to evaluate without storing

[gate1997](#) [compiler-design](#) [target-code-generation](#) [register-allocation](#) [normal](#)

[Answer](#)

2.19.2 Register Allocation: GATE2010-37 [top](#)

<http://gateoverflow.in/2338>

The program below uses six temporary variables a, b, c, d, e, f .

```
a = 1
b = 10
c = 20
d = a + b
e = c + d
f = c + e
b = c + e
e = b + f
d = 5 + e
return d + f
```

Assuming that all operations take their operands from registers, what is the minimum number of registers needed to execute this program without spilling?

- A. 2
- B. 3
- C. 4
- D. 6

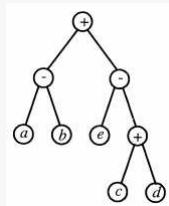
[gate2010](#) [compiler-design](#) [target-code-generation](#) [register-allocation](#) [normal](#)

[Answer](#)

2.19.3 Register Allocation: GATE2011_36 [top](#)

<http://gateoverflow.in/2138>

Consider evaluating the following expression tree on a machine with load-store architecture in which memory can be accessed only through load and store instructions. The variables a, b, c, d , and e are initially stored in memory. The binary operators used in this expression tree can be evaluated by the machine only when operands are in registers. The instructions produce result only in a register. If no intermediate results can be stored in memory, what is the minimum number of registers needed to evaluate this expression?



- (A) 2
- (B) 9
- (C) 5
- (D) 3

[gate2011](#) [compiler-design](#) [register-allocation](#) [normal](#)

[Answer](#)

2.19.4 Register Allocation: GATE2017-1-52 [top](#)

<http://gateoverflow.in/118746>

Consider the expression $(a - 1) * (((b + c)/3) + d)$. Let X be the minimum number of registers required by an *optimal* code generation (without any register spill) algorithm for a load/store architecture, in which (i) *only load and store instructions can have memory operands and (ii) arithmetic instructions can have only register or immediate operands*. The value of X is _____.

gate2017-1 compiler-design register-allocation normal numerical-answers

[Answer](#)

Answers: Register Allocation

2.19.1 Register Allocation: GATE1997_4.9 [top](#)

<http://gateoverflow.in/2250>



A)

\uparrow has higher precedence than $\{*, +, -, /\}$

So, if $op = \uparrow$ implies, we need to evaluate the right hand side of \uparrow first and then do the lhs part, which would definitely require us to store the value of lhs

but if its a '+' or '*', we don't need to store the values evaluated, and on the go can do the operation directly on one register.

4 votes

-- confused_luck (801 points)

2.19.2 Register Allocation: GATE2010-37 [top](#)

<http://gateoverflow.in/2338>



Here in these types of compiler questions, idea is "map/assign multiple temporaries to one registers."

here a, b, and c all are having 3 different values so i need atleast 3 registers r1, r2 and r3.
a is mapped to r1, b to r2 and c to r3.

$d = a + b$, after this line if u notice 'a' is never present on right hand side, so i can map (register of a which is r1) d to r1.
 $e = c + d$, after this line 'd' is never present on rhs, so i can map (register of d which is r1) e to r1.

at this time mapping is

r1 --- e
r2 --- b
r3 --- c

(at this moment i have registers for e, b and c. if i introduce new variable then i may need different register)

now at this point if u see

$f = c + e$
 $b = c + e$

these two are essentially doing same thing, after these two line 'b' and 'f' are same so i can skip computing 'f'. and wherever f is present i will replace it with 'b'. (bcoz neither of 'f' and 'b' are changing after these two lines, so value of these will be 'c+e' forever)

(seems like i introduced one more variable f, and register is needed for that, but actually i did not really introduce 'f'. i am skipping computation of 'f')

now at second last line " $d = 5 + e$ "

here i introduced 'd', i can map it to any of the register r1 or r3, bcoz after this line neither of 'e' or 'c' is required. (value of 'b' is required bcoz i need to return ' $d+f$ ', and 'f' is essentially equal to 'b')

finally code becomes

```
r1 = 1
r2 = 10
r3 = 20
r1 = r1 + r2
r1 = r3 + r1
(skipping 'f' computation)
```

```
r2 = r3 + r1
r2 = r3 + r1
r1 = r2 + r2
r3 = 5 + r1
return r3 + r2
```

Therefore minimum 3 registers needed.

9 votes

-- Sachin Mittal (6k points)

After making the interference graph it can be colored with 3 different colors. Therefore minimum number of color needed to execute the program without spilling would be 3 (B).

10 votes

-- suraj (5k points)

2.19.3 Register Allocation: GATE2011_36 [top](#)

<http://gateoverflow.in/2138>



Selected Answer

Given is Load Store Architecture, that means we can access memory using Load and Store Instructions.

Key Idea:- Pick new register only when it is required.

We want to add **c** and **d**, and initially both are in memory, therefore copy these into registers.

- load R1, c ($R1 \leftarrow c$)
- load R2, d ($R2 \leftarrow d$)

(here no compensation can be done, we need two registers)

- add R1, R1, R2 ($R1 \leftarrow R1 + R2$)

(at this point R1 is holding **c+d** and R2 is holding **d**, i.e. $R1 \leftarrow c + d$ and $R2 \leftarrow d$)

Now, **e** comes into picture and my question is, Can i make use of R1 or R2 to store **e**?
I can not use R1 to store **e** as its value will be needed later but I can use R2.

- load R2, e

(currently $R1 \leftarrow c + d$ and $R2 \leftarrow e$)

- Sub R1, R2, R1 ($R1 \leftarrow R2 - R1$)

Doing this all gives, final value of right sub-tree is stored in R1, and R2 stores **e**.

Now, coming to left subtree, to perform "a-b" we need to copy both variables in registers.

We can copy one of the variable in R2, but we can not obviously copy in R1 as value of R1 will be required later.

- Load R2, a
- Load R3, b (**here comes extra register, and we can not avoid using it.**)

Current mapping is $R2 \leftarrow a$, $R3 \leftarrow b$ and R1 contains final value of Right subtree.

- SUB R2, R2, R3 ($R2 \leftarrow R2 - R3$)
- ADD R1, R1 , R2

Hence answer is 3 i.e. **D**

1 votes

-- Sachin Mittal (6k points)

$R1 \leftarrow -c$, $R2 \leftarrow -d$, $R2 \leftarrow -R1 + R2$, $R1 \leftarrow -e$, $R2 \leftarrow -R1 - R2$

To calculate the rest of the expression we must load a and b into the registers but we need the content of R2 later. So we must use another register. $R1 \leftarrow -a$, $R3 \leftarrow -b$, $R1 \leftarrow -R1 - R3$, $R1 \leftarrow -R1 + R2$

Ans D

15 votes

-- Keith Kr (6.2k points)

2.19.4 Register Allocation: GATE2017-1-52 [top](#)

<http://gateoverflow.in/116746>



Selected Answer

Load R1,b
 Load R2,c
 ADD R1,R2
 Div R1,3
 Load R2,d
 Add R1,R2
 Load R2,a
 Sub R2,1
 Mul R2,R1
 hence minimum 2 registers required

15 votes

-- sriv_shubham (2.4k points)

2.20**Runtime Environments(9) [top](#)****2.20.1 Runtime Environments: CMI2011-A-08 [top](#)**<http://gateoverflow.in/46195>

In programming languages like C, C++, Python . . . the memory used by a program is typically separated into two parts, the stack and the heap. Consider the following statements:

1. A stack is efficient for managing nested function calls.
2. Stack space is limited while heap space is not.
3. The stack cannot be used for persistent data structures.

Then:

- A. 1 and 2 are true but 3 is false.
- B. 1 and 3 are true but 2 is false.
- C. 2 and 3 are true but 1 is false.
- D. All three statements are true.

[cmi2011](#) [compiler-design](#) [runtime-environments](#)**Answer****2.20.2 Runtime Environments: GATE1993_7.7 [top](#)**<http://gateoverflow.in/2295>

A part of the system software which under all circumstances must reside in the main memory is:

- a. text editor
- b. assembler
- c. linker
- d. loader
- e. none of the above

[gate1993](#) [compiler-design](#) [runtime-environments](#) [easy](#)**Answer****2.20.3 Runtime Environments: GATE1995_1.14 [top](#)**<http://gateoverflow.in/2601>

A linker is given object modules for a set of programs that were compiled separately. What information need to be included

in an object module?

- A. Object code
- B. Relocation bits
- C. Names and locations of all external symbols defined in the object module
- D. Absolute addresses of internal symbols

gate1995 compiler-design runtime-environments normal

Answer

2.20.4 Runtime Environments: GATE1997_1.8 [top](#)

<http://gateoverflow.in/2224>

A language L allows declaration of arrays whose sizes are not known during compilation. It is required to make efficient use of memory. Which one of the following is true?

- a. A compiler using static memory allocation can be written for L
- b. A compiler cannot be written for L ; an interpreter must be used
- c. A compiler using dynamic memory allocation can be written for L
- d. None of the above

gate1997 compiler-design easy runtime-environments

Answer

2.20.5 Runtime Environments: GATE1998-1.25, ISRO2008-41 [top](#)

<http://gateoverflow.in/1662>

In a resident – OS computer, which of the following systems must reside in the main memory under all situations?

- A. Assembler
- B. Linker
- C. Loader
- D. Compiler

gate1998 compiler-design runtime-environments normal isro2008

Answer

2.20.6 Runtime Environments: GATE1998-2.15 [top](#)

<http://gateoverflow.in/1687>

Faster access to non-local variables is achieved using an array of pointers to activation records called a

- A. stack
- B. heap
- C. display
- D. activation tree

gate1998 programming compiler-design normal runtime-environments

Answer

2.20.7 Runtime Environments: GATE1998_1.28 [top](#)

<http://gateoverflow.in/1665>

A linker reads four modules whose lengths are 200, 800, 600 and 500 words, respectively. If they are loaded in that order, what are the relocation constants?

- A. 0, 200, 500, 600
- B. 0, 200, 1000, 1600
- C. 200, 500, 600, 800
- D. 200, 700, 1300, 2100

gate1998 compiler-design runtime-environments normal

Answer

2.20.8 Runtime Environments: GATE2001-1.17 [top](#)<http://gateoverflow.in/710>

The process of assigning load addresses to the various parts of the program and adjusting the code and the data in the program to reflect the assigned addresses is called

- A. Assembly
- B. parsing
- C. Relocation
- D. Symbol resolution

gate2001 compiler-design runtime-environments easy

Answer

2.20.9 Runtime Environments: GATE2014-2-18 [top](#)<http://gateoverflow.in/1975>

Which one of the following is **NOT** performed during compilation?

- A. Dynamic memory allocation
- B. Type checking
- C. Symbol table management
- D. Inline expansion

gate2014-2 compiler-design easy runtime-environments

Answer

Answers: Runtime Environments**2.20.1 Runtime Environments: CMI2011-A-08** [top](#)<http://gateoverflow.in/46195>

Option B) 1 and 3 are true but 2 is false.

Because **Size of heap and stack both are limited**. There is nothing which is **unlimited**. Yes but **Size of stack and Size of heap is constant**. It means that If there is **total size available is 10**, and If i used **7 block for stack** then i can only use **3 block for heap** but not more than that. i.e **Size of stack and size of heap are inversely proportional**. I mean that if size of stack increases then size of heap decreases, and vice versa.

4 votes

-- Muktinath Vishwakarma (33.9k points)

2.20.2 Runtime Environments: GATE1993_7.7 [top](#)<http://gateoverflow.in/2295>

Selected Answer

Answer: D

The loader is a program that loads the object program from the secondary memory into the main memory for execution of the program. The loader resides in main memory.

11 votes

-- Rajarshi Sarkar (34.4k points)

2.20.3 Runtime Environments: GATE1995_1.14 [top](#)<http://gateoverflow.in/2601>

Selected Answer

(c) is the answer. For linker to link external symbols (for example in C, to link an extern variable in one module to a global variable in another module), it must know the location of all external symbols. In C external symbols includes all global variables and function names.

(a) is trivially there is an object module. (b) must be there if we need to have relocation capability.

(d) is no way needed.

1 20 votes

-- Arjun Suresh (287k points)

2.20.4 Runtime Environments: GATE1997_1.8 [top](#)

<http://gateoverflow.in/2224>



Selected Answer

C.

Using dynamic memory allocation, memory will be allocated to array at runtime.

1 11 votes

-- Gate Keeda (18.8k points)

2.20.5 Runtime Environments: GATE1998-1.25, ISRO2008-41 [top](#)

<http://gateoverflow.in/1662>



Selected Answer

C) is answer ,

In many operating systems the loader is permanently resident in memory, although some operating systems that support [virtual memory](#) may allow the loader to be located in a region of memory that is [pageable](#).

reference @[http://en.wikipedia.org/wiki/Loader_\(computing\)](http://en.wikipedia.org/wiki/Loader_(computing))

1 8 votes

-- Mithlesh Upadhyay (5.3k points)

2.20.6 Runtime Environments: GATE1998-2.15 [top](#)

<http://gateoverflow.in/1687>



Selected Answer

it is C

properties of displays

- 1> Use a pointer array to store the activation records along the static chain.
- 2> Fast access for non-local but may be complicated to maintain.
- 3> Calling a subprogram in the same level – simply replace and restore.
- 4> Calling a subprogram in the higher level – add an entry and may need to save the old pointers.
- 5> Calling a subprogram in the lower level – shrink the pointer and restore it when the subprogram returns.

<http://users.dickinson.edu/~wahlst/356/ch10.pdf>

1 10 votes

-- sumit kumar singh dixit (2.3k points)

2.20.7 Runtime Environments: GATE1998_1.28 [top](#)

<http://gateoverflow.in/1665>



Selected Answer

answer - B

first module loaded starting at address 0. Size is 200. hence it will occupy first 200 address last address being 199. Second module will be present from 200 and so on.

1 10 votes

-- ankitrokdeonsns (9.1k points)

2.20.8 Runtime Environments: GATE2001-1.17 [top](#)

<http://gateoverflow.in/710>



Selected Answer

Relocation is the process of assigning load addresses to position-dependent code of a program and adjusting the code and data in the program to reflect the assigned addresses.

Hence Option C is Ans

Symbol resolution is the process of searching files and libraries to replace symbolic references or names of libraries with actual usable addresses in memory before running a program.

6 votes

-- Rajesh Pradhan (16.9k points)

2.20.9 Runtime Environments: GATE2014-2-18 [top](#)

<http://gateoverflow.in/1975>



Selected Answer

Dynamic means- at runtime. Dynamic memory allocation happens during the execution time and hence (A) is the answer.

18 votes

-- Arjun Suresh (287k points)

2.21 Static Single Assignment(2)

Static Single Assignment(2)

<http://gateoverflow.in/39675>

Consider the following code segment.

```
x = u - t;
y = x * v;
x = y + w;
y = t - z;
y = x * y;
```

The minimum number of *total* variables required to convert the above code segment to *static single assignment* form is _____.

[gate2016-1](#) [compiler-design](#) [static-single-assignment](#) [normal](#) [numerical-answers](#)

Answer

2.21.2 Static Single Assignment: GATE2017-1-12 [top](#)

<http://gateoverflow.in/118292>

Consider the following intermediate program in three address code

```
p = a - b
q = p * c
p = u * v
q = p + q
```

Which one of the following corresponds to a *static single assignment* form of the above code?

(A)

```
p1 = a - b
q1 = p1 * c
p1 = u * v
q1 = p1 + q1
```

(B)

```
p3 = a - b
q4 = p3 * c
p4 = u * v
q5 = p4 + q4
```

(C)

```
p1 = a - b
q1 = p2 * c
p3 = u * v
q2 = p4 + q3
```

(D)

```
p1 = a - b
q1 = p * c
p2 = u * v
q2 = p + q
```

[gate2017-1](#) [compiler-design](#) [intermediate-code](#) [normal](#) [static-single-assignment](#)
[Answer](#)

Answers: Static Single Assignment

2.21.1 Static Single Assignment: GATE 2016-1-19 [top](#)

<http://gateoverflow.in/39675>


Selected Answer

```
x(temp3) = u(temp1) - t(temp2);
y(temp5) = x * v(temp4);
x(temp7) = y + w(temp6);
y(temp9) = t - z(temp8);
y(temp10) = x * y;
```

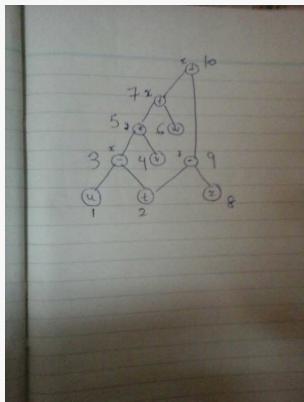
so ans should be.10

[16 votes](#)

-- papesh (22.8k points)

Static single assignment means assignment to register can be done one time only.

so, draw the GRAPH and count number of nodes which will give the number of register required.


[19 votes](#)

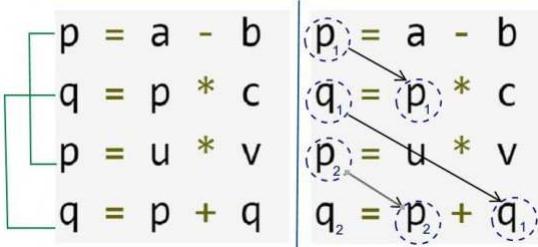
-- Nishank Garg (107 points)

2.21.2 Static Single Assignment: GATE2017-1-12 [top](#)

<http://gateoverflow.in/118292>


Selected Answer

static single assignment
 • Each assignment to a temporary is given a unique name
 • All of the uses reached by that assignment are renamed



https://en.wikipedia.org/wiki/Static_single_assignment_form

<http://www.cse.iitd.ernet.in/~nvkrishna/courses/winter07/ssa.pdf>

<https://www.cs.cmu.edu/~fp/courses/15411-f08/lectures/09-ssa.pdf>

<http://www.seas.harvard.edu/courses/cs252/2011sp/slides/Lec04-SSA.pdf>

so B is ans.

3 votes

-- 2018 (4.9k points)

2.22

Syntax Directed Translation(6) top

2.22.1 Syntax Directed Translation: GATE 2016-1-46 top

<http://gateoverflow.in/39700>

Consider the following Syntax Directed Translation Scheme (*SDTS*), with non-terminals $\{S, A\}$ and terminals $\{a, b\}$.

$$S \rightarrow aA \quad \{\text{print } 1\}$$

$$S \rightarrow a \quad \{\text{print } 2\}$$

$$A \rightarrow Sb \quad \{\text{print } 3\}$$

Using the above *SDTS*, the output printed by a bottom-up parser, for the input *aab* is:

- A. 132
- B. 223
- C. 231
- D. syntax error

[gate2016-1](#) [compiler-design](#) [syntax-directed-translation](#) [normal](#)

[Answer](#)

<http://gateoverflow.in/2622>

2.22.2 Syntax Directed Translation: GATE1995_2.10 top

A shift reduce parser carries out the actions specified within braces immediately after reducing with the corresponding rule of grammar

- $S \rightarrow xxW\{\text{print}'1'\}$
- $S \rightarrow y\{\text{print}'2'\}$
- $W \rightarrow Sz\{\text{print}'3'\}$

What is the translation of $xxxxyz$ using the syntax directed translation scheme described by the above rules?

- A. 23131
- B. 11233
- C. 11231
- D. 33211

[gate1995](#) [compiler-design](#) [grammar](#) [syntax-directed-translation](#) [normal](#)

[Answer](#)

2.22.3 Syntax Directed Translation: GATE1996_20 [top](#)

<http://gateoverflow.in/2772>

Consider the syntax-directed translation schema (SDTS) shown below:

- $E \rightarrow E + E \{ \text{print } "+" \}$
- $E \rightarrow E * E \{ \text{print } "\cdot" \}$
- $E \rightarrow id \{ \text{print id.name} \}$
- $E \rightarrow (E)$

An LR-parser executes the actions associated with the productions immediately after a reduction by the corresponding production. Draw the parse tree and write the translation for the sentence.

$(a+b)*(c+d)$, using SDTS given above.

[gate1996](#) [compiler-design](#) [syntax-directed-translation](#) [normal](#)

[Answer](#)

2.22.4 Syntax Directed Translation: GATE1998_23 [top](#)

<http://gateoverflow.in/1738>

Let the attribute 'val' give the value of a binary number generated by S in the following grammar:

$S \rightarrow L \cdot L \mid L$

$L \rightarrow LB \mid B$

$B \rightarrow 0 \mid 1$

For example, an input 101.101 gives $S.val = 5.625$

Construct a syntax directed translation scheme using only synthesized attributes, to determine $S.val$.

[gate1998](#) [compiler-design](#) [syntax-directed-translation](#) [normal](#)

[Answer](#)

2.22.5 Syntax Directed Translation: GATE2000-19 [top](#)

<http://gateoverflow.in/690>

Consider the syntax directed translation scheme (SDTS) given in the following. Assume attribute evaluation with bottom-up parsing, i.e., attributes are evaluated immediately after a reduction.

$E \rightarrow E_1 * T \{ E.val = E_1.val * T.val \}$

$E \rightarrow T \{ E.val = T.val \}$

$T \rightarrow F - T_1 \{ T.val = F.val - T_1.val \}$

$T \rightarrow F \{ T.val = F.val \}$

$F \rightarrow 2 \{ F.val = 2 \}$

$F \rightarrow 4 \{ F.val = 4 \}$

- Using this SDTS, construct a parse tree for the expression $4 - 2 - 4 * 2$ and also compute its $E.val$.
- It is required to compute the total number of reductions performed to parse a given input. Using synthesized attributes only, modify the SDTS given, without changing the grammar, to find $E.red$, the number of reductions performed while reducing an input to E .

gate2000 compiler-design syntax-directed-translation normal descriptive

[Answer](#)

2.22.6 Syntax Directed Translation: GATE2003-18 [top](#)

<http://gateoverflow.in/508>

In a bottom-up evaluation of a syntax directed definition, inherited attributes can

- A. always be evaluated
- B. be evaluated only if the definition is L-attributed
- C. be evaluated only if the definition has synthesized attributes
- D. never be evaluated

gate2003 compiler-design syntax-directed-translation normal

[Answer](#)

Answers: Syntax Directed Translation

2.22.1 Syntax Directed Translation: GATE 2016-1-46 [top](#)

<http://gateoverflow.in/39700>



Selected Answer

aab could be derived as follows by the bottom up parser:

S->aA prints 1
A->a**S**b prints 3
A->a**a**b prints 2

Now since bottom up parser will work in reverse of right most derivation, so it will print in bottom up fashion i.e., 231 which is option C.

Note that this could also be visualized easily by drawing the derivation tree.

25 votes

-- Monanshi Jain (8.1k points)

2.22.2 Syntax Directed Translation: GATE1995_2.10 [top](#)

<http://gateoverflow.in/262>



Selected Answer

A.

Making a tree and performing post order traversal will yield answer as A.

S-->x x W (Pf'1')
W-->S z (Pf'3')
S-->x x W (Pf'1')
W-->S z (Pf'3')
S-->y (Pf'2').

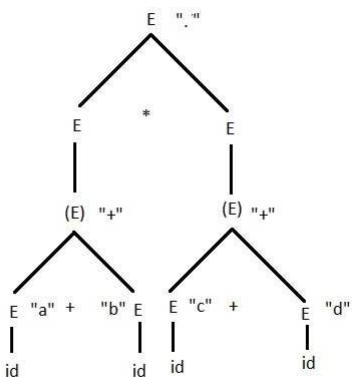
13 votes

-- Gate Keeda (18.8k points)

2.22.3 Syntax Directed Translation: GATE1996_20 [top](#)

<http://gateoverflow.in/2772>

ab+cd++.



9 votes

-- jayendra (7.7k points)

2.22.4 Syntax Directed Translation: GATE1998_23 [top](#)

<http://gateoverflow.in/1738>



Selected Answer

$S-- > L.L \{ S.dv = L_1.dv + L_2.dv/2^{L_2.nb} \}$

| $L \{ S.dv = L.dv \}$

$L-- > LB \{ L.dv = 2 * L_1.dv + B.dv$

$L.nb = L_1.nb + B.nb \}$

| $B \{ L.dv = B.dv$

$L.nb = B.nb \}$

$B-- > 0 \{ B.dv = 0$

$B.nb = 1 \}$

| $1 \{ B.dv = 1$

$B.nb = 1 \}$

here dv = decimal value

nb = number of bits.

8 votes

-- Gate Keeda (18.8k points)

2.22.5 Syntax Directed Translation: GATE2000-19 [top](#)

<http://gateoverflow.in/690>



Selected Answer

A. Given Expression is $4 - 2 - 4 * 2$, which can be rewritten as $((4 - (2 - 4)) * 2)$ which is equal to 12.

9 votes

-- Aditya Gaurav (2.8k points)

2.22.6 Syntax Directed Translation: GATE2003-18 [top](#)

<http://gateoverflow.in/908>

Ans:C

lets see how ,consider following SDD:

A->BC B.i =A.i

B->b B.s=b.val

C->c C.s=c.val

if u draw parse tree

A

B C

b c

now ur traversing bottom up

first you 'll evaluate B.s then you want to evaluate B.i but as we can see as i is an inherited attribute and it is dependent on A.i but we havent evaluated A.i yet coz we are doing a bottom up traversal and we havent seen A yet . so it cannot be evaluated due to Inherited attribute i.

now if A=BC A.s=B.s+C.s

B->b B.s=b.val

C->c C.s=c.val

see that

A

B C

b c

you 'll traverse bottom up ,first eval B.s=b then C.s=c

then when you are at A ,you have already evaluated all the attributes on which A.s is dependent..so Only in this case where SDD has Synthesized attributes ,then only Bottom up Evaluation can be used.

for Inherited attributes we will need a depth first evaluation.

3 votes

-- Anurag Semwal (7.6k points)

2.23

Target Code Generation(2) top

2.23.1 Target Code Generation: GATE2003-59 top

<http://gateoverflow.in/947>

Consider the syntax directed definition shown below.

```
S → id := E {gen(id.place = E.place;)}

E → E1 + E2 {t = newtemp();
    gen(t = E1.place + E2.place);
    E.place = t;}

E → id {E.place = id.place;}
```

Here, *gen* is a function that generates the output code, and *newtemp* is a function that returns the name of a new temporary variable on every call. Assume that *t_i*'s are the temporary variable names generated by *newtemp*. For the statement 'X := Y + Z', the 3-address code sequence generated by this definition is

- A. X = Y + Z
- B. t₁ = Y + Z; X = t₁
- C. t₁ = Y; t₂ = t₁ + Z; X = t₂
- D. t₁ = Y; t₂ = Z; t₃ = t₁ + t₂; X = t₃

[gate2003](#) [compiler-design](#) [target-code-generation](#) [normal](#)

Answer

2.23.2 Target Code Generation: GATE2004-10 top

<http://gateoverflow.in/4069>

Consider the grammar rule E → E₁ – E₂ for arithmetic expressions. The code generated is targeted to a CPU having a single user register. The subtraction operation requires the first operand to be in the register. If E₁ and E₂ do not have any common sub expression, in order to get the shortest possible code

- A. E1 should be evaluated first
 B. E2 should be evaluated first
 C. Evaluation of E1 and E2 should necessarily be interleaved
 D. Order of evaluation of E1 and E2 is of no consequence

gate2004 compiler-design target-code-generation normal

[Answer](#)

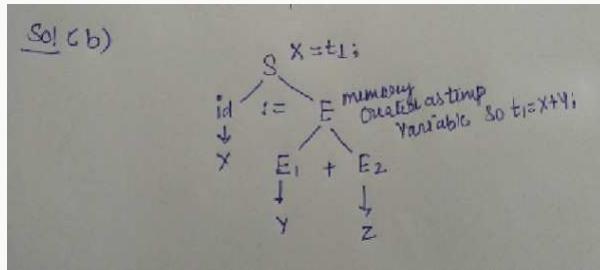
Answers: Target Code Generation

2.23.1 Target Code Generation: GATE2003-59 [top](#)

<http://gateoverflow.in/947>



Selected Answer



using given semantic rules.

[Upvote](#) 7 votes

-- Prashant Singh (47.6k points)

2.23.2 Target Code Generation: GATE2004-10 [top](#)

<http://gateoverflow.in/4069>



Selected Answer

E2 should be evaluated first

After evaluating E2 first and then E1, we will have E1 in the register and thus we can simply do SUB operation with E2 which will be in memory (as we have only a single register). If we do E1 first and then E2, we must move E2 to memory and E1 back to register before doing SUB, which will increase the code size.

[Upvote](#) 21 votes

-- Arjun Suresh (287k points)

2.24

Variable Scope(2) [top](#)

2.24.1 Variable Scope: GATE1987-1-xix [top](#)

<http://gateoverflow.in/80373>

Study the following program written in a block-structured language:

```
Var x, y:integer;
procedure P(n:integer);
begin
  x:=(n+2) / (n-3);
end;

procedure Q
Var x, y:integer;
begin
  x:=3;
  y:=4;
  P(y);
  Write(x)
end;

begin
  x:=7;
  y:=8;
  Q;
end;
```

```
Write(x); _____ (2)  
end.
```

What will be printed by the write statements marked (1) and (2) in the program if the variables are statically scoped?

- A. 3,6
- B. 6,7
- C. 3,7
- D. None of the above.

gate1987 compiler-design variable-scope

Answer

2.24.2 Variable Scope: GATE1987-1-xx [top](#)

<http://gateoverflow.in/80374>

For the program given below what will be printed by the write statements marked (1) and (2) in the program if the variables are dynamically scoped?

```
Var x, y:integer;  
procedure P(n:integer);  
begin  
    x := (n+2) / (n-3);  
end;  
  
procedure Q  
Var x, y:integer;  
begin  
    x:=3;  
    y:=4;  
    P(y);  
    Write(x); _____ (1)  
end;  
  
begin  
    x:=7;  
    y:=8;  
    Q;  
    Write(x); _____ (2)  
end.
```

- A. 3,6
- B. 6,7
- C. 3,7
- D. None of the above

gate1987 compiler-design variable-scope

Answer

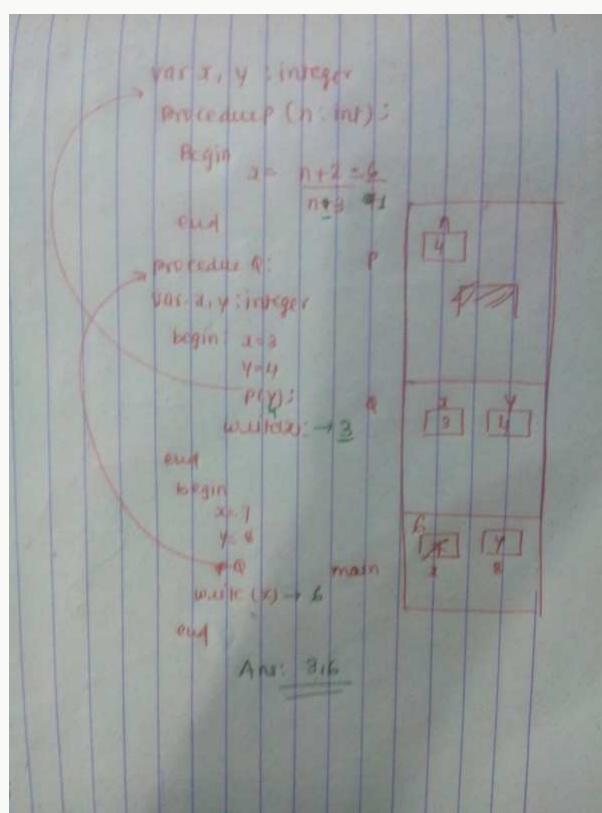
Answers: Variable Scope

2.24.1 Variable Scope: GATE1987-1-xix [top](#)

<http://gateoverflow.in/80373>



Selected Answer



in static scoping free varible is replace by global variable, so option a is correct.

2 votes

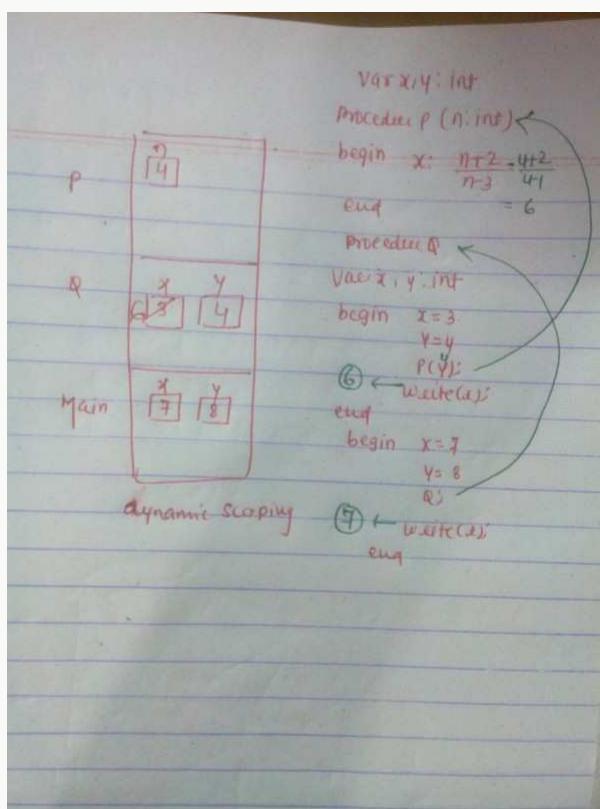
-- Hira Thakur (5.9k points)

2.24.2 Variable Scope: GATE1987-1-xx [top](#)

<http://gateoverflow.in/80374>



Selected Answer



ans is b

1 votes

-- Hira Thakur (5.9k points)

2.25

Viable Prefix(1) top

2.25.1 Viable Prefix: GATE2015-1_13 top

<http://gateoverflow.in/8187>

Which one of the following is TRUE at any valid state in shift-reduce parsing?

- A. Viable prefixes appear only at the bottom of the stack and not inside
- B. Viable prefixes appear only at the top of the stack and not inside
- C. The stack contains only a set of viable prefixes
- D. The stack never contains viable prefixes

[gate2015-1](#) | [compiler-design](#) | [parsing](#) | [normal](#) | [viable-prefix](#)

[Answer](#)

Answers: Viable Prefix

2.25.1 Viable Prefix: GATE2015-1_13 top

<http://gateoverflow.in/8187>



Selected Answer

Answer - C

Explanation -

A handle is actually the one which is always on the top of the stack. A viable prefix(prefix of the Right-hand side of a production or productions), is actually a prefix of the handle and so can never extend past the right end of the handle(i.e. the top of the stack).

The structure of the stack can be considered as a set of viable prefixes -

Stack = {Prefix₁Prefix₂Prefix₃,.....Prefix_{n-1}Prefix_n} and so it is not wrong to say that the stack contains a set of viable prefixes.

Source :- <https://www.youtube.com/watch?v=UeRyF72ObXo&list=PLFB9EC7B8FE963EB8&index=35> @7:50

Upvote 14 votes

-- Ravi Ranjan (2.6k points)

C) should be the answer

Upvote 10 votes

-- GateMaster Prime (1.6k points)

3 Programming & DS: DS (198) top

3.1 Abstract Data Type(1) top

3.1.1 Abstract Data Type: GATE2005-2 top

<http://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](#)

[Answer](#)

Answers: Abstract Data Type

3.1.1 Abstract Data Type: GATE2005-2 top

<http://gateoverflow.in/1344>



Selected Answer

An abstract data type (ADT) supports only the operations which are defined.

Abstract class is one that may not have definitions of all the objects it have. Moreover it can not be instantiated. To instantiate we have to create a subclass then instantiate the class.

Abstract Data Type is like data structure eg. STACK where we have PUSH() POP() operation defined .

Hence they are not the same thing.

<http://www.devx.com/tips/Tip/5681>

15 votes

-- Manali (2.8k points)

3.2 Arrays(14) top

3.2.1 Arrays: GATE1993-12 top

<http://gateoverflow.in/2309>

The following Pascal program segments finds the largest number in a two-dimensional integer array $A[0..n-1, 0..n-1]$ 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](#)

[Answer](#)

3.2.2 Arrays: GATE1994-25 top

<http://gateoverflow.in/2521>

An array A contains n integers in non-decreasing order, $A[1] \leq A[2] \leq \dots \leq A[n]$. 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

Answer

3.2.3 Arrays: GATE1994_1.11 [top](#)

<http://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. $j + \frac{i(i-1)}{2}$
- D. $i + \frac{j(j-1)}{2}$

gate1994 data-structure arrays normal

Answer

3.2.4 Arrays: GATE1997-17 [top](#)

<http://gateoverflow.in/2277>

An array A contains $n \geq 1$ positive integers in the locations $A[1], A[2], \dots, A[n]$. The following program fragment prints the length of a shortest sequence of consecutive elements of A , $A[i], A[i+1], \dots, A[j]$ 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 := 0
min:=n; finish:=false;
while not finish do
  if 0 then
    if j=n then finish:=true
    else
      begin
        j:=j+1;
        sum:= 0
      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

Answer

3.2.5 Arrays: GATE1998_2.14 [top](#)

<http://gateoverflow.in/1688>

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](#)
Answer

3.2.6 Arrays: GATE2000-1.2 [top](#)

<http://gateoverflow.in/625>

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

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

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](#)
Answer

3.2.7 Arrays: GATE2000-15 [top](#)

<http://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".
- 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](#)
Answer

3.2.8 Arrays: GATE2005-5 [top](#)

<http://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](#)

Answer

3.2.9 Arrays: GATE2013-50 [top](#)

<http://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](#)
Answer

3.2.10 Arrays: GATE2013-51 [top](#)

<http://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

[gate2013](#) [data-structure](#) [arrays](#) [normal](#)
Answer

3.2.11 Arrays: GATE2014-3-42 [top](#)

<http://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](#)

[Answer](#)

3.2.12 Arrays: GATE2015-2_31 [top](#)

<http://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](#)

[Answer](#)

3.2.13 Arrays: ISI 2015 PCB C2 B [top](#)

<http://gateoverflow.in/12088>

You are given a array A of size n . Your are told that A comprises three consecutive runs - first a run of a 's, the a run of b 's and finally a run of c 's. Moreover, you are provided an index of i such that $A[i] = b$. Design an $O(\log n)$ time algorithm to determine the number of b 's (i.e., length of the second run) in A .

[data-structure](#) [arrays](#) [isi2015](#)

[Answer](#)

3.2.14 Arrays: TIFR2011-B-30 [top](#)

<http://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](#)

[Answer](#)

Answers: Arrays

3.2.1 Arrays: GATE1993-12 [top](#)

<http://gateoverflow.in/2309>



Selected Answer

We have to traverse all elements in array. The code is doing this row wise.

```
begin
    max:=A[0,0], i:=0, j:=0;
    while (i < n) do
    begin
        if A[i, j]>max then max:=A[i, j];
        if (j < n-1) then j:=j+1;
        else begin
            j:=0;
            i:=i++;
        end
    end
end
```

5 votes

-- Arjun Suresh (287k points)

3.2.2 Arrays: GATE1994-25 [top](#)

<http://gateoverflow.in/2521>

Selected Answer

```
i = 1;
j = n;
while(i != j) {
    if(A[i] + A[j] == M) break;
    else if(A[i] + A[j] < M) i++;
    else j--;
}
```

14 votes

-- ankitrokdeonsns (9.1k points)

3.2.3 Arrays: GATE1994_1.11 [top](#)

<http://gateoverflow.in/2452>

Selected Answer

$j + (\text{sum of natural number till } i-1)$ because if you form a lower triangular matrix it contains elements in rows 1,2,3,...

So C is the correct answer.

PS: Though not mentioned in question, from options it is clear that array index starts from 1 and not 0.

16 votes

-- Bhagirathi Nayak (13.1k points)

answer is C. but the option given is wrong. it is $(j-i) + i(i-1)/2$

10 votes

-- Gate Keeda (18.8k points)

3.2.4 Arrays: GATE1997-17 [top](#)

<http://gateoverflow.in/2277>

Selected Answer

```

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

```

Algorithm

'i' indicates the starting marker and 'j' acts as ending marker for the sum sequence. 'sum' is initialised as the first element in the array because the algorithm proceeds by taking the sum of remaining elements. 'finish' is a boolean variable that indicates exit from the loop.

After entering the loop for the first time with 'finish' as false, the sum is checked if it's strictly less than "M". If that's the case j is incremented and the sum is modified to sum + A[j]. When 'sum' becomes greater than or equal to 'M', 'min' is modified to the latest number of elements that make the sum greater than or equal to 'M' and then, the first element is stripped off from the sum and 'i' is incremented by one to move the initial marker of the sum sequence. The loop runs till 'j' reaches the end of the array.

The algorithm keeps track of 'min' i.e. the number of elements in the minimum sum sequence. This is very similar to the way we find the minimum value in an array by modifying the min value whenever a lesser value is encountered.

6 votes

-- krish__ (717 points)

3.2.5 Arrays: GATE1998_2.14 [top](#)

<http://gateoverflow.in/1628>



Selected Answer

A [LB₁.....UB₁,LB₂.....UB₂]

BA = Base address.

C = size of each element.

Row major order.

$$\text{Loc}(a[i][j]) = BA + [(i-LB_1)(UB_2 - LB_2 + 1) + (j - LB_2)] * C.$$

Column Major order

$$\text{Loc}(a[i][j]) = BA + [(j-LB_2)(UB_1 - LB_1 + 1) + (i - LB_1)] * C.$$

substituting the values. answer is A.

17 votes

-- Gate Keeda (18.8k points)

3.2.6 Arrays: GATE2000-1.2 [top](#)

<http://gateoverflow.in/625>



Selected Answer

The sum of the i^{th} row and i^{th} column is 0 as shown below. Since, the numbers of rows = no. of columns, the total sum will be 0.

0	-1	-2	-3	-4
1	0	-1	-2	-3
2	1	0	-1	-2
3	2	1	0	-1
4	3	2	1	0

12 votes

-- Arjun Suresh (287k points)

3.2.7 Arrays: GATE2000-15 [top](#)

<http://gateoverflow.in/686>



Selected Answer

a)

Initially count= 0;

When we call set(7) - count=1, q[1]=7, p[7]= 1;

when we call set(3) - count=2, q[2]=3, p[3] =2;

when we call set(9) - count=3, q[3]=9, p[9] = 3;

b) Ans- "The first count elements of (array q) contain values i such that set (i) has been called".

c) If set(i) has not been called for some i, then regardless of what p[i] contains, When we call is_set(i) then

```
if (q[p[i]] != i)
    return false;
will always execute, because if set(i) is not called then p[i] != count(any) and for then same count q[count] != i. So if statement will be true and will return false.
```

4 votes

-- Dhananjay Kumar Sharma (24.9k points)

3.2.8 Arrays: GATE2005-5 [top](#)

<http://gateoverflow.in/1347>



Selected Answer

as we our area of interest is only the 50 numbers so take An array of 50 numbers where A[0] corresponds to 51...A[49] corresponds to 100 then after reading an input just increment the counter in correct position as said above

13 votes

-- Bhagirathi Nayak (13.1k points)

3.2.9 Arrays: GATE2013-50 [top](#)

<http://gateoverflow.in/1557>



Selected Answer

A	d	a	b		d	e
---	---	---	---	--	---	---

oldc	a	b	c
newc	d	a	b

for (1)

Here when the element of array A and oldc match , we replace that array element of A with array element of newc . For every element of A array update occurs maximum one time.

Similarly for (2) array element of A has updated with array element of newc less than or equal to one time,

A	
oldc	b c a
newc	c d a

for {3}

Now, for (3) when $i=0$, value of A match with oldc[2] i.e.'a' , and replace with newc[2] i.e. also 'a'. So, no changes when $i=1$ value of array A[1]='b'

match with oldc[0]='b' and replace with newc[0]='c'.

Now, A[1]='c' which equal with next element of oldc[1]='c'.

So, replace again with newc[1]='d'.

Now, we can say here in array A[1] value replace with newc[0] value , and that newc[0] value replace with next newc[1] value.

oldc
newc

for {4}

Similarly for (4) here 2 times replacement for A[0] with element newc[0] and newc[1]

Updating of newc value with another newc value is calling flaw here

So Ans B

5 votes

-- srestha (55.2k points)

The test cases 3 and 4 are the only cases that capture the flaw. The code doesn't work properly when an old character is replaced by a new character and the new character is again replaced by another new character. This doesn't happen in test cases (1) and (2), it happens only in cases (3) and (4).

50. B

51. C

10 votes

-- Vikrant Singh (13.4k points)

3.2.10 Arrays: GATE2013-51 [top](#)

<http://gateoverflow.in/43291>

Selected Answer	
A	
oldc	a b c
newc	d a b

for {1}

Here when the element of array A and oldc match , we replace that array element of A with array element of newc . For every element of A array update occurs maximum one time.

Similarly for (2) array element of A has updated with array element of newc less than or equal to one time,

A	
oldc	b c a
newc	c d a

for {3}

Now, for (3) when $i=0$, value of A match with oldc[2] i.e.'a' , and replace with newc[2] i.e. also 'a'. So, no changes when $i=1$ value of array A[1]='b'

match with oldc[0]='b' and replace with newc[0]='c'.

Now, A[1]='c' which equal with next element of oldc[1]='c'.

So, replace again with newc[1]='d'.

Now, we can say here in array A[1] value replace with newc[0] value , and that newc[0] value replace with next newc[1] value.

oldc
newc

for {4}

Similarly for (4) here 2 times replacement for A[0] with element newc[0] and newc[1]

Updating of newc value with another newc value is calling flaw here

So Ans (C)

6 votes

-- srestha (55.2k points)

3.2.11 Arrays: GATE2014-3-42 [top](#)

<http://gateoverflow.in/2076>



Selected Answer

B)....

12 votes

-- Siddhartha Datta (199 points)

3.2.12 Arrays: GATE2015-2_31 [top](#)

<http://gateoverflow.in/8148>



Selected Answer

The answer should be 5.

1.We first need to shift 2 in place of 1 keeping 5 AND 14 intact as it isn't mentioned in the question that the entire row elements move.

2. 4 is shifted up,next to 2(keeping 12 and infinity intact in column 2).

3. Now in second row 6 is shifted left.

4.18 shifts up to the second row

5. And finally 25 is shifted left to the third column.

So this takes 5 moves and still maintains the tableau property. Also infinity is placed to the right of 25 and below

23(unfilled entries to be filled with ∞). The final table would look as follows.

2	4	5	14
3	6	18	23
10	12	25	∞
31	∞	∞	∞

18 votes

-- Aman verma (211 points)

3.2.13 Arrays: ISI 2015 PCB C2 B [top](#)

<http://gateoverflow.in/120888>

Hint: Modify Binary Search.

Solution:

You should use binary search twice first to find the first occurrence of b . Then apply it again to find the last occurrence.

The following code does this.

BinarySearchForFirst

```
int binarySearchForFirst(int l, int r) {
    int mid;
    while(l <= r){
        mid = l+(r-l)/2;
        if(arya[mid]=='b' && arya[mid-1]=='a') //the desired case
            return mid;
        if(arya[mid-1]=='b') //we're to the right of required index
            r = mid;
            continue;
        if(arya[mid]=='a') //we're to the left or required index
            l = mid;
            continue;
    }
}
```

This returns the first index say i

Similarly find the second index j .

Then the answer is $j - i + 1$

3 votes

-- Akshay Arora (2.2k points)

3.2.14 Arrays: TIFR2011-B-30 [top](#)

<http://gateoverflow.in/20611>



Selected Answer

Option b)
 B is a permutation of array A .

Infact, B gives the reverse index of all the elements of array A . Since the array A contains numbers $[1..n]$ mapped to the locations $[1..n]$ and A is a permutation of the numbers $[1..n]$, the array B will also be a permutation of the numbers $[1..n]$.

For example:

index	1	2	3	4	5	6	7	8
A	5	1	3	7	6	2	8	4
B	2	6	3	8	1	5	4	7

To see that option c is incorrect, let array C be the array attained from doing the same transformation twice, that is, $C[B[i]] = i, \forall i \in [1..n]$. We get,

index	1	2	3	4	5	6	7	8
A	5	1	3	7	6	2	8	4
B	2	6	3	8	1	5	4	7
C	5	1	3	7	6	2	8	4

We can see that $C = A$, which makes option c incorrect.

12 votes

-- Pragy Agarwal (19.3k points)

3.3

Binary Search Tree(29) [top](#)

3.3.1 Binary Search Tree: GATE 2016-2-40 [top](#)

<http://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.

[gate2016-2](#) [data-structure](#) [binary-search-tree](#) [normal](#) [numerical-answers](#)

[Answer](#)

3.3.2 Binary Search Tree: GATE1996_2.14 [top](#)

<http://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

- A. (4, 7)
- B. (7, 4)
- C. (8, 3)
- D. (3, 8)

[gate1996](#) [data-structure](#) [binary-search-tree](#) [normal](#)

[Answer](#)

3.3.3 Binary Search Tree: GATE1996_4 [top](#)

<http://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](#)

Answer**3.3.4 Binary Search Tree: GATE2001-14** [top](#)<http://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](#)**Answer****3.3.5 Binary Search Tree: GATE2003-19, ISRO2009-24** [top](#)<http://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](#)**Answer****3.3.6 Binary Search Tree: GATE2003-6** [top](#)<http://gateoverflow.in/897>

Let $T(n)$ be the number of different binary search trees on n distinct elements.

Then $T(n) = \sum_{k=1}^n T(k-1)T(n-k)$, where x is

- A. $n - k + 1$
- B. $n - k$
- C. $n - k - 1$
- D. $n - k - 2$

[gate2003](#) [normal](#) [binary-search-tree](#)**Answer****3.3.7 Binary Search Tree: GATE2003-63, ISRO2009-25** [top](#)<http://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
- 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](#)

[Answer](#)

3.3.8 Binary Search Tree: GATE2004-4, ISRO2009-26 [top](#)

<http://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](#)

[Answer](#)

3.3.9 Binary Search Tree: GATE2004-85 [top](#)

<http://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:

$$\min \left(\begin{array}{l} \text{number of leaf-nodes} \\ \text{in left-subtree of } x \\ , \end{array} \begin{array}{l} \text{number of leaf-nodes} \\ \text{in right-subtree of } x \end{array} \right)$$

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](#)

[Answer](#)

3.3.10 Binary Search Tree: GATE2005-IT-12 [top](#)

<http://gateoverflow.in/3757>

The numbers $1, 2, \dots, n$ 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](#)

[Answer](#)

3.3.11 Binary Search Tree: GATE2005-IT-55 [top](#)

<http://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](#)

[Answer](#)

3.3.12 Binary Search Tree: GATE2006-IT-45 [top](#)

<http://gateoverflow.in/358>

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](#)

[Answer](#)

3.3.13 Binary Search Tree: GATE2007-IT-29 [top](#)

<http://gateoverflow.in/346>

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

[gate2007-it](#) [data-structure](#) [binary-search-tree](#) [normal](#)

[Answer](#)

3.3.14 Binary Search Tree: GATE2008-46 [top](#)

<http://gateoverflow.in/458>

You are given the postorder traversal, P , of a binary search tree on the n elements $1, 2, \dots, n$. 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](#)

[Answer](#)

3.3.15 Binary Search Tree: GATE2008-IT-12 [top](#)

<http://gateoverflow.in/327>

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](#)
Answer

3.3.16 Binary Search Tree: GATE2008-IT-71 [top](#)

<http://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](#)
Answer

3.3.17 Binary Search Tree: GATE2008-IT-72 [top](#)

<http://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?

- 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](#)
Answer

3.3.18 Binary Search Tree: GATE2008-IT-73 [top](#)

<http://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](#)
Answer

3.3.19 Binary Search Tree: GATE2009-37 [top](#)

<http://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](#)

Answer**3.3.20 Binary Search Tree: GATE2012-5** [top](#)<http://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](#)
Answer**3.3.21 Binary Search Tree: GATE2013-43** [top](#)<http://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](#)
Answer**3.3.22 Binary Search Tree: GATE2013_7** [top](#)<http://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](#)
Answer**3.3.23 Binary Search Tree: GATE2014-3-39** [top](#)<http://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](#)
Answer**3.3.24 Binary Search Tree: GATE2015-1_10** [top](#)<http://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

gate2015-1 | data-structure | binary-search-tree | easy

Answer

3.3.25 Binary Search Tree: GATE2015-1_23 [top](#)

<http://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

Answer

3.3.26 Binary Search Tree: GATE2015-3_13 [top](#)

<http://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

Answer

3.3.27 Binary Search Tree: GATE2017-1-6 [top](#)

<http://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

Answer

3.3.28 Binary Search Tree: GATE2017-2-36 [top](#)

<http://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

Answer

3.3.29 Binary Search Tree: TIFR2010-B-26 [top](#)

<http://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.

tifr2010 | binary-search-tree

Answer

Answers: Binary Search Tree

3.3.1 Binary Search Tree: GATE 2016-2-40 [top](#)

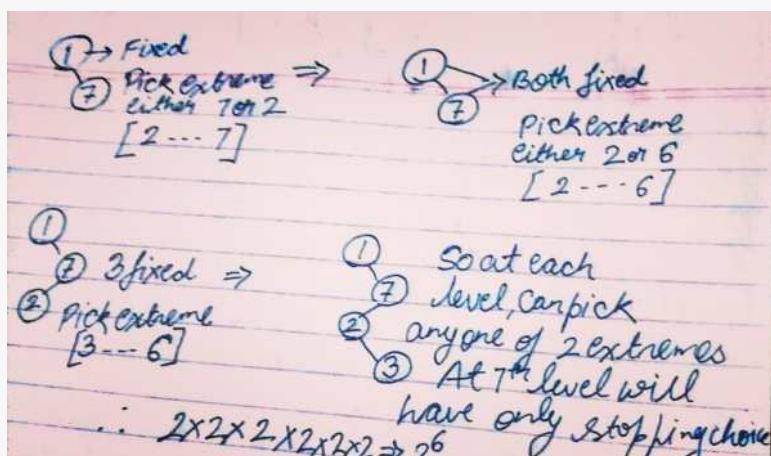
<http://gateoverflow.in/39586>

Selected Answer

We need to fill 7 levels with 7 elements. So, 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.

47 votes

-- Arjun Suresh (287k points)



WELL its RECURRENCE QUESTION ,,,but for that need to observe the pattern ..first and then THE RECURRENCE RELATION WILL BE $N(h)=2N(h-1)$

25 votes

-- Deepesh Kataria (1.8k points)

3.3.2 Binary Search Tree: GATE1996_2.14 [top](#)

<http://gateoverflow.in/2743>

Selected Answer

B.

Root will be 50. now insert one by one, greater to 50 in the right sub tree, lesser in left sub tree.

Or you can simply count the number looking at the i/p. less than 50 are 7. more than 50 are 4.

7 votes

-- Gate Keeda (18.8k points)

3.3.3 Binary Search Tree: GATE1996_4 [top](#)

<http://gateoverflow.in/2756>



Selected Answer

B and E are not possible.

rest all i/p's will have binary trees with only one child. but B and E will have two childs at a point. therefore the probe sequence will not be possible.

For better clarification, make BST's for the given i/p's and probe for 43.

12 votes

-- Gate Keeda (18.8k points)

3.3.4 Binary Search Tree: GATE2001-14 [top](#)

<http://gateoverflow.in/755>

ans for C :

S1: $y = \text{depth}(t \rightarrow \text{right})$

S2: $\text{return}(1 + x)$

S3: $\text{return}(1 + y)$

8 votes

-- jayendra (7.7k points)

3.3.5 Binary Search Tree: GATE2003-19, ISRO2009-24 [top](#)

<http://gateoverflow.in/909>



Selected Answer

In-order traversal returns the elements in sorted order.

Therefore, it's option C

10 votes

-- Gate_15_isHere (639 points)

3.3.6 Binary Search Tree: GATE2003-6 [top](#)

<http://gateoverflow.in/897>



Selected Answer

The summation is for each node, if that node happens to be the root. When a node is root, it will have $(k-1)$ nodes on the left sub tree (k being any number) and correspondingly $(n-k)$ elements on the right sub tree. So, we can write recurrence $T(k-1) * T(n-k)$ for the number of distinct binary search trees, as the numbers on left and right sub trees form BSTs independent of each other and only a difference in one of the sub trees produces a difference in the tree. Hence, answer is B.

Knowing the direct formula can also help in getting the answer but is not recommended.

http://gatecse.in/wiki/Number_of_Binary_trees_possible_with_n_nodes

14 votes

-- Arjun Suresh (287k points)

left subtree + root + right subtree = total node
 $(k-1) + 1 + x = n$
 $x = n - k$

19 votes

-- Digvijay (46k points)

3.3.7 Binary Search Tree: GATE2003-63, ISRO2009-25 [top](#)

<http://gateoverflow.in/950>



Selected Answer

Balanced search tree have height $\log n$

Deletion of smallest element will take $O(\log n)$ time

Finding a element is present/not and doing insertion: $O(\log n)$

Heap(MIN) is also an almost complete binary tree have height $\log n$

Deletion of smallest element will take $O(\log n)$ time (root element removal, replace with last element +balancing)

Finding a element is present/not and insertion: Finding only takes $O(n)$, insertion then balancing take $O(\log n)$. So, total $O(n) + O(\log n) = O(n)$.

Ans B

(even if its maxheap our ans does not change only time for deletion of min will increase $O(n)$)

19 votes

-- Anurag Semwal (7.6k points)

3.3.8 Binary Search Tree: GATE2004-4, ISRO2009-26 [top](#)

<http://gateoverflow.in/1001>



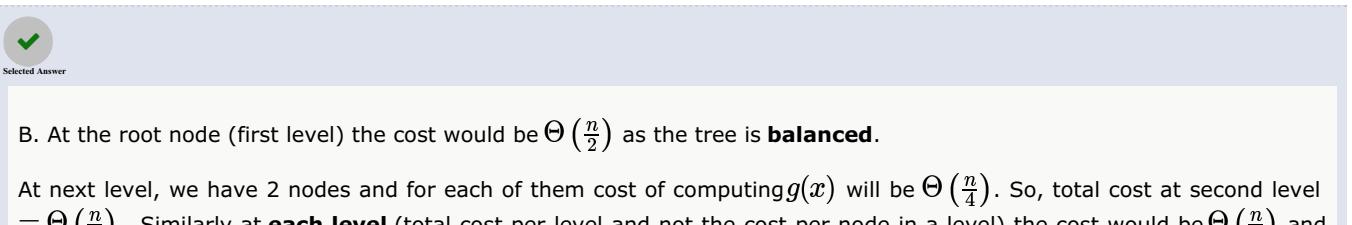
Height is 3

10 votes

-- Prashant Singh (47.6k points)

3.3.9 Binary Search Tree: GATE2004-85 [top](#)

<http://gateoverflow.in/1079>



PS: Even if we change **min** to **max** in the definition of $g(x)$ we get the same answer.

21 votes

-- Shaun Patel (6.9k points)

3.3.10 Binary Search Tree: GATE2005-IT-12 [top](#)

<http://gateoverflow.in/3757>

Selected Answer

Option c

GATE 2005 IT
Solution

p : # elements in RST

→ Depending on x some number will go
LST & RST

→
 Remaining elements for LST = $n - (p+1)$

$1, 2, \dots, n-(p+1)$ $n-p$ $(n-p)+1, \dots, n$
 LST Root RST

$n - (p+1)$ elements 1 element p - element &

13 votes

-- pC (20.7k points)

from 1,...n elements p elements are on the right. so root or first inserted will be at $n-p$

11 votes

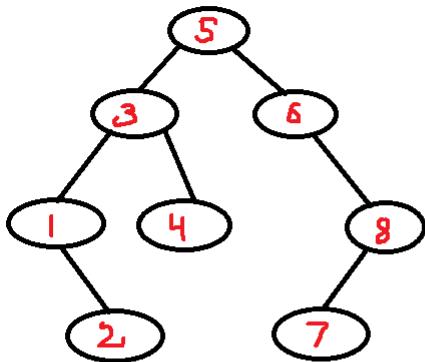
-- Sankaranarayanan P.N (11k points)

3.3.11 Binary Search Tree: GATE2005-IT-55 [top](#)

<http://gateoverflow.in/3816>

Selected Answer

Answer is D.



12 votes

-- Gate Keeda (18.8k points)

3.3.12 Binary Search Tree: GATE2006-IT-45 [top](#)

<http://gateoverflow.in/3588>



Selected Answer

in option C search sequence progress in ...47,68,43,..

at 47 we see that search key 55 is greater and it will be on right side of 47. so in further comparison a value less than 47 will not come

hence option c is wrong

14 votes

-- Sankaranarayanan P.N (11k points)

3.3.13 Binary Search Tree: GATE2007-IT-29 [top](#)

<http://gateoverflow.in/3462>



Selected Answer

10, 20, 40, 50, 70 80, 90

In BST search we if we go from say 10 to 40 while searching for 60, we will never encounter 20. So, 10, 20, 40 and 50 visited, means they are visited in order. Similarly, 90, 80 and 70 are visited in order. So, our required answer will be

No. of possible permutations of 7 numbers

No. of possible permutations of numbers smaller than 60 × No. of possible permutations of numbers larger than 60

(Since only one permutation is valid for both the smaller set of numbers as well as larger set of numbers)

$$= \frac{7!}{43!}$$

$$= 35$$

36 votes

-- Arjun Suresh (287k points)

3.3.14 Binary Search Tree: GATE2008-46 [top](#)

<http://gateoverflow.in/458>

Selected Answer

Last element in post order is the root of tree- find this element in inorder- $\log n$ time.

Now as in quick sort consider this as pivot and split the post order array into 2- possible because all elements smaller than pivot goes to left and all elements larger than pivot goes to right and suppose we have x elements smaller than pivot, these elements will be same in both inorder as well as postorder (order may change). We already got the root, now left child is the left split and right child is the right split.

So, doing this recursively gives time complexity of this approach as

$$T(n) = T(k) + T(n - k - 1) + \log n$$

Solving would give $T(n) = O(n \log n)$ in worst case, by putting $k = 0$ and shown at bottom.

But searching for an element in the inorder traversal of given BST can be done in $O(1)$ because we have n elements from $1..n$ so there is no need to search for an element- if last element in post order is say 5 we take it as root and since 4 elements (1..4) are smaller we split the post order array in to two- (first 4 elements), (6th element onward) and solve recursively. Time complexity for this would be

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

which gives $T(n) = O(n)$.

Since we know that all elements must be traversed at least once, $T(n) = \Omega(n)$ also and so

$$T(n) = \Theta(n).$$

The following code is doing this.

```
//Install graphviz (sudo apt-get install graphviz on Ubuntu) to view output tree
#include<stdio.h>
#include<stdlib.h>
struct tree
{
    struct tree* left;
    struct tree* right;
    int x;
};
struct tree* makenode(int x)
{
    struct tree * root = malloc(sizeof(struct tree));
    root -> x = x;
    root -> left = root -> right = NULL;
    return root;
}

struct tree* makeBST(int *post, int start, int n, int inorder)
{
    if(n <= 0)
        return NULL;
    int pivot = post[start + n - 1];
    struct tree * root = makenode(pivot);
    root -> left = makeBST(post, start, pivot-1 - inorder, inorder );
    root -> right = makeBST(post, pivot - inorder - 1, n - (pivot - inorder), pivot);
    return root;
}
void preorder(struct tree* node)
{
    if(node == NULL)
        return;
    printf("%d ", node->x);
    preorder(node->left);
    preorder(node->right);
}
void printdot(struct tree* node, FILE * f)
{
    if(node == NULL)
        return;
    if(node-> left != NULL)
    {
        fprintf(f, "%d -- %d;\n", node->x, node->left->x);
    }
    if(node-> right != NULL)
    {
        fprintf(f, "%d -- %d;\n", node->x, node->right->x);
    }
}
```

```

    printdot(node->left, f);
    printdot(node->right, f);
}

int main() {
    int i, n, *a;
    printf("Enter n: ");
    scanf("%d", &n);
    a = malloc(n * sizeof(int));
    printf ("Enter post order traversal: ");
    for(i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
    struct tree * tree = makeBST(a, 0, n, 0);
    printf("Pre order traversal is : ");
    preorder(tree);
    printf("\n");
    FILE * f = fopen("tree.dot", "w");
    fprintf(f, "graph tree {\n");
    printdot(tree, f);
    fprintf(f, "}\n");
    fclose(f);

#if defined(__linux__) || (defined(__APPLE__) && defined(__MACH__)) || (defined (__gnu_linux__))
    system("dot -Tpng tree.dot -o output.png; eog output.png");
#endif
}

```

$$T(n) = T(k) + T(n-k-1) + \log n$$

Solving would give $T(n) = O(n \log n)$, by putting $k=0$,
 $T(n) = T(0) + T(n-1) + \log n, \Rightarrow T(n) = O(1) + \log n + \log(n-1) + \log(n-2) + \dots + \log 1 \Rightarrow T(n) = n + \log n$
[\(Stirling's Approximation\)](#)

20 votes

-- Arjun Suresh (287k points)

3.3.15 Binary Search Tree: GATE2008-IT-12 [top](#)

<http://gateoverflow.in/3272>



Selected Answer

A) is true as AVL tree is a balanced search tree that has time complexity of searching $\Theta(\log n)$, but in binary search tree, we can have a completely left/right skewed tree, in which search is $O(n)$.

10 votes

-- Happy Mittal (10.7k points)

3.3.16 Binary Search Tree: GATE2008-IT-71 [top](#)

<http://gateoverflow.in/3385>



Selected Answer

Answer: D

- I. no need to go from 285 to 376 as 273 is less than 285.
- II. no need to go from 381 to 472 as 273 is less than 381.
- IV. no need to go from 395 to 463 as 273 is less than 395.

13 votes

-- Rajarshi Sarkar (34.4k points)

3.3.17 Binary Search Tree: GATE2008-IT-72 [top](#)

<http://gateoverflow.in/3386>



Selected Answer

A) Incorrect because I & IV are not in ascending order.(Inorder sequence of BST is in increasing order)

B) I is a preorder sequence of some BST with 439 as the root . False because if 439 is root, it should be first element in

preorder.

D) IV is a postorder sequence of some BST with 149 as the root, False because if 149 is root, it should be last element in postorder

C) This is correct.

9 votes

-- Akash (41.5k points)

3.3.18 Binary Search Tree: GATE2008-IT-73 [top](#)



Selected Answer

for distinct bst we apply this formula

$$C(2n, n) / n+1$$

$$n=3 \text{ here so } C(6, 3)=20$$

$$\text{abd } 20/4=5$$

so ans is 5

13 votes

-- Abhimanyu Kumar (189 points)

3.3.19 Binary Search Tree: GATE2009-37 [top](#)



Selected Answer

Answer is B.

With 1 node height is 0.

Max height will come when each level contain min nodes.

Minimum Nodes in an AVL tree with height n is $H(n) = H(n-1) + H(n-2) + 1$.

$$H(0) = 1.$$

$$H(1) = 2$$

$$H(2) = H(1) + H(0) + 1 = 2+1+1 = 4$$

$$H(3) = H(2) + H(1) + 1 = 4+2+1 = 7.$$

So the max height with 7 nodes is 3.

18 votes

-- Gate Keeda (18.8k points)

3.3.20 Binary Search Tree: GATE2012-5 [top](#)



Selected Answer

Binary search takes $\Theta(\log n)$ for n elements in the worst case. So, with $(n2^n)$ elements, the worst case time will be

$$\Theta(\log(n2^n))$$

$$= \Theta(\log n + \log 2^n)$$

$$= \Theta(\log n + n)$$

$$= \Theta(n)$$

16 votes

-- Arjun Suresh (287k points)

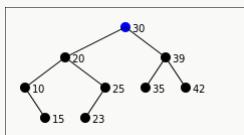
3.3.21 Binary Search Tree: GATE2013-43 [top](#)



Selected Answer

Since it is a binary search tree, its inorder traversal produces a sorted sequence i.e. 10, 15, 20, 23, 25, 30, 35, 39, 42.

Now given inorder and preorder traversals, we get following tree :



From this, we can give postorder traversal sequence as 15,10,23,25,20,35,42,39,30 i.e. option (D).

13 votes

-- Happy Mittal (10.7k points)

3.3.22 Binary Search Tree: GATE2013_7 [top](#)

<http://gateoverflow.in/1416>



Selected Answer

Option (C) is True .

Suppose that we need to insert a node z such that k = key[z]. Using binary search we find a nil such that replacing it by z does not break the BST-property

BST-Insert(x, z, k)

1. : if x = nil then return "Error"
2. : y \leftarrow x
3. : while true do {
4. : if key[y] < k
5. : then z \leftarrow left[y]
6. : else z \leftarrow right[y]
7. : if z = nil break
8. : }
9. : if key[y] > k then left[y] \leftarrow z
10. : else right[p[y]] \leftarrow z

Time Complexity Analysis :

1. Best Case = O(1) , When it is smallest/greatest element and BST contains only all greater/smaller element than inserting element respectively.
2. Avg Case = O(log n) , When it belongs between some elements .
3. Worst Case = O(n) , When it is smallest/greatest element and BST contains only all smaller/greater element than inserting element respectively.

11 votes

-- Bhagirathi Nayak (13.1k points)

3.3.23 Binary Search Tree: GATE2014-3-39 [top](#)

<http://gateoverflow.in/2073>



Selected Answer

```

int getSum(node *root, int L, int H)
{
    // Base Case
    if (root == NULL)
        return 0;

    if (root->key < L)
        return getSum(root->right, L, H);

    if (root->key > H)
        return getSum(root->left, L, H);

    if (root->key >= L && root->key <=H)
        return getSum(root->left, L, H) + root->key +
               getSum(root->right, L, H);
}
  
```

it will be $O(\log n + m)$

it can go max of log depth to find nodes in range and function calls for nodes in range ie m

Note that the code first traverses across height to find the node which lies in range. Once such a node is found, it recurs for left and right children. Two recursive calls are made only if the node is in range. So for every node that is in range, we make at most one extra call (here extra call means calling for a node that is not in range).

source:<http://geeksquiz.com/gate-gate-cs-2014-set-3-question-49/>

7 votes

-- Anurag Semwal (7.6k points)

In worst case for finding

L and

H it will take

$O(\log n)$ time as the given tree is balanced binary search tree.

Now there are

m elements between

L and

H . So to traverse m element it will take

$O(m)$ time (traversal algorithm given below). So, total

$$O(m + \log n) \implies a = 0, b = 1, c = 1, d = 0 \\ \therefore 0 + (10 \times 1) + (100 \times 1) + (1000 \times 0) = 110.$$

To find all the numbers from L to H we can do an inorder traversal from root and discard all elements before L and after H . But this has $O(n)$ time complexity. So, we can do a modification to inorder traversal and combine with binary search as follows:

1. Find L using binary search and keep all nodes encountered in the search in a stack.
2. After finding L add it to stack as well and initialize sum = 0.
3. Now, for all nodes in stack, do an inorder traversal starting from their right node and adding the node value to sum. If H is found, stop the algorithm.

29 votes

-- Kalpish Singhal (2k points)

3.3.24 Binary Search Tree: GATE2015-1_10 [top](#)

<http://gateoverflow.in/8129>



Selected Answer

Quick Solution

Inorder traversal of key are always in ascending order.

So Here I & IV th sequence are in ascending order so Option A is Ans.

4 votes

-- Rajesh Pradhan (16.9k points)

3.3.25 Binary Search Tree: GATE2015-1_23 [top](#)

<http://gateoverflow.in/8221>



Selected Answer

option b, both happens when the BST is skewed.

16 votes

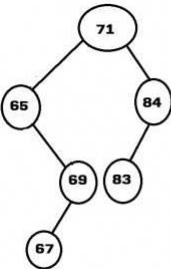
-- GATERUSH (1.1k points)

3.3.26 Binary Search Tree: GATE2015-3_13 [top](#)

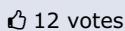
<http://gateoverflow.in/8409>



Selected Answer



on last level 67 , hence option B is True.



-- Bran Stark (391 points)

3.3.27 Binary Search Tree: GATE2017-1-6 top

1000-10000

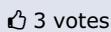


Selected Answer

Min height of a binary tree is given by $\log_2(n+1) - 1 = 4 - 1 = 3$

max height of the binary tree is $n - 1$ (skewed binary tree = $15 - 1 = 14$)

Ans:B)3,14



-- Arnabi (6k points)

3.3.28 Binary Search Tree: GATE2017-2-36 top

http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=1619270

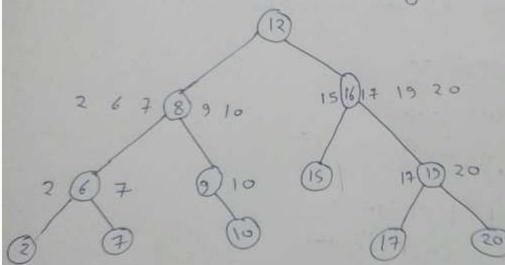


Selected Answer

Pre: (12) 8, 6, 2, 7, 9, 10, 16, 15, 19, 17, 20

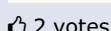
inorder of BST must be sorted in increasing order!

IN order: left right root right left



$$\text{so postorder} = 2 \ 7 \ 6 \ 10 \ 9 \ 8 \ 15 \ 17 \ 20 \ 19 \ 18 \ 12$$

so ans is B



-- 2018 (4-8k points)

3.3.29 Binary Search Tree: TIFR2010-B-26 top

Answer should be (D).

We first find a in the BST in $O(\log n)$ time. Now there are two possibilities, b can be in the right subtree of a or b can be in the right subtree of any of the parents of a . For the first case, we simply search for b , in the right subtree of a and at each step we add the number of elements in the left subtree +1 (**BST being balanced, this can be retrieved from the depth of the node without any finding method**), if we are moving right and simply 1 if we are moving left. When we find b , this sum will give us the required number of elements. This requires $O(\log n)$ additions but we can do smarter by doing

$$N(a) - N(\text{LEFT}(a)) - N(\text{RIGHT}(b))$$

where, $N(x)$ denote the no. of elements in the subtree rooted at x and if $\text{LEFT}(a)$, $\text{RIGHT}(b)$ returning 0 for NULL.

For the second case also we do the same method. But first we find the common ancestor of a and b (possible in $O(\log n)$) - say p and also count the no. of nodes in the right subtree of each node from a to p excluding p . Now, from p to a we proceed the counting as in the earlier case when b was in the subtree at a . So, in the worst case we have to do $O(\log n)$ additions. Here, also we can reduce the no. of additions by doing

$$N(p) - N(\text{LEFT}(a)) - N(\text{RIGHT}(b))$$

where, $N(x)$ denote the no. of elements in the subtree rooted at x and if $\text{LEFT}(a)$, $\text{RIGHT}(b)$ returning 0 for NULL.

17 votes

-- Arjun Suresh (287k points)

3.4

Binary Tree(46) top

3.4.1 Binary Tree: GATE 2016-2-36 top

<http://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 ^ 5 - 3 4 *
- B. - + 1 * 6 7 ^ 2 - 5 * 3 4
- C. - + 1 * 7 6 ^ 2 - 5 * 4 3
- D. 1 7 6 * + 2 5 4 3 * - ^ -

gate2016-2 data-structure binary-tree normal

Answer

3.4.2 Binary Tree: GATE1987-2c top

<http://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

Answer

3.4.3 Binary Tree: GATE1987-2g top

<http://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

Answer**3.4.4 Binary Tree: GATE1987-7b** [top](#)<http://gateoverflow.in/82427>

Construct a binary tree whose preorder traversal is

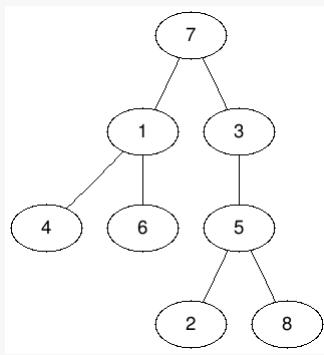
- K L N M P R Q S T

and inorder traversal is

- N L K P R M S Q T

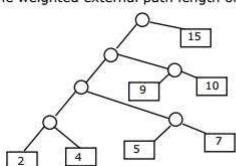
[gate1987](#) [data-structure](#) [binary-tree](#)**Answer****3.4.5 Binary Tree: GATE1991-1,ix** [top](#)<http://gateoverflow.in/502>

If the binary tree in figure is traversed in inorder, then the order in which the nodes will be visited is _____

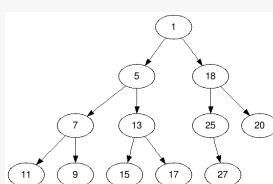
[gate1991](#) [algorithms](#) [binary-tree](#) [easy](#)**Answer****3.4.6 Binary Tree: GATE1991_01,viii** [top](#)<http://gateoverflow.in/506>

The weighted external path length of the binary tree in figure is _____

(viii) The weighted external path length of the binary tree in figure is _____

[gate1991](#) [binary-tree](#) [data-structure](#) [normal](#)**Answer****3.4.7 Binary Tree: GATE1991_14,a** [top](#)<http://gateoverflow.in/541>

Consider the binary tree in the figure below:



(a). What structure is represented by the binary tree?

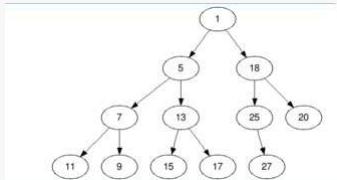
gate1991 | data-structure | binary-tree | time-complexity | normal

Answer

3.4.8 Binary Tree: GATE1991_14,b [top](#)

<http://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.

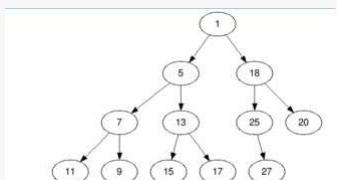
gate1991 | data-structure | binary-tree | normal

Answer

3.4.9 Binary Tree: GATE1991_14,c [top](#)

<http://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

Answer

3.4.10 Binary Tree: GATE1993_16 [top](#)

<http://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

Answer

3.4.11 Binary Tree: GATE1994_8 [top](#)

<http://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

Answer

3.4.12 Binary Tree: GATE1995_1.17 [top](#)

<http://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

Answer

3.4.13 Binary Tree: GATE1995_6 [top](#)<http://gateoverflow.in/2667>

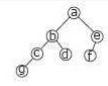
What is the number of binary trees with 3 nodes which when traversed in post-order give the sequence A, B, C ? Draw all these binary trees.

gate1995 data-structure binary-tree normal

Answer

3.4.14 Binary Tree: GATE1996_1.14 [top](#)<http://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"?



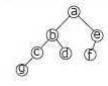
- A. 1
- B. 3
- C. 7
- D. 8

gate1996 data-structure binary-tree normal

Answer

3.4.15 Binary Tree: GATE1996_1.15 [top](#)<http://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

Answer

3.4.16 Binary Tree: GATE1997-4.5 [top](#)<http://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

Answer

3.4.17 Binary Tree: GATE1998_20 [top](#)

<http://gateoverflow.in/1734>

Draw the binary tree with node labels a, b, c, d, e, f and g for which the inorder and postorder traversals result in the following sequences:

Inorder: afbcde

Postorder: afcgedb

gate1998 | data-structure | binary-tree | descriptive

Answer

3.4.18 Binary Tree: GATE2000-1.14 [top](#)

<http://gateoverflow.in/837>

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

Answer

3.4.19 Binary Tree: GATE2000-2.16 [top](#)

<http://gateoverflow.in/863>

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

Answer

3.4.20 Binary Tree: GATE2002-2.12 [top](#)

<http://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

Answer

3.4.21 Binary Tree: GATE2002-6 [top](#)

<http://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

Answer

3.4.22 Binary Tree: GATE2004-35 [top](#)

<http://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](#)

Answer

3.4.23 Binary Tree: GATE2004-43 [top](#)

<http://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);
}
```

The value returned by the function `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

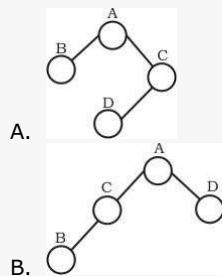
[gate2004](#) [data-structure](#) [binary-tree](#) [normal](#)

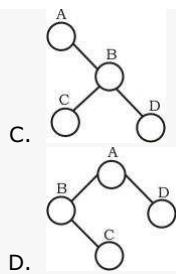
Answer

3.4.24 Binary Tree: GATE2004-IT-54 [top](#)

<http://gateoverflow.in/369>

Which one of the following binary trees has its inorder and preorder traversals as BCAD and ABCD, respectively?




[gate2004-it](#) [binary-tree](#) [easy](#)
[Answer](#)

3.4.25 Binary Tree: GATE2005-33 [top](#)

<http://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](#)
[Answer](#)

3.4.26 Binary Tree: GATE2005-IT-50 [top](#)

<http://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](#)
[Answer](#)

3.4.27 Binary Tree: GATE2006-13 [top](#)

<http://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](#)
[Answer](#)

3.4.28 Binary Tree: GATE2006-IT-71 [top](#)

<http://gateoverflow.in/3815>

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. $\left\lfloor \frac{i}{2} \right\rfloor - 1$

[gate2006-it](#) [data-structure](#) [binary-tree](#) [normal](#)

[Answer](#)

3.4.29 Binary Tree: GATE2006-IT-73 [top](#)

<http://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. $\lfloor \log_2 (i + 1) \rfloor$
- C. $\lfloor \log_2 (i + 1) \rfloor$
- D. $\lfloor \log_2 i \rfloor$

[gate2006-it](#) [data-structure](#) [binary-tree](#) [normal](#)

[Answer](#)

3.4.30 Binary Tree: GATE2006-IT-9 [top](#)

<http://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](#)

[Answer](#)

3.4.31 Binary Tree: GATE2007-12 [top](#)

<http://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](#)

[Answer](#)

3.4.32 Binary Tree: GATE2007-13 [top](#)

<http://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](#)

[Answer](#)

3.4.33 Binary Tree: GATE2007-39, UGCNET-June2015-II-22 [top](#)

<http://gateoverflow.in/1237>

The inorder and preorder traversal of a binary tree are

dbeafcg and abdecfg , respectively

The postorder traversal of the binary tree is:

- A. debfgca
B. edbgfca
C. edbfgca
D. defgbca

[gate2007](#) [data-structure](#) [binary-tree](#) [normal](#) [ugcnetjune2015ii](#)

[Answer](#)

3.4.34 Binary Tree: GATE2007-46 [top](#)

<http://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:

- 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](#)

[Answer](#)

3.4.35 Binary Tree: GATE2008-IT-46 [top](#)

<http://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](#)

[Answer](#)

3.4.36 Binary Tree: GATE2008-IT-76 [top](#)

<http://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. $\lceil ((n_1 + n_2)/2) \rceil$
- D. $n_2 - 1$

[gate2008-it](#) [data-structure](#) [binary-tree](#) [normal](#)

[Answer](#)

3.4.37 Binary Tree: GATE2008-IT-77 [top](#)

<http://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](#)

[Answer](#)

3.4.38 Binary Tree: GATE2010-10 [top](#)

<http://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](#)

[Answer](#)

3.4.39 Binary Tree: GATE2012-47 [top](#)

<http://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`.

```

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
            }
    }
}

```

- 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

Answer

3.4.40 Binary Tree: GATE2014-1-12 [top](#)

<http://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

Answer

3.4.41 Binary Tree: GATE2015-1_25 [top](#)

<http://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

Answer

3.4.42 Binary Tree: GATE2015-2_10 [top](#)

<http://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

Answer

3.4.43 Binary Tree: GATE2015-3_25 [top](#)

<http://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 _____.

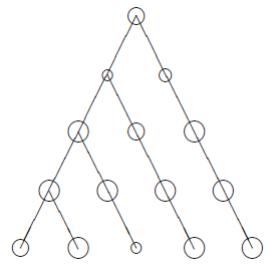
gate2015-3 | data-structure | binary-tree | normal | numerical-answers

Answer

3.4.44 Binary Tree: TIFR2013-B-13 [top](#)

<http://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

Answer

3.4.45 Binary Tree: TIFR2014-B-1 [top](#)

<http://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

Answer

3.4.46 Binary Tree: TIFR2015-B-4 [top](#)

<http://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 $\{1, 2, \dots, 9\}$ 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. $2^4 \cdot 3^2 \cdot 5 \cdot 9 = 6480$
- C. $2^3 \cdot 3 \cdot 5 \cdot 9 = 1080$
- D. $2^4 = 16$
- E. $2^3 \cdot 3^3 = 216$

tifr2015 | binary-tree | permutations-and-combinations

Answer

Answers: Binary Tree

3.4.1 Binary Tree: GATE 2016-2-36 [top](#)

<http://gateoverflow.in/39597>



Selected Answer

Expression given in reverse polish notation (i.e in Post-order)

convert first it into In-order

$3 \ 4 \ * \ 5 \ - \ 2 \wedge \ 6 \ 7 \ * \ 1 \ + \ -$

$(3 * 4) \ 5 \ - \ 2 \wedge \ 6 \ 7 \ * \ 1 \ + \ -$

$((3 * 4) - 5) \ 2 \wedge \ 6 \ 7 \ * \ 1 \ + \ -$

$((3 * 4) - 5) \wedge 2) \ 6 \ 7 \ * \ 1 \ + \ -$

$((3 * 4) - 5) \wedge 2) \ (6 * 7) \ 1 \ + \ -$

$((3 * 4) - 5) \wedge 2) \ ((6 * 7) + 1) \ -$

$((3 * 4) - 5) \wedge 2) \ - \ ((6 * 7) + 1))$

so Inorder expression is $((3 * 4) - 5) \wedge 2) \ - \ ((6 * 7) + 1))$

New-Order traversal is as by ROOT RIGHT LEFT

$((3 * 4) - 5) \wedge 2) \ - \ ((6 * 7) + 1))$

$- \ ((6 * 7) + 1) \ ((3 * 4) - 5) \wedge 2)$

$- \ + 1 \ (6 * 7) \ ((3 * 4) - 5) \wedge 2)$

$- \ + 1 * 7 \ 6 \ ((3 * 4) - 5) \wedge 2)$

$- \ + 1 * 7 \ 6 \wedge 2 \ ((3 * 4) - 5)$

$- \ + 1 * 7 \ 6 \wedge 2 \ - \ 5 \ (3 * 4)$

$- \ + 1 * 7 \ 6 \wedge 2 \ - \ 5 * 4 \ 3$

option C is correct

22 votes

-- Praveen Saini (5.3k points)

Its quite simple actually.

Postorder: left, right, root

Neworder: Root, right, left

(left, right, root) and (Root, right, left) are reverse of each other, right ?

So, these two traversals will be exact reverse of each other! Option (c) ! Answer in few seconds, basic concept ! Observation !

23 votes

-- Ashish Deshmukh (1.5k points)

3.4.2 Binary Tree: GATE1987-2c [top](#)

<http://gateoverflow.in/80579>

Yes it is possible since we can create Binary search tree , we know every Binary search tree is binary tree also but there

are many binary tree possible , so we know that there is many binary tree possible without inorder.

So answer is NO for this question

Refer: <http://www.geeksforgeeks.org/if-you-are-given-two-traversal-sequences-can-you-construct-the-binary-tree/>

5 votes

-- Prashant Singh (47.6k points)

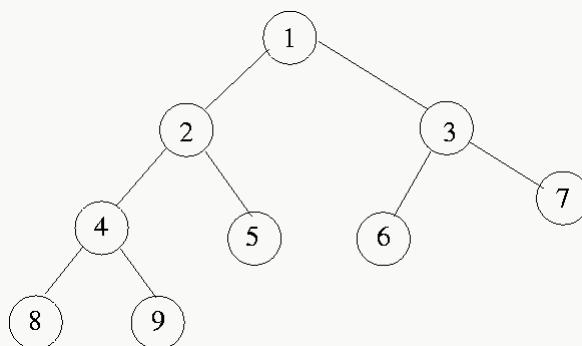
3.4.3 Binary Tree: GATE1987-2g top

<http://gateoverflow.in/80588>



Selected Answer

Condition for binary tree is atmost two immediate child for every internal node



No it is not the condition for binary tree.

5 votes

-- Prashant Singh (47.6k points)

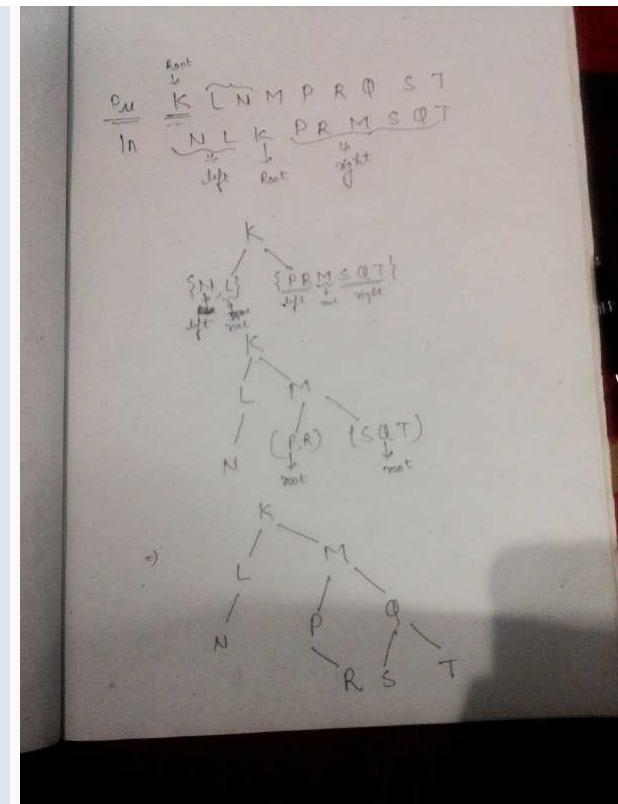
3.4.4 Binary Tree: GATE1987-7b top

<http://gateoverflow.in/82427>



Selected Answer

....



3 votes

-- kirti singh (3.3k points)

3.4.5 Binary Tree: GATE1991-1,ix [top](#)

<http://gateoverflow.in/502>



Selected Answer

41673258

8 votes

-- Keith Kr (6.2k points)

3.4.6 Binary Tree: GATE1991_01,viii [top](#)

<http://gateoverflow.in/508>



Selected Answer

This is straightforward. The nodes of the given tree are given in square boxes. The weights associated with the nodes are the numbers example 15,9,10 etc.

Weighted path length = sigma(for(each node in the tree) (path length)*(weight of the node)).

So answer (written in **path_length*weight** form) = $4*2 + 4*4 + 4*5 + 4*7 + 3*9 + 3*10 + 1*15 = 144$

17 votes

-- arvchamp (171 points)

3.4.7 Binary Tree: GATE1991_14,a [top](#)

<http://gateoverflow.in/541>



Selected Answer

(A) This is min heap. It is obvious looking at tree.

(B) Assuming that we have stored this heap in array structure, Procedure ->

1. Search for element 5 using sequential search in array.

2. Swap it with last element in this case 27.

3. Bubble down it so that min heap property is satisfied.

(C) Deleting element from min heap , $O(\log n)$, but for searching in Heap we need $O(N)$

SO time complexity of sequential search + Delete = $> O(N) + O(\log N) = O(N)$.

We can't use binary search as it is heap.

12 votes

-- Akash (41.5k points)

3.4.8 Binary Tree: GATE1991_14,b top

<http://gateoverflow.in/43026>



Selected Answer

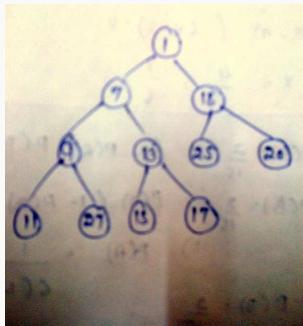
Since given Binary tree is minheap tree.

First swap 27 and 5

Then delete 5

Apply Minheapify

And Structure will be



3 votes

-- Manoj Kumar (37k points)

3.4.9 Binary Tree: GATE1991_14,c top

<http://gateoverflow.in/43027>



Selected Answer

By looking at the values it is clear that It is a **Min-Heap** Data structures. We know that, Heap Data structures are stored in the array.

==> Delete procedure for Min-Heap Data Structure (If you already know value and position of the node):

1) Replace that node with the last element of that tree.

2) Apply Heapify property on that node.

For Example, Let If I want to delete 1, then I will replace that with 27. and apply heapify on that node. Or if i want to delete 5 then I will replace that with 17, and apply heapify on that node.

Time Complexity: In this case time complexity will not be more than $O(\log n)$.

==> Delete procedure for Min-Heap Data Structure (If you know the value but not position) :

1) Find the position of the number by sequential search. (In worst case it will take $O(n)$ time).

2) Replace that node with the last element of that tree.

3) Apply heapify property at that node.

Time Complexity: Wort time complexity of this algorithm will be **O(n + log n)** i.e. **O(n)**.

Note: This is a standard problem of Minimum element deletion from Min-heap tree. Minimum element always resides at top (Root node). We just replace that value with the last element of the tree and apply heapify at the root node. Time complexity of that algorithm is **O(Log n)**.

Here I have written second method only to show that if we have to delete any of the node, and we just know the value but not the position. Since in question it is mentioned that **Arbitrary node**.

13 votes

-- Muktinath Vishwakarma (33.9k points)

3.4.10 Binary Tree: GATE1993_16 [top](#)

<http://gateoverflow.in/231>



Selected Answer

Base Case :- When we have just root then, there are no non leaf nodes. So No of leaves = 1, No of non leaf nodes is = 0. Base case holds.

Induction Hypothesis :- Assume that now for k internal nodes we will have k+1 leaves.

Inducting on no of leaves, Now we add 2 more leaves to this tree. One of k+1 leaf will become internal node. So now we will have k+1 internal node. No of leafs will be K+ 1 - 1(1 leaf just became internal node) + 2(New leafs) . So we proved 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.

3 votes

-- Akash (41.5k points)

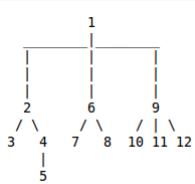
3.4.11 Binary Tree: GATE1994_8 [top](#)

<http://gateoverflow.in/250>



Selected Answer

Answer: The tree is a ternary tree.



4 votes

-- Rajarshi Sarkar (34.4k points)

3.4.12 Binary Tree: GATE1995_1.17 [top](#)

<http://gateoverflow.in/260>



Selected Answer

In Binary Tree a node can have atmost 2 child

Total number of node =>n = node with 0 child + node with 1 child + node with 2 child

$$n = n_0 + n_1 + n_2$$

Total number of edges

$$e = n - 1$$

$$\text{and also } e = n_0 * 0 + n_1 * 1 + n_2 * 2$$

Therefore

$$n - 1 = n_0 * 0 + n_1 * 1 + n_2 * 2$$

$$n_0 + n_1 + n_2 - 1 = n_1 * 1 + n_2 * 2$$

$n_2 = n_0 - 1$ (here leaf node is given n)
 therefore
n₂ = n - 1
option b

14 votes

-- Umang Raman (14.5k points)

3.4.13 Binary Tree: GATE1995_6 [top](#)

<http://gateoverflow.in/2667>



Selected Answer

There are only Five such binary trees.

One with C as root and left child as A and right child B.

Second with C as root, B as left child and A as again left child of B.

Third with C as root, B as left child and A as right child of B.

Fourth with C as root, B as right child and A as right child of B.

Fifth with C as root, B as right child and A as left child of B.

13 votes

-- Gate Keeda (18.8k points)

3.4.14 Binary Tree: GATE1996_1.14 [top](#)

<http://gateoverflow.in/2718>



Selected Answer

B.

a,b,c will become unbalanced with Balance factor as +2,+2,+2 respectively. Balance factor should be -1,0,+1.

Balance factor = Height(LST) - Height(RST).

8 votes

-- Gate Keeda (18.8k points)

3.4.15 Binary Tree: GATE1996_1.15 [top](#)

<http://gateoverflow.in/2719>



Selected Answer

C.

Left-->Right-->Root.

Ref: http://gateoverflow.in/2718/gate1996_1-14

8 votes

-- Gate Keeda (18.8k points)

3.4.16 Binary Tree: GATE1997-4.5 [top](#)

<http://gateoverflow.in/2246>



Selected Answer

Note:-->PreOrder means Root-Left-Right Traversal of tree.

By Option Elimination:-

B. False .Bcz. 5 is root so in preorder sequence first 5 elements must contain 1 to 5 But 6 comes here. So false.

So Now common things in option A,C,D is 5,3 means 5 root then LHS subtree root is 3 . now 3 s LHS is 1,2 so they should come first rather than 4. So option C is False.

Now Check for Option A,D

Root 5's RHS is 6,7,8 now as per Option A,D 7 is Root so 6 sud be Left and 8 should be Right so pre order for Right sub tree is 7,6,8(Root-L-R). Which satisfies option D.

17 votes

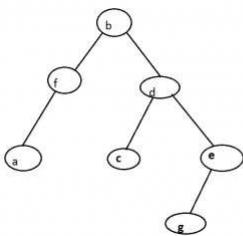
-- Rajesh Pradhan (16.9k points)

3.4.17 Binary Tree: GATE1998_20 [top](#)

<http://gateoverflow.in/1734>



Selected Answer



binary tree

18 votes

-- Anu (10.3k points)

3.4.18 Binary Tree: GATE2000-1.14 [top](#)

<http://gateoverflow.in/637>



Selected Answer

(A) -> (4 5 6 7) this part of answer is not correct. We have (X Y Z) not (W X Y Z). SO this is wrong

(B) -> 3 closing parenthesis, 2 opening parenthesis. This is wrong.

(D) -> Here in (1 (2 3 NULL) (4 5)) , (4 5) this is not allowed. So this is wrong. (It should be (4,5,NULL))

(C) Correct ->

11 votes

-- Akash (41.5k points)

3.4.19 Binary Tree: GATE2000-2.16 [top](#)

<http://gateoverflow.in/663>



Selected Answer

The answer is D.

Take any random sequence and check for the inorder, postorder and preorder Last Node.

12 votes

-- Gate Keeda (18.8k points)

Inorder : Left -> Root -> Right

Preorder : Root -> Left -> Right

Postorder: Left -> Right -> Root

If the binary tree is full (last level is fully filled), the last visited node in Inorder and Preorder must be the rightmost one in the last level. But for a complete binary tree this need not be the case (in a complete binary tree last level need not be fully filled) and LASTPRE will be from the second last level in case the complete binary tree is not full. So, choice (D).

10 votes

-- Arjun Suresh (287k points)

3.4.20 Binary Tree: GATE2002-2.12 [top](#)

<http://gateoverflow.in/842>



Selected Answer

Total number of nodes can be described by the recurrence

$$T(n) = T((n-1)/3) + T(2(n-1)/3) + 1$$

$$T(1) = 1$$

As this makes maximum nodes go to one subtree and that is what we want to get the maximum height with a given number of nodes.

Now, the height of the tree will be

$$H(n) = H(2/3(n-1)) + 1$$

$$H(1) = 0$$

We can draw a recurrence tree and the cost at each level is 1, and the height will be $\log_{(3/2)}n$.
So, D option is the answer.

1 21 votes

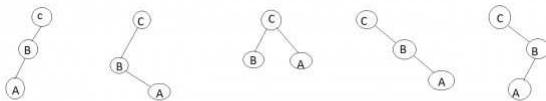
-- Arjun Suresh (287k points)

3.4.21 Binary Tree: GATE2002-6 [top](#)

<http://gateoverflow.in/859>



Selected Answer



5 Binary trees

1 7 votes

-- Anu (10.3k points)

3.4.22 Binary Tree: GATE2004-35 [top](#)

<http://gateoverflow.in/1032>



Selected Answer

Following combination can uniquely identify a tree.

Inorder and Preorder.

Inorder and Postorder.

Inorder and Level-order.

And following do not.

Postorder and Preorder.

Preorder and Level-order.

Postorder and Level-order.

Answer: B

4 votes

-- Shikhar Vashishth (5.2k points)

3.4.23 Binary Tree: GATE2004-43 [top](#)<http://gateoverflow.in/1040>

Selected Answer

It calculates Height of tree. (D)

Easy way to get this answer .

Draw a tree where all 4 parameters are different.

Get a Tree for which Height, No of Internal Nodes & No of Leafs are different & Trace out this algorithm.

8 votes

-- Akash (41.5k points)

3.4.24 Binary Tree: GATE2004-IT-54 [top](#)<http://gateoverflow.in/369>

Selected Answer

inorder traversal is left node right

preorder is node left right

answer: D

9 votes

-- Sankaranarayanan P.N (11k points)

3.4.25 Binary Tree: GATE2005-33 [top](#)<http://gateoverflow.in/1369>

Selected Answer

in order traversal of b binary search tree returns the element in sorted order - ascending (inorder is left parent then right. in a bst left is less than parent and right is greater than parent). In this option 1 is the only sorted list. hence it is the only possibility

13 votes

-- Sankaranarayanan P.N (11k points)

3.4.26 Binary Tree: GATE2005-IT-50 [top](#)<http://gateoverflow.in/3811>

Selected Answer

it should be B)

Since the difference between the nodes in left and right subtree must hold for every node, until the last to last to last level, all levels must be fully filled. So, we get $2^{h-1} - 1$ nodes (No. of nodes in a complete binary tree of height $h - 2$). Now, our aim is to increase two more levels by adding minimum no. of nodes- just add two in nodes one below other to any of the nodes. So, we get $2^{h-1} + 1$ nodes - B option.

13 votes

-- Sneha Goel (1.2k points)

3.4.27 Binary Tree: GATE2006-13 [top](#)<http://gateoverflow.in/974>

Selected Answer

should be D...

Since binary tree can be of any form, the worst case happens for right skewed binary tree. Now, root goes to index 1, its child goes to index 3, its child goes to index 7 and so on the nth vertex goes to $2^n - 1$ th index of array.

13 votes

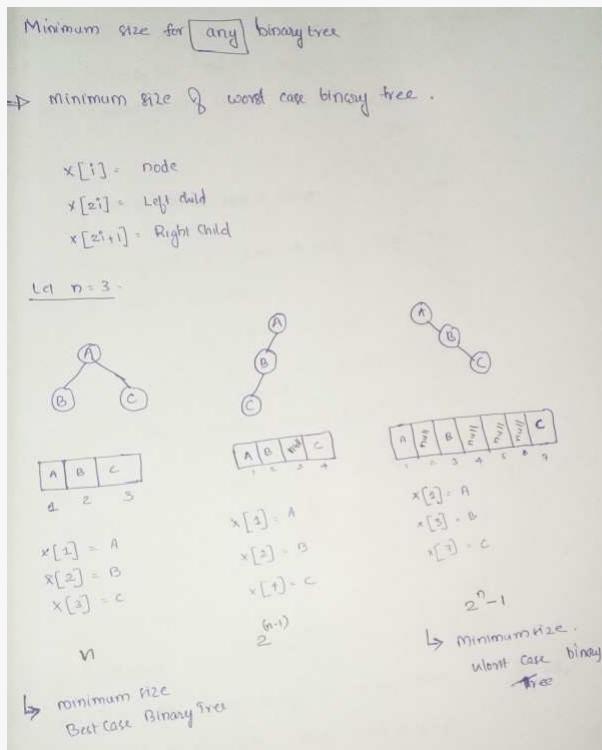
-- Shaun Patel (6.9k points)

Option D

To be able to store " **any** " binary tree on n vertices the minimum size of X should be

" Any Binary Tree and Size should be minimum " .

So We must consider worst case binary tree for this situation and find the minimum space required .



10 votes

-- pC (20.7k points)

3.4.28 Binary Tree: GATE2006-IT-71 [top](#)

<http://gateoverflow.in/3615>

option D

left child of i th element will be at $2i+1$ and right child at $2(i+1)$

11 votes

-- Sankaranarayanan P.N (11k points)

3.4.29 Binary Tree: GATE2006-IT-73 [top](#)

<http://gateoverflow.in/3617>

Floor($\log(i+1)$) draw the tree and realise that the last element at each level is the best choice to arrive at a conclusion

12 votes

-- Bhagirathi Nayak (13.1k points)

3.4.30 Binary Tree: GATE2006-IT-9 [top](#)

<http://gateoverflow.in/3548>

In a binary Tree,

no of nodes of degree 2 = no of leaves - 1.

No of nodes of degree 1 do not affect no of leaves !

No of leafs = No of nodes of degree 2 + 1 = 10 + 1 = 11

18 votes

-- Akash (41.5k points)

A node in a binary tree has degree 0, 1 or 2.

Ref: <http://faculty.cs.niu.edu/~mcmahon/CS241/Notes/bintree.html>

We are given no. of 1 degree node = 5, no. of 2 degree nodes = 10.

Total no. of edges = $1*5 + 2*10 = 25$ (In tree degree is for outgoing edges only, and hence each degree corresponds to an edge)

So, total no. of nodes = $25 + 1 = 26$ (No. of nodes in a tree is 1 less than no. of edges).

Now, no. of leaf nodes (nodes with 0 degree) = $26 - 5 - 10 = 11$.

22 votes

-- Arjun Suresh (287k points)

3.4.31 Binary Tree: GATE2007-12 [top](#)

<http://gateoverflow.in/1210>



Selected Answer

$2^{(h+1)} - 1$ just try this taking a small complete binary

never try to remember these formulae as remembering formulae is an overhead try to take examples in such cases

8 votes

-- Bhagirathi Nayak (13.1k points)

3.4.32 Binary Tree: GATE2007-13 [top](#)

<http://gateoverflow.in/1211>



Selected Answer

can be found with formula... $(2nCn/n+1)$... n being the number of nodes. for the given question... where n=3... answer is 5. Let me also specify here.. that number of Binary Search Trees with n nodes is equal to number of unlabeled Binary trees.

http://gatcse.in/wiki/Number_of_Binary_trees_possible_with_n_nodes

10 votes

-- Gate Keeda (18.8k points)

3.4.33 Binary Tree: GATE2007-39, UGCNET-June2015-II-22 [top](#)

<http://gateoverflow.in/1237>



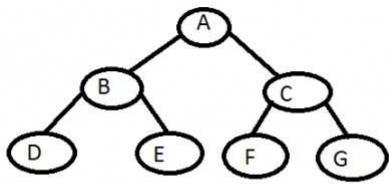
Selected Answer

The answer is A.

Take the first node in preorder traversal - 'a' will be the root of the tree

All nodes to the left of 'a' in inorder traversal will be in the left subtree of 'a' and all elements on the right will be in the right subtree of 'a'.

Take the second element from preorder traversal - 'b' - goes to left subtree of 'a' as it is in the left of 'a' in inorder list. Proceeding likewise we can construct the binary tree as:



4 votes

-- Gate Keeda (18.8k points)

3.4.34 Binary Tree: GATE2007-46 [top](#)

<http://gateoverflow.in/1244>



Answer: C

As the function returns 1 if and only if any node has both left & right children as NULL (that node is a leaf node). Hence, value gets incremented at each leaf node.

8 votes

-- Rajarshi Sarkar (34.4k points)

3.4.35 Binary Tree: GATE2008-IT-46 [top](#)

<http://gateoverflow.in/3356>



In preorder, root comes at the beginning of the traversal sequence and in postorder, root comes at the last of the traversal sequence. So, out of the given sequences only 1 and 2 are having such kind of order i.e K at the beginning and at the last.

Therefore, 2 is the preorder and 1 is postorder and the left sequence i.e 3 will definitely be inorder.

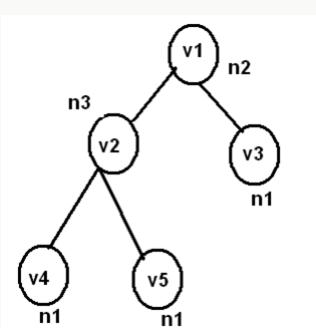
So, option d is correct.

12 votes

-- Vivek sharma (2.2k points)

3.4.36 Binary Tree: GATE2008-IT-76 [top](#)

<http://gateoverflow.in/3390>



assume the above tree so value of n1 is 3 n2 = 1 n3 = 1 check with options now u vil get option B as correct

15 votes

-- Shreyans Dhankhar (2.5k points)

3.4.37 Binary Tree: GATE2008-IT-77 [top](#)

<http://gateoverflow.in/3391>

Selected Answer

```

graph TD
    v1((v1)) --- v2((v2))
    v1 --- v3((v3))
    v2 --- v4((v4))
    v2 --- v5((v5))
    v3 --- n1((n1))
    style v1 fill:#ccc,stroke:#000
    style v2 fill:#ccc,stroke:#000
    style v3 fill:#ccc,stroke:#000
    style v4 fill:#fff,stroke:#000
    style v5 fill:#fff,stroke:#000
    style n1 fill:#fff,stroke:#000
    
```

from above tree we will get the tree below

```

graph TD
    v2((v2)) --- v4((v4))
    v2 --- v3((v3))
    v3 --- v5((v5))
    
```

now check with the options u vil get a as answer

13 votes

-- Shreyans Dhankhar (2.5k points)

3.4.38 Binary Tree: GATE2010-10 [top](#)

<http://gateoverflow.in/2183>

Selected Answer

0 because every node has an odd number of descendants so least odd number 1 and every node is considered to be its own descendant so all nodes have even number of descendants(0,2,4,6...) so every node has either 0 children or 2 children...

9 votes

-- Murali (419 points)

3.4.39 Binary Tree: GATE2012-47 [top](#)

<http://gateoverflow.in/2163>

Selected Answer

answer = **option A**

From the diagram below we are able to see how this works :

```

int height (Treeptr n)
{
    if (n == NULL) return -1;
    if (n->left == NULL)
        if (n->right == NULL) return 0;
    else return [n];
    else if (n->right == NULL)
        if (n->left == NULL) return [n];
    else if (n->right == NULL) return [n];
    else
        return [n];
}
    
```

The appropriate expressions for the two boxes I and II are:

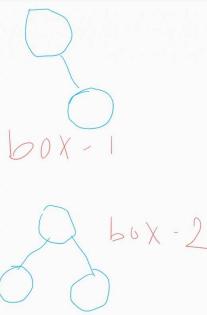
(A) B1: (1+height(left)) B2: (1+height(right))	(B) B1: (height(left)+right)) B2: (1+max(left,right))
(C) B1: height(left) B2: max(left,right)	(D) B1: (1+height(left)) B2: max(left,right)

9 votes

-- Amar Vashishth (27.9k points)

I got a hint ; How to quickly approach this type of question :--In this type of questions where we already know we want to find something(i.e. Height) then simply draw min possible tree which reach us upto the Given Boxes.

For Box1,Box2 tree will be as given in below img :-->



Now Analyse the code for above trees we get Option A as Ans.

12 votes

-- Rajesh Pradhan (16.9k points)

3.4.40 Binary Tree: GATE2014-1-12 [top](#)



Selected Answer

ans: 1..

Explanation:

- (1) Come to the 4th level up from the leaf node of the given binary tree, which can be done using tree traversal in $O(n)$.
- (2) For each node present in the level check whether it's subtree having exactly 4 nodes.. which can be done in constant time for each node, since it's subtree having constant number of nodes..
- (3) nodes in the level is less than n.. so its complexity is $O(n)$

therefore, $a = 1$ and $b = 0$

$a + 10b = 1$... <-Answer

16 votes

-- Vicky Bajoria (4.7k points)

We need to traverse all nodes at least once, and we need only one traversal. If $\text{num}(\text{child1}) + \text{num}(\text{child2}) + 1 = 4$, then output yes.

So, a must be 1 and b = 0, $a + 10b = 1$.

14 votes

-- Arjun Suresh (287k points)

3.4.41 Binary Tree: GATE2015-1_25 [top](#)



Selected Answer

option A is correct because height 5 means level 6 so maximum node = $2^6 - 1 = 63$
and for minimum, at each level only single node so total 6

20 votes

-- Anoop Sonkar (4.8k points)

3.4.42 Binary Tree: GATE2015-2_10 [top](#)

<http://gateoverflow.in/8059>



19

In Binary tree If there are N leaf nodes then the number of Nodes having two children will be N-1. So in this case answer will be 20-1, means 19.

12 votes

-- Raghuveer Dhakad (1.1k points)

3.4.43 Binary Tree: GATE2015-3_25 [top](#)



Selected Answer

Let number of nodes with exactly two children be x, and with exactly one children be y.

Total degree = $200 + 3x + 2y - 1$ (As all nodes with 2 children have degree 3 except the root)

No. of nodes = $x + y + 200$

No. of edges = Total degree/2 = $(200 + 3x + 2y - 1)/2$

No. of edges in a tree = No. of nodes - 1

So, $(200 + 3x + 2y - 1) = 2x + 2y + 400 - 2$

$x = 199$

25 votes

-- Arjun Suresh (287k points)

3.4.44 Binary Tree: TIFR2013-B-13 [top](#)



Selected Answer

The correct answer is option c, $\Theta(\sqrt{n})$.

$$n = 1 + 2 + 3 + \dots + (h + 1)$$

$$= \frac{(h+1)(h+2)}{2}$$

$$2n = h^2 + 3n + 2$$

$$0 = h^2 + 3n + (2 - 2n)$$

$$\Rightarrow h = \frac{-3 + \sqrt{3^2 - 4 \cdot (2 - 2n)}}{2} \quad \because h \geq 0$$

$$= \frac{-3 + \sqrt{8n + 1}}{2}$$

$$\Rightarrow h = \Theta(\sqrt{n})$$

24 votes

-- Pragy Agarwal (19.3k points)

3.4.45 Binary Tree: TIFR2014-B-1 [top](#)

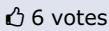


Selected Answer

We can construct binary tree by postorder and inorder traversal

There we get 5 leaves of the tree are a,c,e,h,j

So answer (C)5



6 votes

-- srestha (55.2k points)

3.4.46 Binary Tree: TIFR2015-B-4 [top](#)

<http://gateoverflow.in/29849>

Selected Answer

Option B

for every node -all numbers in one subtree are less than all numbers in the other subtree .

Firstly **choose** a value for **root**- 9 elements = **9 ways**

Now, we hv **8 elements left** - we hv to **choose 3 for left subtree & 5 for right subtree**.

Note: Here we can either chose 3 nodes from beginning or end out of 8 elements we have ! =**2 ways**

Now,we hv 3 elements for left subtree & 5 for right(**Consider subtrees of subtree**).

Left Subtree :

whatever way we place , always one side is smaller than other {6 is smaller than 8 in above example given in question} so, total ways =**3** ! {three places put one by one} = **6 ways**

Right Subtree :

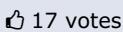
Right subtree has two more sub-trees ,so that elements on one side should be smaller than other**

Steps : 1) Select one element for root = **5 ways**

2) 4 elements left ,Select one element for left =**2 ways** {Either we can chose from left or right}

3) 3 elements left, for right subtree =**3!** ways = **6 ways**

Total ways = $9 \times 2 \times 3! \times 5 \times 2 \times 3! = 2^4 \times 3^2 \times 5 \times 9 = 6480 = \mathbf{B (Ans)}$



17 votes

-- Himanshu Agarwal (15.9k points)

3.5

Graphs(6) [top](#)

<http://gateoverflow.in/39731>

3.5.1 Graphs: GATE 2016-1-38 [top](#)

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 .

$$W = \begin{bmatrix} 0 & 2 & 8 & 5 \\ 2 & 0 & 5 & 8 \\ 8 & 5 & 0 & x \\ 5 & 8 & x & 0 \end{bmatrix}$$

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

Answer

3.5.2 Graphs: GATE1992_03,iii [top](#)

<http://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

Answer

3.5.3 Graphs: GATE1997_6.2 [top](#)

<http://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

Answer

3.5.4 Graphs: GATE2008-42 [top](#)

<http://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

Answer

3.5.5 Graphs: GATE2008-IT-4 [top](#)

<http://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

Answer

3.5.6 Graphs: GATE2014-1-3 [top](#)

<http://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 $E_3 = \{(u, v) \mid \text{there is a path of length } \leq 2 \text{ from } u \text{ to } v \text{ in } E\}$
- 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

Answer

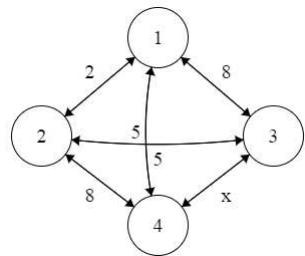
Answers: Graphs

3.5.1 Graphs: GATE 2016-1-38 [top](#)

<http://gateoverflow.in/39731>



Selected Answer



Let us list down the shortest edge between each pair of vertices x, y in the graph

x	y	shortest path from x to y
1	2	2
1	3	there is a direct path from 1 to 3 of weight 8. We can also chose to go via node 4 with total path weight $5+x$. If $5+x < 8$ ($x < 3$) then shortest path is $5+x$ otherwise shortest path is the weight of the direct path i.e 8
1	4	5
2	3	5
2	4	there is a direct path from 2 to 4 of weight 8. We can also chose to go via node 3 with total path weight $5+x$. If $5+x < 8$ ($x < 3$) then shortest path is $5+x$ otherwise shortest path is the weight of the direct path i.e 8
3	4	We can chose to go through the direct path of weight x or via node 2 with weight $5+8 = 13$. If $x < 13$ then we will choose x to be the shortest path otherwise 13 is the shortest path

- Case 1: Let us say $x < 3$. Say $x = 2$.

When we put $x = 2$ the above table is modified as

x	y	shortest path from x to y
1	2	2
1	3	7
1	4	5
2	3	5
2	4	7
3	4	2 // Note that the shortest path between node 3 and 4 is $x = 2$

- Case 2: $3 \leq x < 13$. Let's say $x = 12$. The table is now modified as

x	y	shortest path from x to y
1	2	2
1	3	8
1	4	5
2	3	5
2	4	8
3	4	12 // Note that the shortest path between node 3 and 4 is $x = 12$

Now the question asks you to find the largest possible integer value of x such that shortest path between atleast one pair of nodes in the graph is equal to x . for values $x = 2, 3, 4, \dots, 12$ the shortest path between node 3 and 4 is equal to x .

The largest among this is $x = 12$. So the answer is 12

21 votes

-- janakyMurthy (551 points)

Answer is $x=12$. Here, when we read the last sentence of the question, i.e the largest possible integer value of x . when $x=11$, shortest path is edge with weight x only. But when $x=12$, there are 2 shortest paths, and we can say that the edge with weight x is also a shortest path. Atleast 1 shortest path contain edge x . So answer will be 12.

24 votes

-- Sreyas S (1.8k points)

3.5.2 Graphs: GATE1992_03,iii [top](#)

<http://gateoverflow.in/580>



Selected Answer

Answer: $n-p$

Corollary: If G is a forest with n vertices and p components, then G has $n-p$ edges.

As, 0 edges for $p-1$ vertices ($p-1$ components) and $n-p$ edges for $n-p+1$ vertices (1 component). So, total of $n-p$ edges for p components.

3 votes

-- Rajarshi Sarkar (34.4k points)

3.5.3 Graphs: GATE1997_6.2 [top](#)

<http://gateoverflow.in/2258>



Selected Answer

From the description it is clear that vertices are connected as follows:

1-9-17-...-97
2-10-18-...-98
3-11-19-...-99
4-12-20-...-100
5-13-21-...-93
6-14-22-...-94
7-15-23-...-95
8-16-24-...-96

We have covered all vertices using 8 vertex sets considering only $|i - j| = 8$. Using $|i - j| = 12$ we can see the vertex 1 is connected to 13, 2-14, 3-15 and 4-16, so the top 4 vertex sets are in fact connected to the bottom 4 sets, thus reducing the connected components to 4.

12 votes

-- Arjun Suresh (287k points)

3.5.4 Graphs: GATE2008-42 [top](#)

<http://gateoverflow.in/1872>



Selected Answer

There are 2 spanning trees (a spanning tree connects all n vertices) for G which are edge disjoint. A spanning tree for n nodes require $n - 1$ edges and so 2 edge-disjoint spanning trees requires $2n - 2$ edges. As G has only $2n - 2$ edges, it is clear that it doesn't have any edge outside that of the two spanning trees. Now lets see the cases:

Lets take any subgraph of G with k vertices. The remaining subgraph will have $n - k$ vertices. Between these two subgraphs (provided both has at least one vertex) there should be at least 2 edges, as we have 2 spanning trees in G . So, (b) is TRUE. Also, in the subgraph with k vertices, we cannot have more than $2k - 2$ edges as this would mean that in the other subgraph with $n - k$ vertices, we would have less than $2n - 2k$ edges, making 2 spanning trees impossible in it. So, (a) is also TRUE.

A spanning tree covers all the vertices. So, 2 edge-disjoint spanning trees in G means, between every pair of vertices in G we have two edge-disjoint paths (length of paths may vary). So, (c) is also TRUE.

So, that leaves option (d) as answer. It is not quite hard to give a counter example for (d).

14 votes

-- Arjun Suresh (287k points)

3.5.5 Graphs: GATE2008-IT-4 [top](#)<http://gateoverflow.in/3264>

Selected Answer

Answer: C

1--2--3--4--5--6--7--8--9

(2,5,8) is the maximal independent set for a chain of 9 nodes. If we add any other node to the set then it will not be MIS.

 14 votes

-- Rajarshi Sarkar (34.4k points)

3.5.6 Graphs: GATE2014-1-3 [top](#)<http://gateoverflow.in/1754>

Selected Answer

A is false. Consider just two vertices connected to each other. So, we have one SCC. The new graph won't have any edges and so 2 SCC.

B is true. In a directed graph an SCC will have a path from each vertex to every other vertex. So, changing the direction of all the edges, won't change the SCC.

D is false. Consider any graph with isolated vertices- we loose those components.

C is a bit tricky. Any edge is a path of length 1. So, the new graph will have all the edges from old one. Also, we are adding new edges (u, v) . So, does this modify any SCC? No, because we add an edge (u, v) , only if there is already a path of length ≤ 2 from u to v - so we do not create a new path. So, both B and C must answer, though GATE key says only B. 21 votes

-- Arjun Suresh (287k points)

3.6**Hashing(17)** [top](#)**3.6.1 Hashing: GATE1989-1-vii, ISRO2015-14** [top](#)<http://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](#)

Answer

3.6.2 Hashing: GATE1996_1.13 [top](#)<http://gateoverflow.in/2711>

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](#)

[Answer](#)

3.6.3 Hashing: GATE1996_15 [top](#)

<http://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

$$h(x) = (\text{ord}(x) - \text{ord}("a")) + 1 \mod 10$$

and linear probing to resolve collisions.

- a. Which insertions cause collisions?
- b. Display the final hash table.

[gate1996](#) [data-structure](#) [hashing](#) [normal](#)

[Answer](#)

3.6.4 Hashing: GATE1997_12 [top](#)

<http://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](#)

[Answer](#)

3.6.5 Hashing: GATE2004-7 [top](#)

<http://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](#)

[Answer](#)

3.6.6 Hashing: GATE2005-IT-16 [top](#)

<http://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](#)

[Answer](#)

3.6.7 Hashing: GATE2006-IT-20 [top](#)

<http://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](#)

[Answer](#)

3.6.8 Hashing: GATE2007-40 [top](#)

<http://gateoverflow.in/1238>

Consider a hash table of size seven, with starting index zero, and a hash function $(3x + 4) \bmod 7$. 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](#)

[Answer](#)

3.6.9 Hashing: GATE2007-IT-28 [top](#)

<http://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](#)

[Answer](#)

3.6.10 Hashing: GATE2008-IT-48 [top](#)

<http://gateoverflow.in/3358>

Consider a hash table of size 11 that uses open addressing with linear probing. Let $h(k) = k \bmod 11$ 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](#)

[Answer](#)

3.6.11 Hashing: GATE2009-36 [top](#)

<http://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 $h(k) = k \bmod 10$ 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
5
3
6
23
7
5
8
18
9

D
0
1
2
2
12
3
13
4
5
5
15
6
7
8
18
9

[gate2009](#) [data-structure](#) [hashing](#) [normal](#)

[Answer](#)

3.6.12 Hashing: GATE2010-52 [top](#)

<http://gateoverflow.in/2360>

A hash table of length 10 uses open addressing with hash function

$h(k) = k \bmod 10$, 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](#) [difficult](#)
Answer

3.6.13 Hashing: GATE2010-53 [top](#)

<http://gateoverflow.in/43327>

A hash table of length 10 uses open addressing with hash function $h(k) = k \bmod 10$, 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?

- A. 10
- B. 20
- C. 30
- D. 40

[data-structure](#) [hashing](#) [difficult](#) [gate2010](#)
Answer

3.6.14 Hashing: GATE2014-1-40 [top](#)

<http://gateoverflow.in/1918>

Consider a hash table with 9 slots. The hash function is $h(k) = k \bmod 9$. 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](#)
Answer

3.6.15 Hashing: GATE2014-3-40 [top](#)

<http://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](#)
Answer

3.6.16 Hashing: GATE2015-2_33 [top](#)

<http://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. $h(i) = i^2 \bmod 10$
- B. $h(i) = i^3 \bmod 10$
- C. $h(i) = (11 * i^2) \bmod 10$
- D. $h(i) = (12 * i^2) \bmod 10$

[gate2015-2](#) [data-structure](#) [hashing](#) [normal](#)

[Answer](#)

3.6.17 Hashing: GATE2015-3_17 [top](#)

<http://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](#)

[Answer](#)

Answers: Hashing

3.6.1 Hashing: GATE1989-1-vii, ISRO2015-14 [top](#)

<http://gateoverflow.in/1005>



No of comparison in worst case for an element not in hash table is size of largest cluster +1. This is because the probe stops as soon as an empty slot is found (we r using linear probing here).

Size of largest cluster is 4 (s6, s3, s7, s1)

No of comparison is $4 + 1 = 5$

24 votes

-- Digvijay (46k points)

3.6.2 Hashing: GATE1996_1.13 [top](#)

<http://gateoverflow.in/2711>



A. False :- search operation can go worst in chaining if all elements are stored under a single bucket.

B. False . Pointer space is overhead in chaining.

C .is true BCZ in Open Addressing sometimes though element is present we cant delete it if Empty Bucket comes in between while searching for that element ;Such Limitation is not there in Chaining.

5 votes

-- Rajesh Pradhan (16.9k points)

3.6.3 Hashing: GATE1996_15 [top](#)

<http://gateoverflow.in/2767>



Here Order(x)-Order ('a') means count difference between that character & a.

Assuming a = 0, b = 1 & so on.

a) J & M cause collision.

b) Final Hash Table

Index	Key
0	T
1	K
2	J

Index	Key
4	N
5	Y
6	P
7	M
8	R
9	S

6 votes

-- Akash (41.5k points)

3.6.4 Hashing: GATE1997_12 [top](#)



Selected Answer

A) probability that other buckets are selected = $(n-1)/n$

this should happen k times and each of k events are independent so $(n-1)^k/n^k$

B) when k=1 prob of no collision = n/n

for k=2 prob of no collision = n/n * (n-1)/n

for k=k prob of no collision = n/n * (n-1)/n * (n-2)/n ... * (n-k+1)/n for k<=n

for k>n prob = 0

C) prob of collision at k=1 = $(k-1)/n$

prob of collision at k=2 = n/n * (k-1)/n

prob of collision at k=3 = n/n * (n-1)/n * (k-1)/n

prob of collision at k=k = n/n * (n-1)/n * (n-2)/n ... * 2/n * (k-1)/n for k<=n

for k>n prob = 1

7 votes

-- Danish (3.5k points)

3.6.5 Hashing: GATE2004-7 [top](#)



Selected Answer

option c is correct because the last digit of every digit given is equal in i and ii

7 votes

-- Bhagirathi Nayak (13.1k points)

3.6.6 Hashing: GATE2005-IT-16 [top](#)



Selected Answer

43 in loc 3

165 in loc 5

62 in loc 2

123 in loc 4 (collision and next free space)

142 in loc 6 (collision in 2, and 3,4,5 already occupied)

hence answer D

9 votes

-- Sankaranarayanan P.N (11k points)

3.6.7 Hashing: GATE2006-IT-20 [top](#)

<http://gateoverflow.in/3559>



Selected Answer

Answer is C)

I) A hash function takes a message of arbitrary length and generates a fixed length code.. This is correct, this is directly from definition of hash function. Ref -> https://en.wikipedia.org/wiki/Hash_function

II) As I is correct II is wrong !

III) This is true. example :-> Hash function N%10, this will generate same values for 1 as well as 11 !

(Even in cryptographic hash functions collision happens, just it is not easy to find colluding instances !)

10 votes

-- Akash (41.5k points)

3.6.8 Hashing: GATE2007-40 [top](#)

<http://gateoverflow.in/1238>



Selected Answer

The answer is B.

1 will occupy location 0, 3 will occupy location 6, 8 hashed to location 0 which is already occupied so, it will be hashed to one location next to it. i.e. to location 1.

Since 10 also clashes, so it will be hashed to location 2.

6 votes

-- Gate Keeda (18.8k points)

3.6.9 Hashing: GATE2007-IT-28 [top](#)

<http://gateoverflow.in/3461>



Selected Answer

The question is a bit ambiguous.

After hashing of how many keys, will the probability that any new key hashed collides with an existing one exceed 0.5.

Here, 'new key hashed' is the ambiguity. It can mean the probability of a collision in the next 'hash', or the probability of a collision in any of the hashes of the 'new keys' starting from the first insertion. For the first case answer must be 10 to get probability equal to 0.5, and so 11 must be the answer for probability > 0.5. Thus we can conclude from given choices, it is the second case.

So, we need to find n such that after n hashes, probability of collision (in any of the n hashes) > 0.5 .

Probability that there will be a collision after n hashes = 1 - Probability that there was no collision in the first n hashes

$$= 1 - 1 \cdot \frac{19}{20} \cdot \frac{18}{20} \cdots \frac{20-n+1}{20} .$$

So, we need,

$$0.5 < 1 - 1 \cdot \frac{19}{20} \cdot \frac{18}{20} \cdots \frac{20-n+1}{20} .$$

$$\Rightarrow \frac{19}{20} \cdot \frac{18}{20} \cdots \frac{20-n+1}{20} < 0.5 .$$

For $n = 5$, we get, 0.5814 and for $n = 6$, we get 0.43605. So, answer should be $n = 6$.

Ref: <http://www.cse.iitd.ernet.in/~bagchi/courses/discrete-book/ch5.pdf>

18 votes

-- Arjun Suresh (287k points)

3.6.10 Hashing: GATE2008-IT-48 [top](#)

<http://gateoverflow.in/3358>

 Selected Answer

Index	Key
0	87
1	11
2	13
3	36
4	92
5	4
6	71
7	14
8	
9	
10	43

D is answer

 6 votes -- Prashant Singh (47.6k points)

3.6.11 Hashing: GATE2009-36 top

<http://gateoverflow.in/1322>

 Selected Answer

C is the correct option ..directly from the definition of linear probing. In linear probing, when a hashed location is already filled, locations are linearly probed until a free one is found.

<http://courses.cs.washington.edu/courses/cse326/00wi/handouts/lecture16/sld015.htm>

 12 votes -- Bhagirathi Nayak (13.1k points)

3.6.12 Hashing: GATE2010-52 top

<http://gateoverflow.in/2360>

 Selected Answer

Option C

46, 34, 42, 23, 52, 33

46 - position 6

34 position 4

42 position 2

23 position 3

52 position 2 - collision next empty is 5

33 position 3- collision next empty is 7

 7 votes -- Sankaranarayanan P.N (11k points)

3.6.13 Hashing: GATE2010-53 [top](#)

<http://gateoverflow.in/43327>



Selected Answer
53 - option (C).

Slots 3,4,5 and 6 must be filled before 33 comes. Similarly slots 2,3 and 4 must be filled before 52 comes. And 52 must come before 33, as it is not occupying slot 2. So, 33 must be at the end and 52 can come at position 4 or 5.

Let 52 come at position 4. Slots 2, 3 and 4 must be filled before 52 leaving only slot 6 left for the element coming at position 5 which should be 46. So, the first 3 elements can come in any order giving $3! = 6$ ways.

Let 52 come at position 5. Now, the first four elements can come in any order. So, $4! = 24$ ways.

So, total number of different insertion sequences possible = $24 + 6 = 30$

23 votes

-- Arjun Suresh (287k points)

answer = **option C**

element

33 has managed to hold position at slot #7 it means elements should occupy slot #3 to slot #6 before it in the sequence. Currently it seems like all element except

42 should come before

33 in the sequence.

But, element 52 requires that 42 comes before it. and 33 requires that 52 comes before it. This means that 42 has to come before 33. this makes element 33 to occupy last position in the sequence.

now for element 52 to occupy its place these can be two cases :

case1 : {42, 23, 34} 52

case2: {42, 23, 34, 46}, 52

case 1 means that those three elements comes before

52 =

$3! = 6$ ways

case 2 means that those four elements comes before

52 =

$4! = 24$ ways

combining all info we get,

total number of sequences possible that will form the same hash table as above

$= (6 + 24) \times 1 = 30$

14 votes

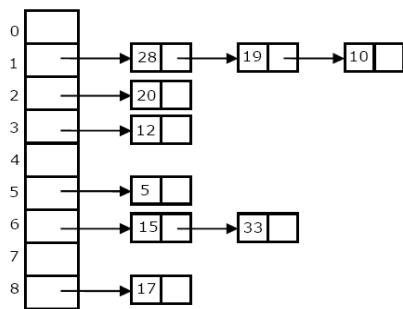
-- Amar Vashishth (27.9k points)

3.6.14 Hashing: GATE2014-1-40 [top](#)

<http://gateoverflow.in/1918>



Selected Answer



So, Maximum & minimum chain lengths are 3 & 0 respectively.

$$\text{Average chain length} = (0+3+1+1+0+1+2+0+1)/9 = 1.$$

So, Ans is A

16 votes

-- Jay (1.2k points)

3.6.15 Hashing: GATE2014-3-40 [top](#)

<http://gateoverflow.in/2074>



Selected Answer

We have 100 slots each of which are picked with equal probability by the hash function (since hashing is uniform). So, to avoid first 3 slots, the hash function has to pick from the remaining 97 slots. And repetition is allowed, since chaining is used- meaning a list of elements are stored in a slot and not a single element.

$$\begin{aligned} \text{So, required probability} &= \frac{97}{100} \times \frac{97}{100} \times \frac{97}{100} \\ &= (97 \times 97 \times 97)/100^3 \end{aligned}$$

23 votes

-- Arjun Suresh (287k points)

3.6.16 Hashing: GATE2015-2_33 [top](#)

<http://gateoverflow.in/8152>



Selected Answer

Since mod 10 is used, the last digit matters.

If you do cube all numbers from 0 to 9, you get following

Number	Cube	Last Digit in Cube
0	0	0
1	1	1
2	8	8
3	27	7
4	64	4
5	125	5
6	216	6
7	343	3
8	512	2
9	729	9

Therefore all numbers from 0 to 2020 are equally divided in 10 buckets. If we make a table for square, we don't get equal distribution. In the following table. 1, 4, 6 and 9 are repeated, so these buckets would have more entries and buckets 2, 3, 7 and 8 would be empty.

Number	Square	Last Digit in Cube
0	0	0
1	1	1
2	4	4
3	9	9
4	16	6
5	25	5
6	36	6
7	49	9
8	64	4
9	81	1

<http://geeksquiz.com/gate-gate-cs-2015-set-2-question-43/>

19 votes

-- Anu (10.3k points)

3.6.17 Hashing: GATE2015-3_17 [top](#)

<http://gateoverflow.in/8414>



Selected Answer

A critical statistic for a hash table is the **load factor**, that is the number of entries divided by the number of buckets:

$$\text{Load factor} = n/k$$

where:

$$\begin{aligned} n &= \text{number of entries} \\ k &= \text{number of buckets} \end{aligned}$$

As the load factor grows larger, the hash table becomes slower, and it may even fail to work (depending on the method used).

here Load factor = $2000/25 = 80$

7 votes

-- Akash (41.5k points)

load factor = total no element/total no of slots
 $\Rightarrow 2000/25 = 80$

13 votes

-- Anoop Sonkar (4.8k points)

3.7

Heap(24) [top](#)

3.7.1 Heap: GATE 2016-1-37 [top](#)

<http://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

Answer

3.7.2 Heap: GATE 2016-2-34 [top](#)

<http://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

Answer

3.7.3 Heap: GATE1996_2.11 [top](#)

<http://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

Answer

3.7.4 Heap: GATE1999_12 [top](#)

<http://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.
- 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

Answer

3.7.5 Heap: GATE2001-1.15 [top](#)

<http://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. $\left\lfloor \frac{i}{2} \right\rfloor$
- C. $\left\lceil \frac{i}{2} \right\rceil$
- D. $\frac{(i+1)}{2}$

gate2001 | data-structure | heap | easy

Answer

3.7.6 Heap: GATE2003-23 [top](#)

<http://gateoverflow.in/1110>

In a min-heap with n elements with the smallest element at the root, the j^{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

Answer

3.7.7 Heap: GATE2004-37 [top](#)

<http://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.
-
- ```

graph TD
 32((32)) --- 30((30))
 32 --- 25((25))
 30 --- 15((15))
 30 --- 12((12))
 25 --- 20((20))
 25 --- 16((16))

```
- B.
- 
- ```

graph TD
    32((32)) --- 25((25))
    32 --- 30((30))
    25 --- 12((12))
    25 --- 15((15))
    30 --- 20((20))
    30 --- 16((16))
  
```
- C.
-
- ```

graph TD
 32((32)) --- 30((30))
 32 --- 25((25))
 30 --- 15((15))
 30 --- 12((12))
 25 --- 16((16))
 25 --- 20((20))

```
- D.
- 
- ```

graph TD
    32((32)) --- 25((25))
    32 --- 30((30))
    25 --- 12((12))
    25 --- 15((15))
    30 --- 16((16))
    30 --- 20((20))
  
```

gate2004 data-structure heap normal

[Answer](#)

3.7.8 Heap: GATE2004-IT-53 [top](#)

<http://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 $\lfloor(n - 1) / 2\rfloor, \lfloor(n - 3) / 2\rfloor, \dots, 0$. 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. $O(n \log n)$

gate2004-it data-structure heap normal

[Answer](#)

3.7.9 Heap: GATE2005-34 [top](#)

<http://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

[Answer](#)

3.7.10 Heap: GATE2006-10 [top](#)

<http://gateoverflow.in/889>

In a binary max heap containing n numbers, the smallest element can be found in time

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

gate2006 data-structure heap easy

[Answer](#)

3.7.11 Heap: GATE2006-76 [top](#)

<http://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

[Answer](#)

3.7.12 Heap: GATE2006-77 [top](#)

<http://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

[Answer](#)

3.7.13 Heap: GATE2006-IT-44 [top](#)

<http://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](#)

[Answer](#)

3.7.14 Heap: GATE2006-IT-72 [top](#)

<http://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](#)

[Answer](#)

3.7.15 Heap: GATE2007-47 [top](#)

<http://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](#)

[Answer](#)

3.7.16 Heap: GATE2009-59 [top](#)

<http://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](#)

[Answer](#)

3.7.17 Heap: GATE2009-60 [top](#)

<http://gateoverflow.in/4346>

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}

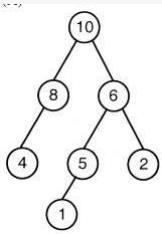
gate2009 data-structure heap normal

Answer

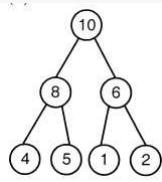
3.7.18 Heap: GATE2011_23 [top](#)<http://gateoverflow.in/2125>

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?

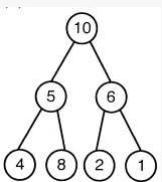
(A)



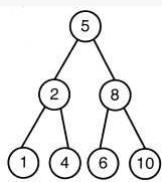
(B)



(C)



(D)



gate2011 data-structure heap easy

Answer

3.7.19 Heap: GATE2014-2-12 [top](#)<http://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

Answer**3.7.20 Heap: GATE2015-1_32** [top](#)<http://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](#)**Answer****3.7.21 Heap: GATE2015-2_17** [top](#)<http://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

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

[gate2015-2](#) [data-structure](#) [heap](#) [normal](#)**Answer****3.7.22 Heap: GATE2015-3_19** [top](#)<http://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

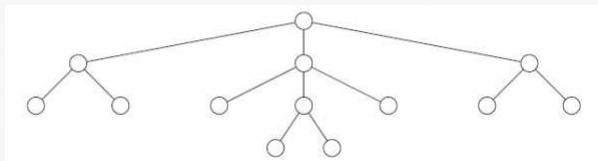
- A. 4
- B. 5
- C. 2
- D. 3

[gate2015-3](#) [data-structure](#) [heap](#) [normal](#)**Answer****3.7.23 Heap: ISI2014-CS-2b** [top](#)<http://gateoverflow.in/47438>

Let H_1 and H_2 be two complete binary trees that are heaps as well. Assume H_1 and H_2 are max-heaps, each of size n . Design and analyze an efficient algorithm to merge H_1 and H_2 to a new max-heap H of size $2n$.

[descriptive](#) [isi2014](#) [algorithms](#) [binary-tree](#) [heap](#)**Answer****3.7.24 Heap: TIFR2014-B-19** [top](#)<http://gateoverflow.in/27352>

Consider the following tree with 13 nodes.



Suppose the nodes of the tree are randomly assigned distinct labels from $\{1, 2, \dots, 13\}$, 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](#)

[Answer](#)

Answers: Heap

3.7.1 Heap: GATE 2016-1-37 [top](#)

<http://gateoverflow.in/39705>



Selected Answer

Answer would be B) O(d) but not O(1).. as we need to apply heapify.. and suppose if we are deleting root, in worst case would take O(d) time..

20 votes

-- Abhilash Panicker (8.7k points)

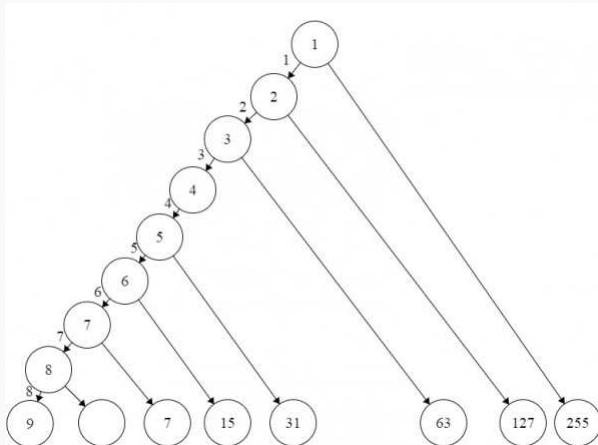
3.7.2 Heap: GATE 2016-2-34 [top](#)

<http://gateoverflow.in/39585>



Selected Answer

Here answer is 8. With 1024 nodes, we can easily build min heap Check following diagram



Now once we place 1-9 then remaining elements can be placed easily to fill up heap (While keeping heap property of course) Total elements we need for this heap is 512, we have given 1024 ! So Yes, 8 is answer !

24 votes

-- Akash (41.5k points)

3.7.3 Heap: GATE1996_2.11 [top](#)

<http://gateoverflow.in/2740>



Selected Answer

"The minimum number of interchanges needed to convert the array
 89, 19, 40, 17, 12, 10, 2, 5, 7, 11, 6, 9, 70
 into a heap with the maximum element at the root node is:"

This is the correction.

Answer: C.

Only element 70 violates the rule. Hence, it must be shifted to its proper position.

Step1: swap(10, 70)

Step2: swap(40, 70)

Hence, only 2 interchanges are required.

10 votes

-- Gate Keeda (18.8k points)

3.7.4 Heap: GATE1999_12 [top](#)

<http://gateoverflow.in/1511>



Selected Answer

(A)

Note My Convention:-

no. Of full node=F

no. Of leaf node=L

 Base Case: H = 0.

A binary tree of height 0 is just a single node with no children, and therefore has 1 leaf.

$$F+1 = L$$

$$0+1=1$$

so the base case satisfies the induction hypothesis (see below).

Induction Hypothesis(I.H): Suppose that for some $k \geq 0$, all binary trees of height $\leq k$ have $(F+1) = L$ leaves .

Induction Step: Let T be a binary tree of height $k+1$. Then T's left and right subtrees are each binary trees of height $\leq k$, and thus by the I.H. both subtree have $(F+1)$ leaves. The number of leaves in T is equal to the sum of the number of leaves in T's subtrees,

$$(F+1)_{\text{left sub tree}} + (F+1)_{\text{right sub tree}} = L_{\text{left sub tree}} + L_{\text{right sub tree}}$$

$$2F+2=2L$$

$$2(F+1)=2(L)$$

$$\therefore F+1=L \text{ (proved)}$$

Hence the hypothesis holds for $k+1$, and so the theorem is proved.

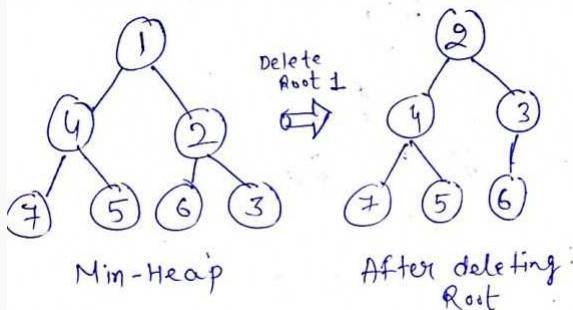
(B)

Here in question they mentioned to insert element in given order into an empty Heap.

So here we have to use Insertion Method to create the heap instead of using Heapify Method to build the heap.

Plz refer below img the LHS shows the resultant heap after doing insertion of the keys into initial empty heap.

RHS heap is the result of deletion of root.

 4 votes

-- Rajesh Pradhan (16.9k points)

3.7.5 Heap: GATE2001-1.15 top

<http://gateoverflow.in/708>

Selected Answer

for node at index i

left child(L) at 2i

right child(R) at 2i+1

for node at index i

parent will be at floor i/2

 11 votes

-- Pooja Palod (31.2k points)

3.7.6 Heap: GATE2003-23 top

<http://gateoverflow.in/1110>

Selected Answer

Time to find the smallest element on a min-heap- one retrieve operation - $\Theta(1)$ Time to find the second smallest element on a min-heap- requires $2^2 - 1 = 3$ check operations to find the second smallest element out of 3 elements - $\Theta(1)$ Time to find the 7th smallest element - requires $O(2^7 - 1) = O(127)$ check operations to find the seventh smallest element out of 127 possible ones - $\Theta(1)$ In short if the number of required operations is independent of the input size n, then it is always $\Theta(1)$.

(Here, we are doing a level order traversal of the heap and checking the elements)

If we are not allowed to traverse the heap and allowed only default heap-operations, we will be restricted with doing Extract-min 7 times which would be $O(\log n)$. 30 votes

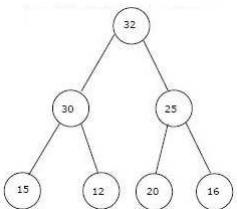
-- gatecse (13.1k points)

3.7.7 Heap: GATE2004-37 top

<http://gateoverflow.in/1034>

Selected Answer

answer = **option A**



Just keep inserting elements making sure resulting Tree is nearly Complete.(Heap Property) .

While inserting any node, if you find that Value of New Node > Value of it's parent, bubble it up to keep Max heap property

8 votes

-- Akash (41.5k points)

3.7.8 Heap: GATE2004-IT-53 [top](#)

<http://gateoverflow.in/3696>



Selected Answer

By using **Build Heap method** we can create heap from complete binary tree which will take O(n).

15 votes

-- Sneha Goel (1.2k points)

3.7.9 Heap: GATE2005-34 [top](#)

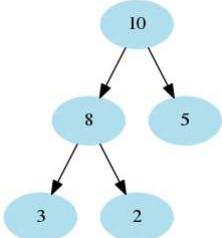
<http://gateoverflow.in/1370>



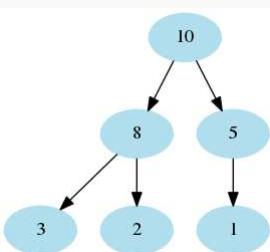
Selected Answer

ans is D....whenever we insert an element in heap,it will always inserted in last level from left to right..so here we insert element 1 and 7 as a child of node 5.then we perform heapify algorithm until we get the min/max heap..so here finally in above question we get the heap whose level order traversal is 10,8,7,3,2,1,5

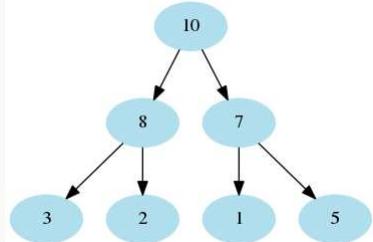
Initial heap:



After insert of 1



After insert of 7



10 votes

-- neha pawar (4.4k points)

3.7.10 Heap: GATE2006-10 top

<http://gateoverflow.in/889>



Selected Answer

(A) $O(n)$

In a max heap, the smallest element is always present at a leaf node. Heap being a complete binary tree, there can be up to $\frac{n}{2}$ leaf nodes and to examine all of them we would need $O(n)$ time.

11 votes

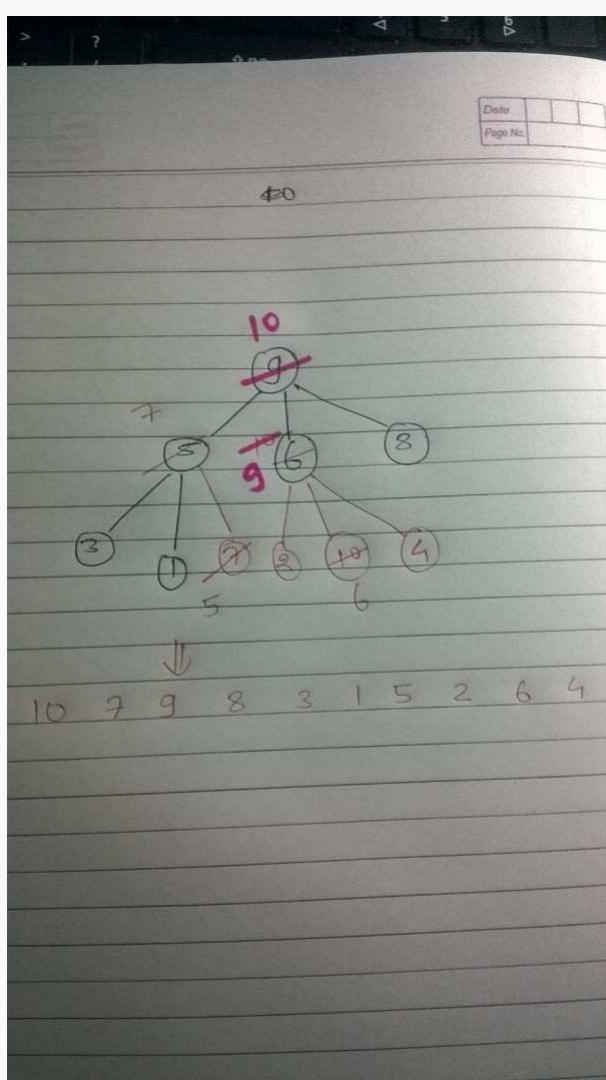
-- Keith Kr (6.2k points)

3.7.11 Heap: GATE2006-76 top

<http://gateoverflow.in/1852>



Selected Answer



For 77, heap will be constructed like this, based on the correct answer of 76(which is option D)

8 votes

-- Anurag Pandey (12.8k points)

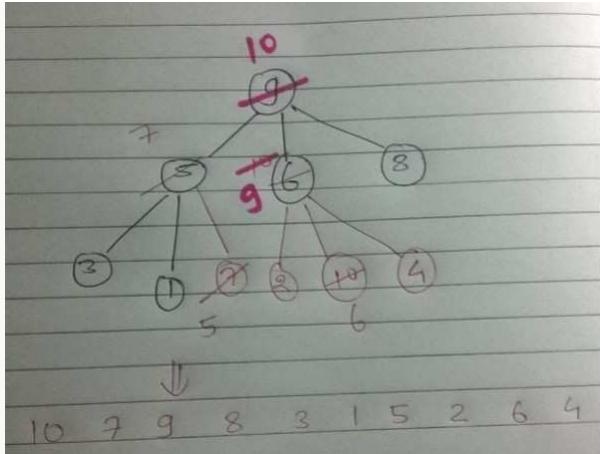
3.7.12 Heap: GATE2006-77 top

<http://gateoverflow.in/87191>



Selected Answer

Ans to ques 76 is 9, 5, 6, 8, 3, 1
and ans for ques 77 given by [Anurag Pandeya](#)



2 votes

-- Lokesh . (9.5k points)

3.7.13 Heap: GATE2006-IT-44 [top](#)

<http://gateoverflow.in/3587>



for a heap(max heap) parent should be greater than or equal to children. in a heap of [1..n] left child of ith node will be at $2*i$ th position and right child will be at $2*i+1$ position

so for given options we can verify it

option C seems to be following the property

9 votes

-- Sankaranarayanan P.N (11k points)

3.7.14 Heap: GATE2006-IT-72 [top](#)

<http://gateoverflow.in/3616>



The question is saying best case which will be when only one swap will be required which will be order of 1. As no option matches just call heapify at the root - $O(\log n)$.

9 votes

-- No Need (13.7k points)

3.7.15 Heap: GATE2007-47 [top](#)

<http://gateoverflow.in/1245>



number of elements in the path from new leaf to root = $\log n$, and all elements are sorted from leaf to root so we can do a binary search which will result in $O(\log \log n)$ number of comparisons.

Since in heap is a complete binary tree, in an array implementation, from every node index, we can know its depth and this will be the n for binary search.

23 votes

-- Vikrant Singh (13.4k points)

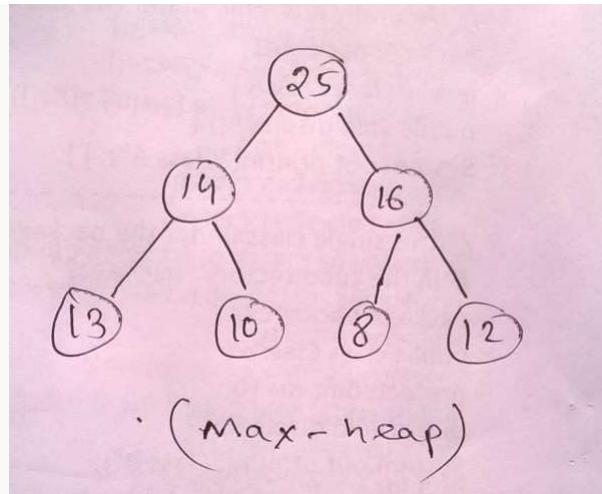
3.7.16 Heap: GATE2009-59 [top](#)

<http://gateoverflow.in/1341>



Answer : (C)

The binary max-Heap looks like this :

 2 votes

-- Dipak Majhi (685 points)

Taking the given array as level order traversal, we can build binary tree.

- (A) 13 comes as child of 12, which is not allowed in a binary max-heap
- (B) 16 comes as child of 14 violating max-heap property
- (C) is a valid binary max-heap as all children are smaller than their parent
- (D) 16 comes as child of 12, violating max-heap property

 11 votes

-- Arjun Suresh (287k points)

3.7.17 Heap: GATE2009-60 top

<http://gateoverflow.in/43466>

Selected Answer

During delete, the root element is removed, replaced with the last element and heap property is corrected by pushing the root downwards. So, for first delete,

25 14 16 13 10 8 12 -> 12 14 16 13 10 8 -> 16 14 12 13 10 8 (the element not satisfying max-heap property is exchanged with the largest of its children) (heap property satisfied)

Second delete:

16 14 12 13 10 8 -> 8 14 12 13 10 -> 14 8 12 13 10 -> 14 13 12 8 10 (heap property satisfied)

<http://homepages.ius.edu/RWISMAN/C455/html/notes/Chapter6/heapify.htm>

 9 votes

-- Arjun Suresh (287k points)

3.7.18 Heap: GATE2011_23 top

<http://gateoverflow.in/2125>

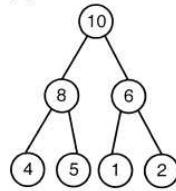
Selected Answer

in option a - it is not a max heap because it is not complete binary tree .

in option c-it is complete binary tree but not follow the max heap property i.e. the values of parent nodes always greater then child nodes

and there node of value 5 is less then one of its children.

in option d- similar to above c option explaination here node of value 2 is less then to the value 4 .



correct option is (B) that is satisfy both properties and all of the max heap .

10 votes

-- ASHU2015 (277 points)

3.7.19 Heap: GATE2014-2-12 [top](#)

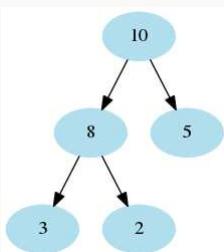
<http://gateoverflow.in/1967>



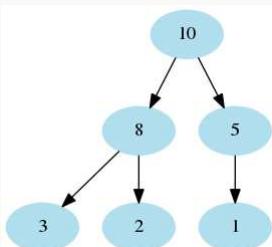
Selected Answer

ans is A....whenever insertion will be done in heap ,it will always inserted in last level from left to right.so we insert 1 and 7as a child of node 5 now we perform heapify algorithm until heap property will satisfied..and then we get the heap whose level order traversal is 10,8,7,3,2,1,5

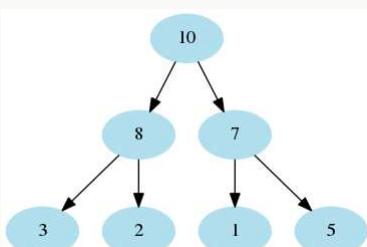
Initial heap



After insert of 1



After insert of 7



10 votes

-- neha pawar (4.4k points)

3.7.20 Heap: GATE2015-1_32 [top](#)

<http://gateoverflow.in/8273>



Selected Answer

Heap is complete binary tree. To insert a new element, we put it at the end of the tree and move up towards root till heap property is satisfied. Here, 35 comes as child of 15, with the path 40-30-15-35. So, we swap 15, 35 and then 30, 35 to get the new path 40-35-30-15. So, new heap will be 40 35 20 10 30 16 17 8 4 15.

18 votes

-- Arjun Suresh (287k points)

3.7.21 Heap: GATE2015-2_17 [top](#)



Selected Answer

Ans A.

Here, lower bound imply best algorithm which works for all cases and hence we should consider worst-case.

Max-Heapify(root)

22 votes

-- Vikrant Singh (13.4k points)

3.7.22 Heap: GATE2015-3_19 [top](#)



Selected Answer

first interchange 15-100
2nd 50-100
3rd 89-100
so total interchange 3 so option D is correct.

16 votes

-- Anoop Sonkar (4.8k points)

3.7.23 Heap: ISI2014-CS-2b [top](#)

First copy the two arrays of H1 and H2 into a new array of size 2n...then apply build heap operation to get H...Time complexity=O(2n)=O(n)

2 votes

-- debanjan sarkar (3.5k points)

3.7.24 Heap: TIFR2014-B-19 [top](#)



Selected Answer

(C) is Correct -

$$\text{Probability} = \frac{\text{Number of Favorable Outcomes(minheaps)}}{\text{Total Trees}(13!)} = \frac{1}{13!}$$

Now, total Min-Heaps :

Firstly we select **minimum element** i.e. 1 **for root = 1 way**

We have 3 subtrees of 3,6,3 sizes respectively

First Subtree with 3 elements:

Steps:

1) Choose 3 elements = **${}^{12}C_3$ ways**

2) Give minimum to root = **1 way**

3) 2 elements , 2 children(left & right) = **2 ways** { Any of the 2 elements can be given to left or right child}

Second Subtree with 3 elements :

Steps:

1) Choose 3 elements from 9 left elements = 9C_3 ways

2) Similar to above assign these = 2 ways

Subtree with 6 elements: { We already have 6 elements left now }

1 Root , 1left_most child , 1 right_most child , 3 in middle **

Steps:

1) Choose root = 1 way {minimum}

2) Now 5 elements left, choose leftmost child = 5 ways (choose anyone)

3) Now 4 elements left, choose right_most child = 4 ways (choose anyone)

4) Now 3 elements left for middle :

a) assign root = 1 way

b) assign left and right anyone = 2 ways

For, Total Heaps multiply all ways **above**.

Total probability =

$$\frac{\text{Total Heaps}}{\text{Total Trees}(13!)} =$$

$$\frac{12! * 2 * 9! * 2 * 5 * 4 * 2}{13! * 3! * 9! * 3! * 6!} =$$

$$\left(\frac{1}{13}\right) \left(\frac{1}{6}\right) \left(\frac{1}{3}\right)^3 =$$

17 votes

-- Himanshu Agarwal (15.9k points)

3.8

Linked Lists(19) top

3.8.1 Linked Lists: GATE 2016-2-15 top

<http://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](#)

Answer

3.8.2 Linked Lists: GATE1987-1-XV top

<http://gateoverflow.in/80298>

In a circular linked list organisation, insertion of a record involves modification of

- A. One pointer.
- B. Two pointers.
- C. Multiple pointers.
- D. No pointer.

[gate1987](#) [data-structure](#) [linked-lists](#)

Answer

3.8.3 Linked Lists: GATE1993_13 [top](#)

<http://gateoverflow.in/2310>

Consider a singly linked list having n nodes. The data items d_1, d_2, \dots, 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 $d_1, d_2, \dots, d_j, d, \dots, d_n$ in order without using the header.

gate1993 data-structure linked-lists normal

Answer

3.8.4 Linked Lists: GATE1994-1.17, UGCNET-Sep2013-II-32 [top](#)

<http://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

Answer

3.8.5 Linked Lists: GATE1995-2.22 [top](#)

<http://gateoverflow.in/2834>

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

Answer

3.8.6 Linked Lists: GATE1997-18 [top](#)

<http://gateoverflow.in/2278>

Consider the following piece of 'C' code fragment that removes duplicates from an ordered list of integers.

```
Node *removeDuplicates (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.

- How many times is the comparison in statement S1 made?
- What is the minimum and the maximum number of times statements marked S2 get executed?
- What is the significance of the value in the integer pointed to by j when the function completes?

gate1997 | data-structure | linked-lists | normal

[Answer](#)

3.8.7 Linked Lists: GATE1997_1.4 [top](#)

<http://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?

- Singly linked list
- Doubly linked list
- Circular doubly linked list
- Array implementation of list

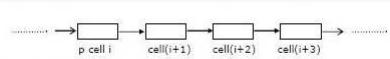
gate1997 | data-structure | linked-lists | easy

[Answer](#)

3.8.8 Linked Lists: GATE1998_19a [top](#)

<http://gateoverflow.in/1733>

- 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
q -> next := (q -> next) -> next
(p -> next) -> next := q
```

gate1998 | data-structure | linked-lists | normal

[Answer](#)

3.8.9 Linked Lists: GATE1999_11b [top](#)

<http://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

[Answer](#)

3.8.10 Linked Lists: GATE2002-1.5 [top](#)

<http://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

- $\log n$
- $\frac{n}{2}$
- $\log_2 n - 1$
- n

gate2002 | easy | data-structure | linked-lists

[Answer](#)

3.8.11 Linked Lists: GATE2003-90 [top](#)

<http://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

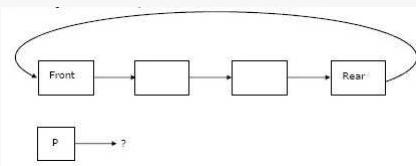
[gate2003](#) [data-structure](#) [linked-lists](#) [normal](#)

[Answer](#)

3.8.12 Linked Lists: GATE2004-36 [top](#)

<http://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](#)

[Answer](#)

3.8.13 Linked Lists: GATE2004-40 [top](#)

<http://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](#)

[Answer](#)

3.8.14 Linked Lists: GATE2004-IT-13 [top](#)

<http://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

Answer

3.8.15 Linked Lists: GATE2005-IT-54 [top](#)

<http://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

Answer

3.8.16 Linked Lists: GATE2008-62 [top](#)

<http://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

Answer

3.8.17 Linked Lists: GATE2010-36 [top](#)

<http://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;
    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->next = head; head = p;
- B. q->next = NULL; head = p; p->next = head;
- C. head = p; p->next =q; q->next = NULL;
- D. q->next = NULL; p->next = head; head = p;

gate2010 data-structure linked-lists normal

[Answer](#)

3.8.18 Linked Lists: GATE2017-1-08 [top](#)

<http://gateoverflow.in/116711>

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.

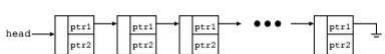
gate2017-1 data-structure linked-lists normal

[Answer](#)

3.8.19 Linked Lists: ISI2015-CS-3a [top](#)

<http://gateoverflow.in/47273>

Consider a linked list containing n nodes, where each node contains two pointers $ptr1$ and $ptr2$. For each node, $ptr1$ points to the next node of the list. Describe how pointer $ptr2$ should be set up for each node so that you will be able to locate the i -th node from the start node in the list traversing no more than $\lceil \log i \rceil + [i/2]$ nodes.



descriptive isi2015 data-structure linked-lists

[Answer](#)

Answers: Linked Lists

3.8.1 Linked Lists: GATE 2016-2-15 [top](#)

<http://gateoverflow.in/3957>



Tip Q means Theata !

Here I believe answer is C

Delete O(1)

Insert O(N)

Find O(N)

Decrease Key => O(N) //Because we need to search position in Linked list. (It is similar to a Delete followed by an Insert with the decreased value)

O(n) delete => O(N * 1) = O(N)

O(logN) find => O(logN * N) => O(NlogN)

O(logN) insert => O(NlogN)

O(N) decrease key => O(N*N) => O(N²)

Even though it says at start we got N elements, then we are deleting Q(N) elements, here Q(N) can be anything like N/2, N/4, N/3 and list need not be empty, then above explanation holds !

In case it actually deleted all elements at start analysis will be something like below =>

all N are deleted, Time complexity O(1) each delete , total delete O(N)

Now LogN insert, it'll take 1 + 2 + LogN time, then (LogN * logN-1)/2 => O((logN)²)

Now LogN finds => it'll take LogN time per find (because find take O(N) but here N = LogN
=> O((LogN)²)

Now N decrease key, Size of list is Log N, each decrease key can take O(size of list)

So size of list * no of decrease key => N * log N => O(NlogN)

there is no option like O(NlogN)

so correct answer is O(N²)

22 votes

-- Akash (41.5k points)

Delete - θ(1) time as pointer is directly given

Insert - O(N) time, we may need to insert at the end of the sorted list..

Find - θ(N) time.. list we need to search sequentially..

Decrease key - θ(N) time, pointer is directly given, delete then insert

Now using above..

$\theta(N) * \theta(1) + O(\log N) * O(N) + O(\log N) * \theta(N) + \theta(N) * \theta(N)$

using property of asymptotic notation, and removing lower terms, we get O(N²)..

So Answer O(N²) C

13 votes

-- Abhilash Panicker (8.7k points)

3.8.2 Linked Lists: GATE1987-1-xv [top](#)

<http://gateoverflow.in/8029>



Selected Answer

suppose we have to insert node p after node q then

$p \rightarrow \text{next} = q \rightarrow \text{next}$

$q \rightarrow \text{next} = p$

so two pointers

b)

1 votes

-- Sanket_ (3.9k points)

3.8.3 Linked Lists: GATE1993_13 [top](#)

<http://gateoverflow.in/2310>



Selected Answer

Following 2 lines are enough to realise above constraint :->>

1. $Y \rightarrow \text{next} = X \rightarrow \text{next}$

2. $X \rightarrow \text{next} = Y$

4 votes

-- Rajesh Pradhan (16.9k points)

3.8.4 Linked Lists: GATE1994-1.17, UGCNET-Sep2013-II-32 [top](#)

<http://gateoverflow.in/2460>



Selected Answer

LIinked list are suitable for

Insertion sort:->> No need to swap here just find appropriate place and join the link

Polynomial manipulation:->> Linked List is natural soln for polynomial manipulation <http://iiith.vlab.co.in/?sub=21&brch=46&sim=490&cnt=697>

Radix sort:-> here we are putting digits according to same position(unit,tens) into buckets;

which can be effectively handle by linked list.

Not Suitable for

Binary search:-> Bcz finding mid element itself takes $O(n)$ time.

So Option B is ans.

5 votes

-- Rajesh Pradhan (16.9k points)

B. Because in binary search we need to have access to the mid of the list in constant time. and finding the mid itself in a linked list takes $O(n)$ time which makes no sense to Binary search which otherwise takes $O(\log n)$.

14 votes

-- Gate Keeda (18.8k points)

3.8.5 Linked Lists: GATE1995-2.22 [top](#)

<http://gateoverflow.in/2634>



Selected Answer

Its D according to me.

Binary search using linked list is not efficient as it will not give $O(\log n)$, because we will not be able to find the mid in constant time. Finding mid in linked list takes $O(n)$ time.

Recursive programs are not efficient because they take a lot of space, Recursive methods will often throw StackOverflowException while processing big sets. moreover it has its own advantages too.

16 votes

-- Gate Keeda (18.8k points)

3.8.6 Linked Lists: GATE1997-18 [top](#)

<http://gateoverflow.in/2278>



Selected Answer

- a. As we are comparing here pair wise so for n elements we require compulsory $n-1$ comparision
- b. S2 is executed only for distinct elements so **max n-1** times and **min 0** when all r duplicates or list contain no or 1 element.
- c. j holds the count on number of distinct elements in the ordered list.

2 votes

-- Rajesh Pradhan (16.9k points)

3.8.7 Linked Lists: GATE1997_1.4 [top](#)

<http://gateoverflow.in/2220>



Selected Answer

- A) & B) Here it is not possible to do it in $O(1)$, unless we have pointer to end of one list. As we have not given that pointer, A & B are not option.
- D) It is not possible to do here in $O(1)$, because we will need to allocate memory for bigger array to hold both list & Copy it.
- C) It is possible in $O(1)$ as we can break list at any location & connect it anywhere. We don't need to traverse to end of anything here !

17 votes

-- Akash (41.5k points)

3.8.8 Linked Lists: GATE1998_19a [top](#)

<http://gateoverflow.in/1733>



Selected Answer

Swaps the two nodes next to p in the linked list.

6 votes

-- Arjun Suresh (287k points)

3.8.9 Linked Lists: GATE1999_11b [top](#)

<http://gateoverflow.in/93575>



Selected Answer

```

let A,B,C,D,E,F be the data ..
A---> B ----> C----> E ---->F
let pointer to E be p..
a node with data D has to be inserted before E..
i ll do one thing , add D just after node E .. it ll take constant time..
Pnext --->Address_of_D
Address_of_D.next-----> Address_of_F

A---> B ----> C----> E -----> F

take a temporary variable and swap the value E and D..
temp = p.data
p.data= p.next.data
p.next.data= temp
now linked list wil look like ..
A---> B ----> C----> D-----> E ---->F
still one more work left.. now pointer p pointing to D so increment pointer p to point data E..
p= p--->next

```

9 votes

-- Digvijay (46k points)

3.8.10 Linked Lists: GATE2002-1.5 [top](#)

<http://gateoverflow.in/809>

Selected Answer

A & C is not correct as we can not do binary search in Linked list.

B seems like average case, be we are asked for worst case.

Worst case is we do not find the element in list. We might end up searching entire list & comparing with each element. So answer -> D. n

7 votes

-- Akash (41.5k points)

3.8.11 Linked Lists: GATE2003-90 [top](#)

<http://gateoverflow.in/973>

Selected Answer

It returns 1 if and only if the linked list is sorted in non-decreasing order- B option.

It returns 1 if the list is empty or has just 1 element also, but if and only if makes A option false.

11 votes

-- Bhagirathi Nayak (13.1k points)

3.8.12 Linked Lists: GATE2004-36 [top](#)

<http://gateoverflow.in/1033>

Selected Answer

The pointer points to the Rear node.

EnQueue: Insert newNode after Rear, and make Rear point to the newly inserted node:

```
//struct node *newNode;
newNode->next = rear->next;
rear->next = newNode;
rear=newNode;
```

DeQueue: Delete the Front node, and make the second node the front node.

```
//rear->next points to the front node.
//front->next points to the second node.
struct node* front;
front = rear->next;
rear->next = front->next;
free(front);
```

32 votes

-- Pragy Agarwal (19.3k points)

3.8.13 Linked Lists: GATE2004-40 [top](#)

<http://gateoverflow.in/1037>

Selected Answer

answer - D

membership is linear search - $O(n_1 + n_2)$

cardinality is linear - $O(n_1 + n_2)$

for union we need to ensure no duplicate elements should be present - $O(n_1 \times n_2)$ for each element we need to check if that element exists in other set

for intersection also for every element in set1 we need to scan set2 - $O(n_1 \times n_2)$

18 votes

-- ankitrokdeonsns (9.1k points)

3.8.14 Linked Lists: GATE2004-IT-13 [top](#)

<http://gateoverflow.in/3654>

In the worst case x could be last or second last node, In that case full traversal of the list is required. Therefore answer is (A).

16 votes

-- suraj (5k points)

Since Q is pointing to node X, it can be done in O(1) time..

Algo:

```
Q -> data = Q ->next -> data; // Copy the value of next node into Q.  
del = Q -> next; // take another pointer variable pointing to next node of Q.  
Q -> next = Q -> next ->next;  
free (del);
```

15 votes

-- gate_asp (655 points)

3.8.15 Linked Lists: GATE2005-IT-54 [top](#)

<http://gateoverflow.in/3815>



Selected Answer

i think it's **B**) 2 1 4 3 6 5 7:

as,

p and q are swapping each other.where q is p->next all the time.

9 votes

-- sumit kumar singh dixit (2.3k points)

3.8.16 Linked Lists: GATE2008-62 [top](#)

<http://gateoverflow.in/485>



Selected Answer

The loop is interchanging the adjacent elements of the list. But after each interchange, next interchange starts from the unchanged elements only (due to p = q -> next;).

1st iteration 1 2 3 4 5 6 7
=> 2 1 3 4 5 6 7

2nd iteration 2 1 4 3 5 6 7

3rd iteration 2 1 4 3 6 5 7

p pointing to null q pointing to 7, as p is false hence q=p? p->next:0; will return q=0 ending the loop
answer = **option B**

7 votes

-- Manali (2.8k points)

3.8.17 Linked Lists: GATE2010-36 [top](#)

<http://gateoverflow.in/2337>



Selected Answer

as per given code p points to last node which should be head in modified.

q is the previous node of tail which should be tail for modified

answer D

15 votes

-- Sankaranarayanan P.N (11k points)

3.8.18 Linked Lists: GATE2017-1-08 [top](#)

<http://gateoverflow.in/118711>



Selected Answer

Here is the implemented code in C (-std=c99).

```
#include <stdio.h>
#include <stdlib.h>

#define M 5
#define N 4

int M_data[M] = {1,2,3,4,5};
int N_data[N] = {6,7,8,9};

typedef struct node {
    int data;
    struct node * next;
}node;

void join(node *m,node *n) {
    node * p = n;
    while(p->next != NULL) p = p->next;
    p->next = m;
}

node * bulk_insert(int list_data[],int size) {
    node * head = NULL;
    for(int i=0;i<size;i++) {
        node * newnode = malloc(sizeof(node));
        newnode->data = list_data[i];
        newnode->next = NULL;

        if(head == NULL) {
            head = newnode;
        }else {
            node * temp = head;
            while(temp->next != NULL) temp = temp->next;
            temp->next = newnode;
        }
    }
    return head;
}
void display(node *);
void list_dealloc(node *); /*deallocation prototype*/

int main() {
    node * m = NULL;
    node * n = NULL;
    // insert m_list data
    m = bulk_insert(M_data,M);
    n = bulk_insert(N_data,N); // commenting this causes runtime error
                                // is list n is empty
    printf("\n before join operation :\n");
    display(m);
    display(n);

    join(m,n);

    printf("\n after join operation :\n");
    display(m);
    display(n);

    //list_dealloc(m); no need now
    list_dealloc(n); // OK
    return 0;
}

void display(node *head) {
    while(head != NULL) {
        printf("%d->",head->data);
        head = head->next;
    }
    printf("null\n");
}
```

```
void list_dealloc(node * head) {
    while(head != NULL) {
        node * temp = head;
        head = head->next;
        free(temp);
    }
}
```

With both **n** and **m** and **n** being **non-empty** linked list, then,

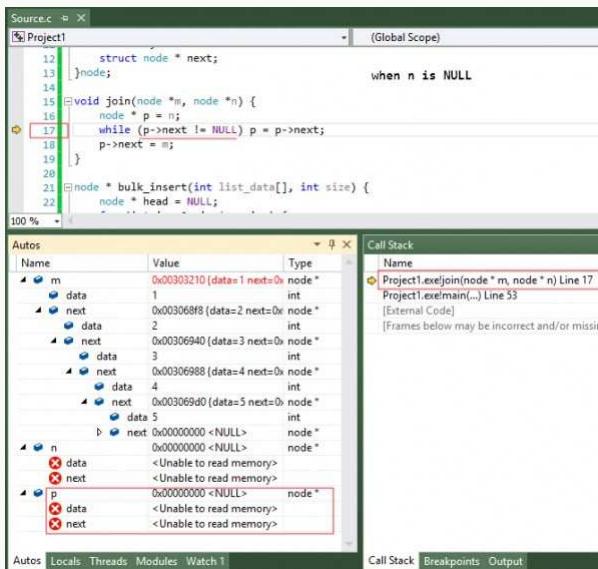
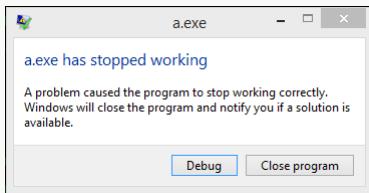
O/P:

```
before join operation :
1->2->3->4->5->null
6->7->8->9->null

after join operation :
1->2->3->4->5->null
6->7->8->9->1->2->3->4->5->null
```

With **n** being **empty** linked list, then,

O/P:



-- Debashish Deka (49.1k points)

here it is explicitly mentioned that LL are terminated by NULL

so suppose i have m as 1->2-> NULL

and n= 3->4->NULL

so according to the code fragment we need to append M to list N

but in the process is the list N is empty but valid like this

node*head;

head->NULL:

it may dereference a NULL pointer
so

i think

B is correct answer here

correct me if i am wrong

18 votes

-- Aboveallplayer (18.5k points)

3.8.19 Linked Lists: ISI2015-CS-3a [top](#)

<http://gateoverflow.in/47273>



Selected Answer

Link 1st node's ptr2 pointer to 2nd node. then 2nd node to 4th node. 4th node to 8th node, 8th node to 16th so on.

For example I need 100th node then do like this- go to **1-->2..>4..>8-->16-->32-->64-->128**. Then it takes **log i** time.

Then to go to 100 start at 64 and linear traversal has to be done for 36 times.

This linear traversal should take **i/2** node traversals in the worst case.

Overall it would take traversing no more than **log i + i/2** nodes.

2 votes

-- Gaurab Ghosh (1.9k points)

3.9

Queues(11) [top](#)

<http://gateoverflow.in/39667>

3.9.1 Queues: GATE 2016-1-10 [top](#)

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

Answer

3.9.2 Queues: GATE 2016-1-41 [top](#)

<http://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](#)
Answer

3.9.3 Queues: GATE1992-09 [top](#)

<http://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
If S is not empty, return
- (ii). FIND-one element of the set S
ONE(S): (any arbitrary element will do)
- (iii). ADD (X): Add integer X to set S
- (iv). 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](#)
Answer

3.9.4 Queues: GATE1994_26 [top](#)

<http://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](#)
Answer

3.9.5 Queues: GATE1996-1.12 [top](#)

<http://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](#)
Answer

3.9.6 Queues: GATE2001-2.16 [top](#)

<http://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

Answer

3.9.7 Queues: GATE2006-49 [top](#)

<http://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);
        }
        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

Answer

3.9.8 Queues: GATE2007-IT-30 [top](#)

<http://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

Answer**3.9.9 Queues: GATE2012_35** [top](#)<http://gateoverflow.in/1758>

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](#)**Answer****3.9.10 Queues: GATE2013-44** [top](#)<http://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 queue?

(A) $\Theta(n)$ (B) $\Theta(n + k)$ (C) $\Theta(nk)$ (D) $\Theta(n^2)$ [gate2013](#) [data-structure](#) [algorithms](#) [normal](#) [queues](#)**Answer****3.9.11 Queues: GATE2017-2-13** [top](#)<http://gateoverflow.in/116253>

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

Answer

Answers: Queues

3.9.1 Queues: GATE 2016-1-10 [top](#)

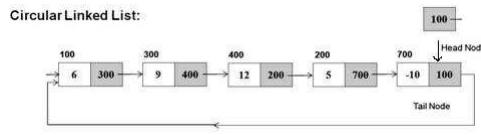
<http://gateoverflow.in/39667>



Selected Answer

Answer A - Circular Queue Implementation

- Both operations can be performed in O(1) time in Circular Queue implementation where Enqueue and Dequeue operation are done at last node. Single pointer needed at last node.



Reffer : <http://www.mathcs.emory.edu/~cheung/Courses/171/Syllabus/8-List/array-queue2.html>

22 votes

-- Abhilash Panicker (8.7k points)

3.9.2 Queues: GATE 2016-1-41 [top](#)

<http://gateoverflow.in/39684>



Selected Answer

256 when 16,15,14,...,1 are present in queue

alternately 15 dequeue & push, 15 pop & enqueue followed by 1 dequeue & push i.e. 31 iterations

brings it to the state 16,15,...,2 in queue and 1 in stack

29 iterations to get to 16,15,...,3 in queue and 2,1 in stack and so on

$31+29+27+\dots+1=16^2=256$

26 votes

-- Krishna murthy (461 points)

3.9.3 Queues: GATE1992-09 [top](#)

<http://gateoverflow.in/588>

A queue with a hashtable.

Intialize hashtable with 0.

When inserting X into the queue update hashtable[X]=0 to hashtable[X]=1.

- If hashtable[X]=1 then return true.
- Return the element at the front or rear of the queue.

(iii) Add the element X to the queue at the rear end and update hashtable[X]=0 to hashtable[X]=1.

(iv) Delete the element X from the front end of the queue and update hashtable[X]=1 to hashtable[X]=0.

3 votes

-- Rajarshi Sarkar (34.4k points)

3.9.4 Queues: GATE1994_26 [top](#)

<http://gateoverflow.in/2522>



Selected Answer

We can do this by first extracting items one by one from Q, and inserting them to S. After all items are done, S will contain the items in reverse order. Now, pop the elements from S and insert to Q. After this operation, items in Q will be in reverse order from the starting. Now, extract items from Q and push on to stack and we are done.

Do

Delete an item from Q

Push the item to S

While (! empty Q);

Do

Pop an item from S

Insert the item to Q

While (! empty S);

Do

Delete an item from Q

Push the item to S

While (! empty Q);

16 votes

-- Arjun Suresh (287k points)

3.9.5 Queues: GATE1996-1.12 [top](#)

<http://gateoverflow.in/2716>



Selected Answer

A.

i) and iv) are false.

[http://en.wikipedia.org/wiki/List_\(abstract_data_type\)#Operations](http://en.wikipedia.org/wiki/List_(abstract_data_type)#Operations)

12 votes

-- Gate Keeda (18.8k points)

3.9.6 Queues: GATE2001-2.16 [top](#)

<http://gateoverflow.in/734>



Selected Answer

A queue can be implemented using two stacks.

Let queue be represented as " q " and stacks used to implement q be "stack1" and "stack2".

q can be implemented in two ways:

Method 1 (By making EnQueue operation costly)

This method makes sure that newly entered element is always at the bottom of stack 1, so that deQueue operation just pops from stack1. To put the element at top of stack1, stack2 is used.

`enQueue(q, x)`

- 1) While stack1 is not empty, push everything from stack1 to stack2.
- 2) Push x to stack1 (assuming size of stacks is unlimited).
- 3) Push everything back to stack1.

`deQueue(q)`

- 1) If stack1 is empty then error
- 2) Pop an item from stack1 and return it

Method 2 (By making deQueue operation costly)

In this method, in en-queue operation, the new element is entered at the top of stack1. In de-queue operation, if stack2 is empty then all the elements are moved to stack2 and finally top of stack2 is returned.

`enQueue(q, x)`

- 1) Push x to stack1 (assuming size of stacks is unlimited).

`deQueue(q)`

- 1) If both stacks are empty then error.
- 2) If stack2 is empty
 - While stack1 is not empty, push everything from stack1 to stack2.
- 3) Pop the element from stack2 and return it.

5 votes

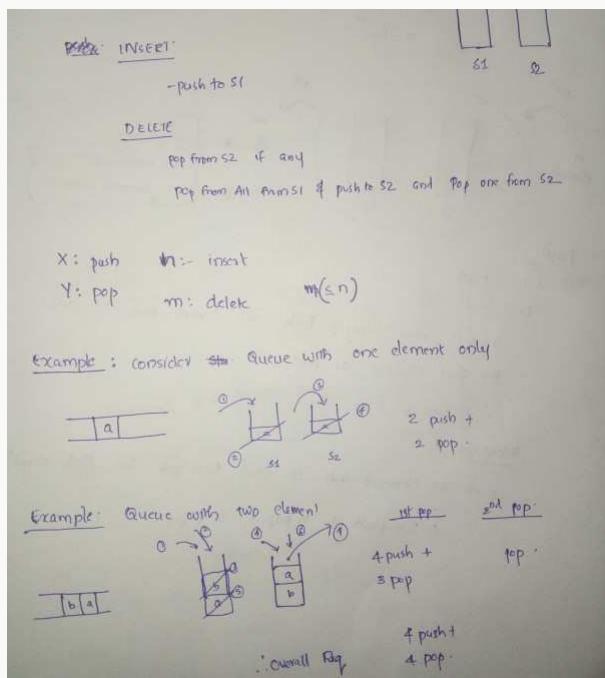
-- Dipak Majhi (685 points)

3.9.7 Queues: GATE2006-49 top

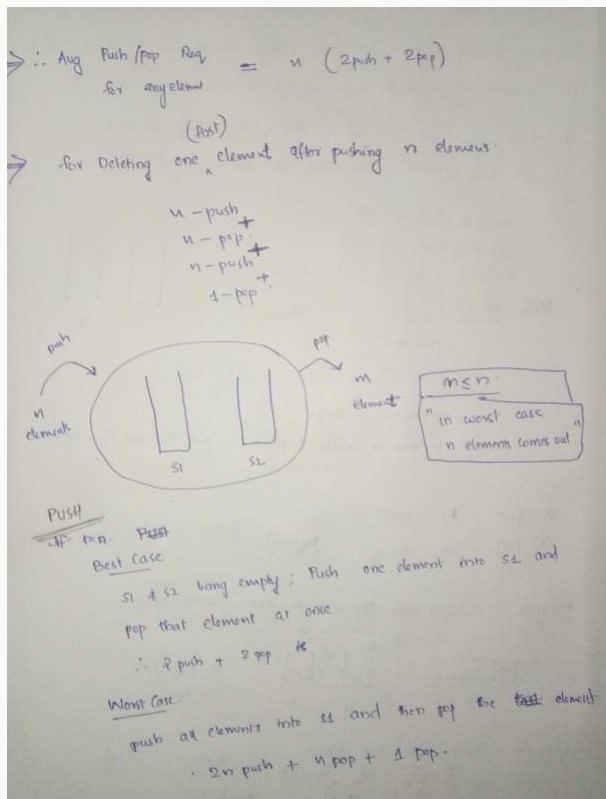
<http://gateoverflow.in/1828>



Selected Answer



Page 1



Page 2

Best Case ↗ push

pushes req. for m elements to come out = $2m$ pushes
 \therefore Remaining elements = $n - m$
 $(m-n)$ elements has to be pushed atleast $\} = \Theta(n-m)$ pushes
 once to stack S_1

\therefore Total number of pushes = $2m + n-m$
 $= (m+n)$ pushes

Worst Case ↗ push

Total pushes if elements in worst case = n
 pushes req. for n elements = $2n$ pushes

$\therefore m+n < X < 2n$

POP

Best Case: $2m < Y < n+m$

push one element and element comes out.
 m elements popped from S_1 and S_2 = $2m$ pops

Worst Case

All ' n ' elements pushed into S_1 n
 All ' n ' elements popped into S_2 $n \rightarrow n$
 Delete ' m ' delete from S_2 $\rightarrow m \rightarrow m$
 \therefore Total # Pop = $n+m$

[Please excuse for the poor handwriting]

10 votes

-- pC (20.7k points)

Answer is (a)

The order in which insert and delete operations are performed matters here.

The best case: Insert and delete operations are performed alternatively. In every delete operation, 2 pop and 1 push operations are performed. So, total $m+n$ push (n push for insert() and m push for delete()) operations and $2m$ pop operations are performed.

The worst case: First n elements are inserted and then m elements are deleted. In first delete operation, $n+1$ pop operations and n push operation are performed. Other than first, in all delete operations, 1 pop operation is performed. So, total $m+n$ pop operations and $2n$ push operations are performed (n push for insert() and m push for delete())

22 votes

-- Kalpana Bhargav (3.2k points)

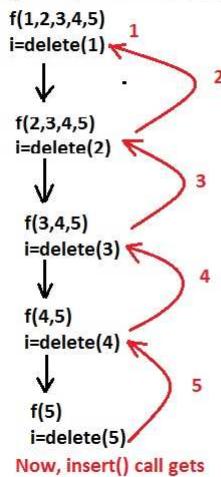
3.9.8 Queues: GATE2007-IT-30 [top](#)

<http://gateoverflow.in/3463>



Selected Answer

Suppose Q contains 1,2,3,4,5



insert() will inserts the value in just reverse order.

11 votes

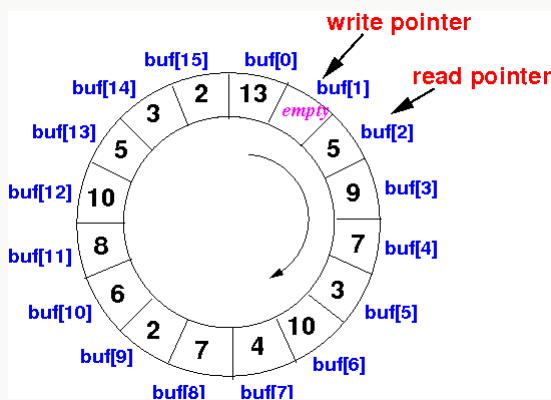
-- srestha (55.2k points)

3.9.9 Queues: GATE2012_35 top

<http://gateoverflow.in/1756>



Selected Answer



rear = Write

front = Read

full: (REAR+1) mod n == FRONT

empty: REAR == FRONT

Only option A matches.

9 votes

-- Prashant Singh (47.6k points)

3.9.10 Queues: GATE2013-44 top

<http://gateoverflow.in/61>



Selected Answer

There are three possible operations on queue- Enqueue, Dequeue and MultiDequeue. MultiDequeue is calling Dequeue multiple times based on a global variable k . Since, the queue is initially empty, whatever be the order of these operations, there cannot be more no. of Dequeue operations than Enqueue operations. Hence, the total no. of operations will be n only.

 27 votes

-- Arjun Suresh (287k points)

3.9.11 Queues: GATE2017-2-13 [top](#)



Selected Answer

Answer is Next pointer to Rear node has Pointer to Front node.

Hence, only *II* is correct.

 18 votes

-- Prashant Singh (47.6k points)

3.10

Stack(17) [top](#)

3.10.1 Stack: GATE1991_03,vii [top](#)

<http://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](#)

[Answer](#)

3.10.2 Stack: GATE1994_1.14 [top](#)

<http://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](#)

[Answer](#)

3.10.3 Stack: GATE1995_2.21 [top](#)

<http://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

[Answer](#)

3.10.4 Stack: GATE1997_4.7 [top](#)

<http://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

[Answer](#)

3.10.5 Stack: GATE2000-13 [top](#)

<http://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

[Answer](#)

3.10.6 Stack: GATE2003-64 [top](#)

<http://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

[Answer](#)

3.10.7 Stack: GATE2004-3 [top](#)

<http://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 $top2$ ($top1 < top2$) point to the location of the topmost element in each of the stacks. If the space is to be used efficiently, the condition for "stack full" is

- A. $(\text{top1} = \text{MAXSIZE} / 2)$ and $(\text{top2} = \text{MAXSIZE} / 2 + 1)$
- B. $\text{top1} + \text{top2} = \text{MAXSIZE}$
- C. $(\text{top1} = \text{MAXSIZE} / 2)$ or $(\text{top2} = \text{MAXSIZE})$
- D. $\text{top1} = \text{top2} - 1$

gate2004 data-structure stack easy

[Answer](#)

3.10.8 Stack: GATE2004-38, ISRO2009-27 [top](#)

<http://gateoverflow.in/1035>

Assume that the operators $+, -, \times$ are left associative and $^$ is right associative. The order of precedence (from highest to lowest) is $^, \times, +, -$. The postfix expression corresponding to the infix expression $a + b \times c - d^e^f$ is

- A. $abc \times +def^{^{\wedge}}-$
- B. $abc \times +de^f^{\wedge}-$
- C. $ab + c \times d - e^f^{\wedge}$
- D. $- + a \times bc^{\wedge}def$

gate2004 stack isro2009

[Answer](#)

3.10.9 Stack: GATE2004-5 [top](#)

<http://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

[Answer](#)

3.10.10 Stack: GATE2004-IT-52 [top](#)

<http://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

[Answer](#)

3.10.11 Stack: GATE2005-IT-13 [top](#)

<http://gateoverflow.in/3758>

A function f defined on stacks of integers satisfies the following properties. $f(\emptyset) = 0$ and $f(\text{push}(S, i)) = \max(f(S), 0) + i$ 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](#)
Answer**3.10.12 Stack: GATE2007-38, ISRO2016-27** [top](#)<http://gateoverflow.in/1236>

The following postfix expression with single digit operands is evaluated using a stack:

$$8 \ 2 \ 3 \ ^ \ / \ 2 \ 3 \ * \ + \ 5 \ 1 \ * \ -$$

Note that \wedge 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](#)
Answer**3.10.13 Stack: GATE2007-IT-32** [top](#)<http://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);
     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?

5 2 * 3 3 2 + *

- A. 15
- B. 25
- C. 30
- D. 150

[gate2007-it](#) [stack](#) [normal](#)
Answer**3.10.14 Stack: GATE2014-2-41** [top](#)<http://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](#)
Answer

3.10.15 Stack: GATE2015-2_38 [top](#)

<http://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](#)
Answer

3.10.16 Stack: GATE2015-3_12 [top](#)

<http://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](#)
Answer

3.10.17 Stack: TIFR2017-B-3 [top](#)

<http://gateoverflow.in/85679>

We have an implementation that supports the following operations on a stack (in the instructions below, s is the name of the stack).

- $\text{isempty}(s)$: returns **True** if s is empty, and **False** otherwise.
- $\text{top}(s)$: returns the top element of the stack, but does not pop the stack; returns **null** if the stack is empty.
- $\text{push}(s, x)$: places x on top of the stack.
- $\text{pop}(s)$: 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](#)

[Answer](#)

Answers: Stack

3.10.1 Stack: GATE1991_03, vii [top](#)



Selected Answer

Lets try something different when you read the word pop then delete the last pushed element and print it ..now delete the push word which we have already executed ..now go on from left to right and do the same

So, Output will be 20, 20, 10, 10, 20

10 votes

-- Bhagirathi Nayak (13.1k points)

Nothing to explain. It will be B.

11 votes

-- Gate Keeda (18.8k points)

3.10.2 Stack: GATE1994_1.14 [top](#)



Selected Answer

push 1 push 2 push 3 pop 3 push 4 pop 4 push 5 pop 5 pop 2 pop 1 then o/p is 3,4,5,2,1 option b

14 votes

-- Sankaranarayanan P.N (11k points)

3.10.3 Stack: GATE1995_2.21 [top](#)



Selected Answer

$$A + B * (C + D) / F + D * E$$

answer = A B C D + * F / + D E * +

14 votes

-- Amar Vashishth (27.9k points)

3.10.4 Stack: GATE1997_4.7 [top](#)

[http://gateoverflow.in/2248](#)



Selected Answer

Implementing stack using priority queue require first element inserted in stack will be deleted at last, and to implement it using `deletemin()` operation of queue will require first element inserted in queue must have highest priority.

So the keys must be in strictly decreasing order.

10 votes

-- Suraj Kaushal (361 points)

3.10.5 Stack: GATE2000-13 [top](#)

<http://gateoverflow.in/684>



Selected Answer

a) For enqueue push operation is sufficient

For dequeue operation do the following

-reverse, pop, reverse

b) Contents of stack from top to bottom:

i) 7 10

ii) 2 5 7 10

iii) 80

8 votes

-- Pooja Palod (31.2k points)

3.10.6 Stack: GATE2003-64 [top](#)

<http://gateoverflow.in/951>



Selected Answer

let us represent stack-life of i^{th} element as $S(i)$. The i^{th} element will be in stack till $(n - i)$ elements are pushed and popped. Plus one more Y for the time interval between the push of i_{th} element and the $i + 1^{th}$ element. So,

$$S(i) = Y + 2.(n - i)(Y + X) = Y + 2.(n - i)Z = Y + 2nZ - 2iZ$$

where $Z = Y + X$

$$\text{average stack-life will, } A = \sum \frac{S(i)}{n}$$

$$nA = nY + 2.n.n.Z - 2.Z.\Sigma i$$

$$nA = nY + 2.n.n.Z - 2.Z \frac{(n(n+1))}{2}$$

$$nA = nY + 2.n.n.Z - Z(n.n) - n.Z$$

$$A = Y + 2.n.Z - (n + 1).Z$$

$$A = Y + (n - 1).Z = Y + (n - 1)(X + Y) = n(X + Y) - X$$

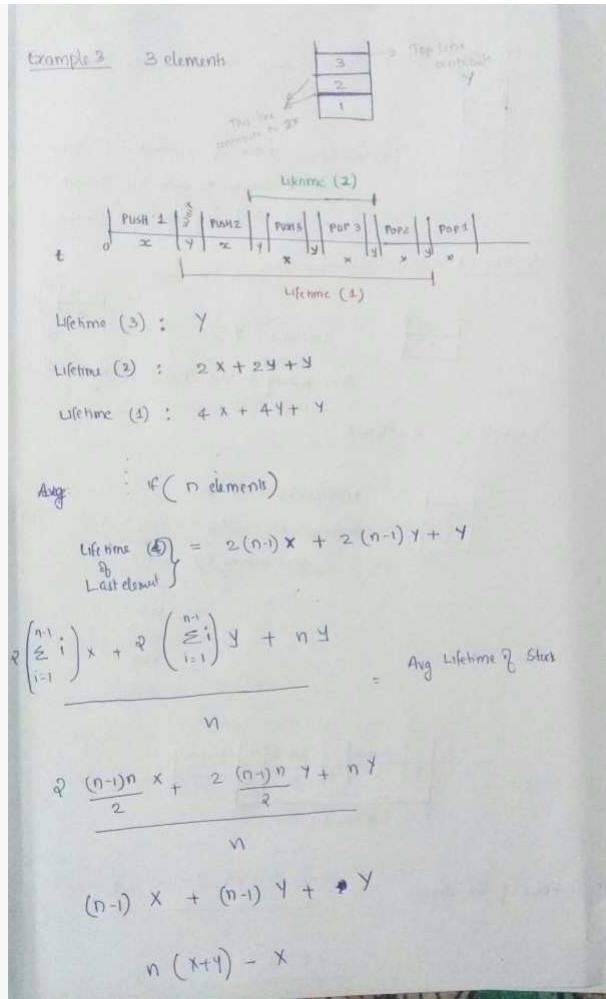
19 votes

-- Vikrant Singh (13.4k points)

Lifetime of m : End Of Push to Start of Pop

- n - PUSH will take X seconds each
- n - POP will take X seconds each
- Time Between two PUSH or two POP or one PUSH one POP is Y second

Consider an example with stack contains 3 elements as shown in Image below



Option C

 11 votes

-- pC (20.7k points)

3.10.7 Stack: GATE2004-3 [top](#)

<http://gateoverflow.in/1000>

Selected Answer

ans d)

Since the stacks are growing from opposite ends, initially $\text{top1} = 1$ and $\text{top2} = \text{MAXSIZE}$. Now, to make the space usage most efficient we should allow one stack to use the maximum possible space as long as other stack doesn't need it. So, either of the stack can grow as long as there is space on the array and hence the condition must be $\text{top1} = \text{top2} - 1$;

 16 votes

-- Aditi Dan (5.3k points)

3.10.8 Stack: GATE2004-38, ISRO2009-27 [top](#)

<http://gateoverflow.in/1035>

Selected Answer

Ans : A

Here is the procedure first :

Scan Infix Expression from left to right whenever you see operand just print it.

But In case of operator

if(stack is empty) then push it.

if(precedence(tos) < precedence(current operator)) push it.

else if (precedence(tos) > precedence(current operator)) pop and print.

else if (precedence(tos) == precedence(current operator)) then check for associativity. In case Operators are Left to right then pop and print it otherwise push the current operator (Right to Left Associativity)
And once you have scanned infix expression completely. Make sure pop all the element and print it in same order.

Here the infix expression is $a+b\times c-d^{\wedge}e^{\wedge}f$

a : print it

+ : push into the Operator Stack

b : print it

* : its having higher precedence than + then push into Operator Stack

c : print it

'-' : '-' is having less precedence than '*' so pop from operator stack and print '*'. after this stack will be having '+' on top. which is having same precedence as '-' but both are left to right associative then just pop '+' and print it. Now stack is empty. Push '-' to it.

d : print it

'^' : top of the stack is having '-' which has lower precedence than '^' so simply push '^' into stack

e : print it.

'^' : Now top of the stack is '^' and has same precedence so associativity will come to picture. Since '^' is right associative as given in question. So '^' will be pushed.

f : print it.

Now we have scanned entire infix expression. Now pop the stack until it becomes empty. This way you will get $abc^*+def^{\wedge}{}^{\wedge}{}^{\wedge}$

13 votes

-- IgnitorSandeep (489 points)

3.10.9 Stack: GATE2004-5 [top](#)

<http://gateoverflow.in/1002>



Selected Answer

STACK Scan the expression from left to right whenever a left parenthesis is encountered just PUSH it into stack and whenever a right parenthesis is encountered just POP it from stack .if at the end of expression we are left with an empty stack then it is a correctly parenthesized expression

11 votes

-- Bhagirathi Nayak (13.1k points)

3.10.10 Stack: GATE2004-IT-52 [top](#)

<http://gateoverflow.in/3695>



Selected Answer

A. push a & pop a, push b & pop b, push c & pop c, and finally push d and pop d
sequence of popped elements will come to abcd

B. first push abcd, and after that pop one by one sequence of popped elements will come to dcba

C. push abc, and after that pop one by one sequence of popped elements will come to cba, now push d and pop d, final sequence comes to cbad

D. this sequence is not possible because 'a' can not be popped before 'b' any how

7 votes

-- Manu Thakur (5.8k points)

3.10.11 Stack: GATE2005-IT-13 [top](#)

<http://gateoverflow.in/3758>



Selected Answer

- i Element to be pushed
- Initial State $f(\phi) = 0$ For Empty Stack $F(S)$ is 0
- Then we push each element (i) one by one and calculate $f(s)$ for each insertion as given

$$F_{new}(S) = \max(F_{previous}(S), 0) + i$$

Is the function to compute $F(s)$ for each insertions

1. INSERT 2 into Stack
 $F_{previous}(S) = 0$ [Stack was empty]
 $i = 2$ (inserting element is i)
 $F_{new}(S) = \max(F_{previous}(S), 0) + i$

$F_{new}(S) = \max(0,0) + 2 = 2$

2. INSERT -3 into Stack

$F_{previous}(S) = 2$ [Stack was empty]
 $i = -3$ (inserting element is i)
 $F_{new}(S) = \max(F_{previous}(S), 0) + i$
 $F_{new}(S) = \max(2, 0) + -3 = -1$

Similarly ,

i : The element to be pushed
S: Stack

Initially f(S)=0.

f(S)	max(f(S),0)	i	Updated_f(S)=max(f(S),0)+i
0	0	2	2
2	2	-3	-1
-1	0	2	2
2	2	-1	1
1	1	2	3

Thus, the answer is **Option C**.

The value of $F(S)$ after inserting all elements into stack is **3**

1 25 votes

-- Shridhar (313 points)

3.10.12 Stack: GATE2007-38, ISRO2016-27 [top](#)

<http://gateoverflow.in/1236>



Selected Answer

push 8 so stack is 8
push 2 so stack is 8 2
push 8 2 3
^ pop 3 and 2 perform opn 2^3 and push to stack. stack is 8 8
/ pop 8 and 8 perform 8/8 and push result to stack . stack is 1
push 2 stack is 1 2
push 3 stack is 1 2 3
* pop 3 and 2 perform by 2*3 and push . stack is 1 6

hence answer is A

1 12 votes

-- Sankaranarayanan P.N (11k points)

3.10.13 Stack: GATE2007-IT-32 [top](#)

<http://gateoverflow.in/3465>



Selected Answer

B) 25
let first part
5 ----push
2-----push
push-----5*2=10. (pops 5 and 2)

```
push 3
push 3
push 2
push 3+2 = 5 (pops 2 and 3)
push 5*3 = 15 (pops (5 and 3)
push 15 + 10 = 25 (pops (15 and 10)
```

14 votes

-- Arpit Dhuriya (3k points)

3.10.14 Stack: GATE2014-2-41 [top](#)

<http://gateoverflow.in/2007>

Selected Answer

(C) is the answer. While ENQUEUE we REVERSE the stack, PUSH the element and then again REVERSE the stack. For DEQUEUE we simply POP the element.

(Option (B) can be used to get the first element from the stack by doing a POP after REVERSE for DEQUEUE and PUSH for ENQUEUE. But we have to restore the stack using REVERSE (otherwise next POP won't work) which means DEQUEUE actually needs 3 instructions and not 2)

16 votes

-- Arjun Suresh (287k points)

3.10.15 Stack: GATE2015-2_38 [top](#)

<http://gateoverflow.in/8164>

Selected Answer

Answer: 15

The code is pushing 5 and 10 on stack and then popping the top two elements and printing their sum.

Refer here: <http://ideone.com/kIUDqT>

21 votes

-- Rajarshi Sarkar (34.4k points)

```
Initially stack is empty = -1
stkFunc (-1, 10); this function

case -1: size = val; break; and static size= 10 // size memory declare one time only// and control comes out of switch b/c of
break

stkFunc (0, 5); this function run

case 0: if (stkTop < size ) A[stkTop++]=val; break; here stktop is static value so memory declare at compile time only now check
if condition 0< 10 true then A[stktop++]== A[0+1]=val= 5 i.e. push 5 into stack break comes so control comes outside

stkFunc (0, 10); this comes

case 0: if (stkTop < size ) A[stkTop++]=val; break; same as above make A[stkTop++]= 10 i.e. push 10 into stack and break comes
so control comes outside

printf ("%d\n", stkFunc(1, 0)+ stkFunc(1, 0));

this function

stkFunc(1, 0) this will run

default: if (stkTop) return A[--stkTop] return top of stack which is 10

stkFunc(1, 0) this will run

default: if (stkTop) return A[--stkTop] return top of stack which is 5

printf ("%d\n", stkFunc(1, 0)+ stkFunc(1, 0));= 5+10=15 15 will be printed
```

17 votes

-- Prashant Singh (47.6k points)

3.10.16 Stack: GATE2015-3_12 [top](#)<http://gateoverflow.in/8408>

we have to keep symbol into stack and when we get two operands followed by operator ..we will apply operator on last two operands

symbol

symbol	stack
10	10 (keep in stack)
5	10 5 (keep in stack)
+	10 5 + = 10+5 = 15 (apply operator on last 2 operands)
60	15 60 (keep in stack)
6	15 60 6 (keep in stack)
/	15 60 6 / = 15 10 (apply operator on last 2 operands)
*	15 10 * = 150 (apply operator on last 2 operands)
8	150 8 (Keep in stack)
-	150 8 - = 150 - 8 = 142 (apply operator on last 2 operands)

So answer is 142

18 votes

-- Praveen Saini (53k points)

3.10.17 Stack: TIFR2017-B-3 [top](#)<http://gateoverflow.in/95679>

D option

First push (((on stack. Now when) comes pop all (((and push) on stack. Now push ((and stack become)((. Now when) come it pop all ((from stack and new stack become)). Again) comes and stack become)) . Now push (((on stack and the stack becomes ((())). Now pop one by one and get option D as the answer.

6 votes

-- Mehak Sharma (1.3k points)

3.11**Trees(14)** [top](#)**3.11.1 Trees: GATE1992_02,vii** [top](#)<http://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](#)
Answer

3.11.2 Trees: GATE1994_5 [top](#)

<http://gateoverflow.in/2501>

A 3 – 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 – ary tree with n internal nodes is $2(n - 1)$.

[gate1994](#) [data-structure](#) [trees](#) [proof](#)
Answer

3.11.3 Trees: GATE1998_1.24 [top](#)

<http://gateoverflow.in/1661>

Which of the following statements is false?

- (a) A tree with 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](#)
Answer

3.11.4 Trees: GATE1998_2.11 [top](#)

<http://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](#)
Answer

3.11.5 Trees: GATE1998_21 [top](#)

<http://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](#)
Answer

3.11.6 Trees: GATE2002-2.9 [top](#)

<http://gateoverflow.in/839>

The number of leaf nodes in a rooted tree of n nodes, with each node having 0 or 3 children is:

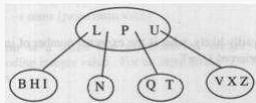
- A. $\frac{n}{2}$
- B. $\frac{(n-1)}{3}$
- C. $\frac{(n-1)}{2}$
- D. $\frac{(2n+1)}{3}$

[gate2002](#) [data-structure](#) [trees](#) [normal](#)
Answer

3.11.7 Trees: GATE2003-65 [top](#)

<http://gateoverflow.in/852>

Consider the following 2-3-4 tree (i.e., B-tree with a minimum degree of two) in which each data item is a letter. The usual alphabetical ordering of letters is used in constructing the tree.



What is the result of inserting G in the above tree?

- A.

```

graph TD
    P((P)) --- GL((G, L))
    P --- U((U))
    GL --- B((B))
    GL --- H((H))
    GL --- I((I))
    U --- QT((QT))
    U --- VXXZ((VXXZ))
  
```
- B.

```

graph TD
    P((P)) --- HL((H, L))
    P --- U((U))
    HL --- BG((B, G))
    HL --- I((I))
    U --- QT((QT))
    U --- VXXZ((VXXZ))
  
```
- C.

```

graph TD
    P((P)) --- ILU((I, L, U))
    P --- VXXZ((VXXZ))
    ILU --- BGH((B, G, H))
    ILU --- L((L))
    ILU --- N((N))
    ILU --- QT((QT))
  
```
- D. None of the above

[gate2003](#) [algorithms](#) [trees](#) [normal](#)

[Answer](#)

3.11.8 Trees: GATE2004-6 [top](#)

<http://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](#)

[Answer](#)

3.11.9 Trees: GATE2005-36 [top](#)

<http://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 nodes is:

- A. nk
- B. $(n - 1)k + 1$
- C. $n(k - 1) + 1$
- D. $n(k - 1)$

[gate2005](#) [data-structure](#) [trees](#) [normal](#)

[Answer](#)

3.11.10 Trees: GATE2007-43 [top](#)

<http://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

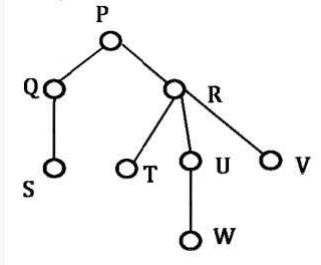
D. 6

gate2007 | data-structure | trees | normal

Answer

3.11.11 Trees: GATE2014-3-12 [top](#)<http://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

Answer

3.11.12 Trees: GATE2014-3-41 [top](#)<http://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

Answer

3.11.13 Trees: GATE2017-1-20 [top](#)<http://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

Answer**3.11.14 Trees: TIFR2012-B-15** [top](#)<http://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, \dots . 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 you say about the distance between u_i and u_{i+1} , as $i = 1, 2, \dots$?

- 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](#)**Answer****Answers: Trees****3.11.1 Trees: GATE1992_02,vii** [top](#)<http://gateoverflow.in/562>

Selected Answer

Answer (a)4, (d) 7

4-> When each leaf has 3 children. So $9/3 = 3$ Internal nodes, Then one internal node connects those internal nodes.

7-> When each leaf has 2 children & one leaf out of 4 gets 3 children. Ex -> $8/4 = 2$ child per internal node. Then one of that internal node gets extra third child. Then 2 internal nodes to connect these 4. Then 1 internal node to connect this 2. So $4+2+1 = 7$.

No other way is possible.

8 votes

-- Akash (41.5k points)

3.11.2 Trees: GATE1994_5 [top](#)<http://gateoverflow.in/2501>

No of nodes at level $i = 3^i$

Let height of tree be h

So total no of internal nodes = $3^0 + 3^1 + 3^2 + \dots + 3^{h-1} = \frac{3^h - 1}{2}$

$2n = 3^h - 1$

No of leaf nodes = $3^h = 2n + 1 = 2(n - 1) + 3$

Let us prove by induction

Base case

$n = 1$ (one internal node i.e., root node)

No of leaves = $2(1 - 1) + 3 = 3$

Let it be true for n internal nodes

Now we prove for m nodes where $m = n + 1$

We have $L(m) = 2(n + 1 - 1) + 3$

Also $L(m) = L(n) + 3 - 1 = 2(n - 1) + 3 + 3 - 1 = 2n + 3$

So if $L(n)$ is true then $L(n + 1)$ is also true

Hence proved by induction.

4 votes

-- Pooja Palod (31.2k points)

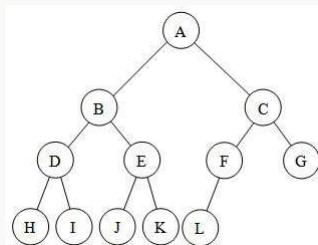
3.11.3 Trees: GATE1998_1.24 [top](#)

<http://gateoverflow.in/1661>



Selected Answer

- Tree with n nodes must have $n-1$ edges.
- A labeled rooted binary tree can be uniquely constructed given its postorder and preorder traversal results. (inorder must be needed with either preorder or postorder for that)
- A complete binary tree with n nodes can have n leaves also
- Example:



Since: **A complete binary tree** is a binary tree in which every level, except possibly the last, is completely filled, and all nodes are as far left as possible. So **false**

- The maximum number of nodes in a binary tree of height h is

$$1 + 2 + 4 + \dots + 2^h = 2^{h+1} - 1 \text{ So true}$$

Answer is b and c both.

Since 2 answers are there I would choose b, because in some places by "complete" binary tree they mean "full binary tree" which has all the levels completely filled.

8 votes

-- Prashant Singh (47.6k points)

3.11.4 Trees: GATE1998_2.11 [top](#)

<http://gateoverflow.in/1683>



Selected Answer

Answer :->

A)

$$x(n-1) + 1$$

Originally when we have root , there is only 1 node, which is leaf. (There is no internal node.) **From this base case "+1" part of formula comes.**

When we n children to root, we make root internal. So then Total Leaves = $= 1(n-1) + 1 = n$.

In complete n ary tree every time you add n children to node, you add $n-1$ leaves & make that node to which you are inserting children internal. (+ n for leaves, -1 for node which you are attaching). So if you had originally few leaves, you add $n-1$ "New" leaves to them. This is how $x(n-1) + 1$ makes sense.

8 votes

-- Akash (41.5k points)

3.11.5 Trees: GATE1998_21 [top](#)

<http://gateoverflow.in/1735>



a) Consider a function $N(h)$ which represents the smallest number of nodes n for an AVL tree with height h and satisfies $n = N(h)$.

For $h=0$ we have, number of nodes = 1. So $N(0) = 1$

For $h = 1$, we have, number of nodes = 2. We could take 3, but we require the smallest graph (a graph with smallest number of nodes) so we take 2. It means that to create a tree with height 1 we need **at least** 2 nodes.

So $N(1) = 2$

Now, for $h = 2$, we need to create a node with a child subtree of height 1. This may be the right or left subtree. But since this is an AVL tree, to balance a child subtree of height let's say H_s , we need the other child to have a height of H_s-1 , H_s or H_s+1 . But we take H_s-1 for minimal case. In simple words, a node with height 5 must have a child with height 4 (H_s) and another child with height 3 (H_s-1). So $N(2)$ can be obtained as:

$N(2) = N(1) + (0) + 1$ (1 is added to count the parent node, $N(1)$ or $N(H_s)$ and $N(0)$ or $N(H_s-1)$ represent two subtrees.)

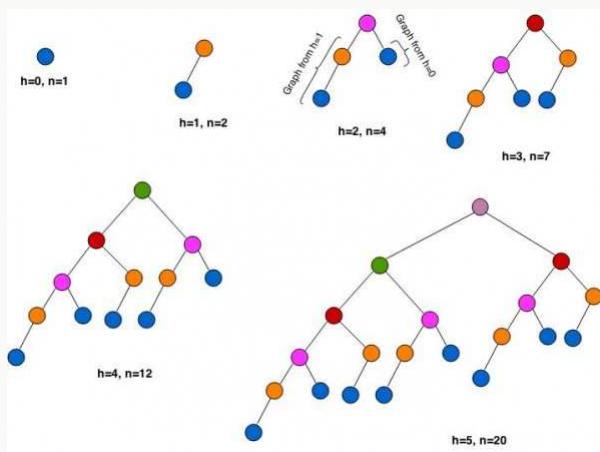
Similarly:

$N(3) = N(2) + N(1) + 1$

and generalizing:

$$N(h) = N(h-1) + N(h-2) + 1$$

This recursion can be graphically seen as below:



b) Using the above recursion, we need to find $N(8)$

$$N(0) = 1$$

$$N(1) = 2$$

$$N(2) = N(1) + N(0) + 1 = 1 + 2 + 1 = 4$$

$$N(3) = N(2) + N(1) + 1 = 2 + 4 + 1 = 7$$

$$N(4) = N(3) + N(2) + 1 = 4 + 7 + 1 = 12$$

$$N(5) = N(4) + N(3) + 1 = 7 + 12 + 1 = 20$$

$$N(6) = N(5) + N(4) + 1 = 12 + 20 + 1 = 33$$

$$N(7) = N(6) + N(5) + 1 = 20 + 33 + 1 = 54$$

$$N(8) = N(7) + N(6) + 1 = 33 + 54 + 1 = 88$$

So answer for b) is 88.



Selected Answer

Answer :-> C) $n(k-1) + 1$

Originally when we have root , there is only 1 node, which is leaf.(There is no internal node.) From that "+1" part of formula comes from this base case.

When we k children to nodes, we make root internal. So then Total Leaves = $n(k-1) + 1 = (k-1) + 1 = k$

In k complete k ary tree every time you add k children , you add k-1 leaves.(+k for leaves, -1 for node which you are attaching)

4 votes

-- Akash (41.5k points)

3.11.10 Trees: GATE2007-43 [top](#)

<http://gateoverflow.in/1241>



Selected Answer

If you do little bit experiments on no of leaves, Internal nodes, you will realize that they have equation like following :-

No of leaves (L) = $(n-1) * \text{Internal Nodes (I)} + 1$

here we need to find n.

Putting values

$$41 = (n-1) * 10 + 1$$

$$(n-1) * 10 = 40$$

$$n-1 = 4$$

$$n = 5$$

So answer = C

10 votes

-- Akash (41.5k points)

Sum of degrees in tree = $L + I * (n+1) - 1 = 10n + 50$ (Each leaf node has degree 1 and all internal nodes have degree $k+1$, except root which has degree k)

So, number of edges = $5n + 25$ (Number of edges in a graph (hence applicable for tree also) is half the sum of degrees as each edge contribute 2 to the sum of degrees)

In a tree with n nodes we have n-1 edges, so with $41+10 = 51$ nodes, there must be 50 edges.

$$\text{So, } 5n + 25 = 50$$

$$5n = 25 \Rightarrow n = 5$$

15 votes

-- Arjun Suresh (287k points)

3.11.11 Trees: GATE2014-3-12 [top](#)

<http://gateoverflow.in/2046>



Selected Answer

A.

the inorder traversal order of a ternary tree is left-->root-->middle-->right.

20 votes

-- Gate Keeda (18.8k points)

3.11.12 Trees: GATE2014-3-41 [top](#)

<http://gateoverflow.in/2075>



Selected Answer

Here, the condition for count value = 1 is

if (tree→leftMostchild == Null)

- so, if there is no left-most child of the tree (or the sub-tree or the current node called in recursion)
- Which means there is no child to that particular node (since if there is no left-most child, there is no child at all as per the tree representation given).
- ∴ the node under consideration is a leaf node.
- The function recursively counts, and adds to value, whenever a leaf node is encountered.
- So, The function returns the number of leaf nodes in the tree.

13 votes

-- Kalpish Singhal (2k points)

3.11.13 Trees: GATE2017-1-20 [top](#)



Selected Answer

Tree with n vertices which means n-1 edges.

$n = 10 \therefore \text{edges} = n - 1 = 9$.

$\therefore \text{Sum of degree of all vertices} \leq 2E \leq 2*9 \leq 18$

7 votes

-- Kantikumar (3.3k points)

3.11.14 Trees: TIFR2012-B-15 [top](#)



This post explains it nicely

<https://www.quora.com/How-does-following-algorithm-for-finding-longest-path-in-tree-work>

3 votes

-- sudipta roy (389 points)

4 Programming & DS: Programming (107) [top](#)

4.1

Aliasing(1) [top](#)

4.1.1 Aliasing: GATE2000-1.16 [top](#)

<http://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](#)
[Answer](#)

Answers: Aliasing

4.1.1 Aliasing: GATE2000-1.16 [top](#)

<http://gateoverflow.in/639>

Selected Answer

A OPTION

↳ 7 votes

-- vinay kumar (637 points)

4.2

Goto(1) [top](#)

4.2.1 Goto: GATE1994-1.5 [top](#)

<http://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](#)
[Answer](#)

Answers: Goto

4.2.1 Goto: GATE1994-1.5 [top](#)

<http://gateoverflow.in/2442>

Selected Answer

Use of goto takes out the structural decomposition of the code and hence it becomes very difficult to verify or debug the code. As far as performance or memory impact is concerned, goto has no effect on them.

↳ 12 votes

-- Arjun Suresh (287k points)

4.3

Identify Function(4) [top](#)

4.3.1 Identify Function: GATE1995-3 [top](#)

<http://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 | normal | identify-function

Answer

4.3.2 Identify Function: GATE1998-2.13 [top](#)

<http://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

Answer

4.3.3 Identify Function: GATE2004-IT-15 [top](#)

<http://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

Answer

4.3.4 Identify Function: GATE2017-2-43 [top](#)

<http://gateoverflow.in/116386>

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](#)

[Answer](#)

Answers: Identify Function

4.3.1 Identify Function: GATE1995-3 [top](#)

<http://gateoverflow.in/2639>

Max. value that can be accommodated inside an unsigned byte = 255 and unsigned int = 65535

A and B are given in Hexadecimal.

A = 5CH = (92)₁₀

B = 92H = (146)₁₀

X = A + B = (238)₁₀ = EEH

Y = abs(A - B) = (54)₁₀ = 36H

W = A - B = (-54)₁₀ = CAH

Z = A * B = (13432)₁₀ = 3478H

2 votes

-- Ravi Ranjan (2.6k points)

4.3.2 Identify Function: GATE1998-2.13 [top](#)

<http://gateoverflow.in/1685>



Selected Answer

call by value: 36,
call by reference: undefined behaviour for C/C++ but 42 for languages having * as a sequence point.

f(x) * f(x);

If the value of x is being modified inside the function (call by reference) we cannot be sure if this modified value or the old value will be passed as argument for the second call to f(). This is because left and right operand of any arithmetic expression in C/C++ can be evaluated in any order. For languages like Java, strict left-right order is maintained.

2 votes

-- Arjun Suresh (287k points)

4.3.3 Identify Function: GATE2004-IT-15 [top](#)

<http://gateoverflow.in/3656>



Selected Answer

Firstly our requirement is for x=1 it makes '0' and for x= 0 it makes '1'

Let's consider options one by one:

A) $X = 1 + X$

For $x = 1$, it gives 2 So, False

C) $X = X - 1$

For $x=0$, it gives -1. So, False

D) $X = 1 \% X$

For $x = 0$, it gives 1 % 0 . I think it is undefined.

Even if we consider $X = X \% 1$

Then for $x = 0$, it gives $0 \% 1 = 0$ But we require 1.

B) $X = 1 - X$

Here B is correct , as

For $x = 0$, it gives 1.

For $x = 1$, it gives 0.

So, **Option (B)** is correct..

8 votes

-- Himanshu Agarwal (15.9k points)

4.3.4 Identify Function: GATE2017-2-43 [top](#)



Selected Answer

Well ,the above program is sorting the array in descending order.

Initially, *while loop* starts execution by *evaluating the initial condition*

`while(done==0)`

For the first time the first for loop will be executed completely ,the content of array will be as follows :

5,3,4,6,2,1

After the second for executed completely the content of array will be as follows:

6,5,3,4,2,1

But the value variable done is still 0 so while loop will execute again,so now the content of array after executing the first for loop will be 6,5,4,3,2,1 and no change in second for loop but still the done variable is 0.

So while loop execute again,now done variable is modified to 1 and there will be no change in done variable because inside first and second for loop no if condition will satisfied .

Finally the while condition is evaluated false and value of array[3] will be printed which is 3.

7 votes

-- Manoj Kumar (37k points)

4.4

Loop Invariants(8) [top](#)

4.4.1 Loop Invariants: GATE 2016-2-35 [top](#)

<http://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](#)

Answer

4.4.2 Loop Invariants: GATE1991-1,vi [top](#)

<http://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](#)

Answer

4.4.3 Loop Invariants: GATE2004-32 [top](#)

<http://gateoverflow.in/1029>

Consider the following program fragment for reversing the digits in a given integer to obtain a new integer.

Let $n = d_1 d_2 \dots d_m$.

```

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. $n = d_1 d_2 \dots d_{m-i}$ and $rev = d_m d_{m-1} \dots d_{m-i+1}$
- B. $n = d_{m-i+1} \dots d_{m-1} d_m$ or $rev = d_{m-i} \dots d_2 d_1$
- C. $n \neq rev$
- D. $n = d_1 d_2 \dots d_m$ or $rev = d_m \dots d_2 d_1$

[gate2004](#) | [programming](#) | [loop-invariants](#) | [normal](#)

Answer

4.4.4 Loop Invariants: GATE2015-1_33 [top](#)

<http://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

```

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](#)

[Answer](#)

4.4.5 Loop Invariants: GATE2017-2-37 [top](#)

<http://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;
    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. $(q==r) \wedge (r==0)$
- B. $(x>0) \wedge (r==x) \wedge (y>0)$
- C. $(q==0) \wedge (r==x) \wedge (y > 0)$
- D. $(q==0) \wedge (y>0)$

[gate2017-2](#) | [programming](#) | [loop-invariants](#)

[Answer](#)

4.4.6 Loop Invariants: TIFR2010-B-30 [top](#)

<http://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 ' \neq ' 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. $s = \sum_{j=0}^N b[j] \wedge 0 \leq i \leq N$
- B. $s = \sum_{j=0}^{i-1} b[j] \wedge 0 \leq i < N$
- C. $s = \sum_{j=0}^i b[j] \wedge 0 < i \leq N$
- D. $s = \sum_{j=1}^N b[j] \wedge 0 \leq i < N$

E. $s = \sum_{j=0}^{i-1} b[j] \& 0 \leq i \leq N$

tifr2010 | programming | loop-invariants

Answer

4.4.7 Loop Invariants: TIFR2010-B-37 [top](#)

<http://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.
- e. The program will terminate with $z = a^b$

tifr2010 | programming | loop-invariants

Answer

4.4.8 Loop Invariants: TIFR2017-B-5 [top](#)

<http://gateoverflow.in/95683>

Consider the following pseudocode 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

Answer

Answers: Loop Invariants

4.4.1 Loop Invariants: GATE 2016-2-35 [top](#)

<http://gateoverflow.in/39578>



Selected Answer

TAKE
 $x = 10, y = 3$

In first case
Iteration 1 (Before)
 $res = 1$ $a^b = x^y$
 $a = 10$
 $b = 3$ $b \times 2 = 1$
 $res = res \times a = 1 \times 10 = 10$
 $b = b - 1 = 2$

Iteration 2 (Before)
 $res = 10$ ~~option X is wrong here~~
 $a = 10$ opt $x^y = 1000$
 $b = 2$ $a^b = 100$
 $x^y \neq a^b$

(D) $b = y$ D is not correct
 $(res \times a)^b = x^y$
 $(10 \times 10)^2 = 10^3$
 $(100)^2 \neq 1000$

one
Using ~~first~~ iteration we
eliminated option A and D

Iteration 3 (Before)
 $res = b \times 2 = 0$
 $a = 10 \times 10 = 100$
 $b = 2 / 2 = 1$

Now
 $res = 10$
 $a = 100$
 $b = 1$

(E) $(res \times a)^b = (res \times a)^b$
 $(10 \times 100)^1 = (10 \times 10)^1$
 $1000 \neq 100$

id Before 3rd iteration we eliminated
option B. Ans is ~~E~~ [C]

Answer ==> C

18 votes

-- Akash (41.5k points)

4.4.2 Loop Invariants: GATE1991-1,vi [top](#)

<http://gateoverflow.in/504>



Selected Answer

A loop invariant is a condition that is always be same before the loop starts , while in the loop and after the loop ends for each iteration.

Here $i \bmod 2 = 0$ is the loop invariant.

11 votes

-- Monanshi Jain (8.1k points)

4.4.3 Loop Invariants: GATE2004-32 [top](#)



Selected Answer

A loop invariant is something that holds at the start of a loop, across each iteration (inside an iteration it can change but before the iteration ends original condition must be true) and at the end also. So, we can check for the satisfiability of the condition at the loop header for start of the loop, for each iteration and also at the exit.

Here, in each iteration the right most digit of n , is moving to the right end of rev . So, answer is A. i.e. the 2 conditions given in A choice are true on entry to loop, after each iteration (not necessarily during an iteration), and at end of loop.

5 votes

-- Arjun Suresh (287k points)

4.4.4 Loop Invariants: GATE2015-1_33 [top](#)



Selected Answer

The loop terminates when $r < y$. So, $r < y$ is one post condition.

In each iteration q is incremented by 1 and y is subtracted from r . Initial value of r is x . So, loop iterates x/y times and q will be equal to x/y and $r = x \% y \Rightarrow x = qy + r$;

So, B choice.

18 votes

-- Arjun Suresh (287k points)

4.4.5 Loop Invariants: GATE2017-2-37 [top](#)



Selected Answer

Here, $x == (y * q + r)$ says q = quotient and r = remainder.

To divide a number with repeated subtraction, quotient should be initialized to 0 and should be incremented for each subtraction.

Initially $q=0 \Rightarrow r = x$.

\therefore Initial conditions should be C] ($q == 0$) && ($r == x$) && ($y > 0$).

6 votes

-- Kantikumar (3.3k points)

4.4.6 Loop Invariants: TIFR2010-B-30 [top](#)



Selected Answer

Whenever we encounter the [*], the variable s holds the sum of all elements $b[0]$ to $b[i - 1]$.

When we first enter the loop, $i = 0$, and s doesn't have any elements summed up.

When we last enter the loop, $i = (N - 1)$ and s contains the sum of elements $b[0]$ through $b[N - 2]$.

We leave the loop when $i = N$, and s gets the sum of elements $b[0]$ to $b[N - 1]$

The only option that matches this behavior is **option E**

$$s = \sum_{j=0}^{i-1} b[j] \& 0 \leq i \leq N$$

6 votes

-- Pragy Agarwal (19.3k points)

4.4.7 Loop Invariants: TIFR2010-B-37 [top](#)

<http://gateoverflow.in/19251>

Typing error in ques:y=y%2

Ans D

```
if x is odd then{
z =a*b will be o/p
}
if x is even {
case 1: y is even then x =2*x and z=0 will be o/p
case 2: y is odd then loop will not terminate .
}
```

3 votes

-- Saurav Shrivastava (1.2k points)

4.4.8 Loop Invariants: TIFR2017-B-5 [top](#)

<http://gateoverflow.in/95883>

Consider the situations when loop gets broken -

1. if i becomes 10, or i becomes equal to y. => i is correct.
2. if y > 10, the control will come out of the loop when i becomes 10. => ii is correct.
3. During I iteration - j =1, i =2

II iteration - j = 2, i =3

III iteration - j = 6, i =4

IV iteration must not occur, because j will become 24. Hence during 3rd iteration break statement must had been executed and hence, y must be 4 (equal to i during the 3rd iteration).

This implies iii is also correct.

Hence, option d is correct.

5 votes

-- tarun_svbk (1k points)

4.5

Programming Constructs(1) [top](#)

4.5.1 Programming Constructs: GATE1999-2.5 [top](#)

<http://gateoverflow.in/7483>

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

[Answer](#)

Answers: Programming Constructs

4.5.1 Programming Constructs: GATE1999-2.5 [top](#)

<http://gateoverflow.in/1483>

Ans)B

Arbitrary goto, recursive call and repeat may enter infinite loop, and hence terminates program may not be able to answer if 'the program does terminate'.

7 votes

-- bahirNaik (3.6k points)

4.6

Programming In C(60) [top](#)

4.6.1 Programming In C: GATE 2016-1-12 [top](#)

<http://gateoverflow.in/3963>

Consider the following "C" program.

```
void f(int, short);
void main()
{
    int i = 100;
    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

[Answer](#)

4.6.2 Programming In C: GATE 2016-1-15 [top](#)

<http://gateoverflow.in/3964>

Consider the following C program.

```
# include <stdio.h>
void mystery (int *ptr, int *ptrb) {
    int *temp;
    temp = ptrb;
    ptrb = ptr;
    ptr = 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](#)
Answer

4.6.3 Programming In C: GATE 2016-1-34 [top](#)

<http://gateoverflow.in/39704>

The following function computes the maximum value contained in an integer array $P[]$ of size n ($n \geq 1$).

```
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](#)
Answer

4.6.4 Programming In C: GATE 2016-2-12 [top](#)

<http://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](#)
Answer

4.6.5 Programming In C: GATE 2016-2-37 [top](#)

<http://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

[Answer](#)

4.6.6 Programming In C: GATE2000-1.11 [top](#)

<http://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

[Answer](#)

4.6.7 Programming In C: GATE2000-1.12 [top](#)

<http://gateoverflow.in/635>

The most appropriate matching for the following pairs

X: m = malloc(5); m = NULL;	1: using dangling pointers
Y: free(n); n -> value = 5;	2: using uninitialized pointers
Z: char *p, *p ='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

[Answer](#)

4.6.8 Programming In C: GATE2000-1.17, ISRO2015-79 [top](#)

<http://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](#)
Answer

4.6.9 Programming In C: GATE2000-2.20 [top](#)

<http://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

[gate2000](#) [programming](#) [programming-in-c](#) [easy](#)
Answer

4.6.10 Programming In C: GATE2001-2.18 [top](#)

<http://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](#)
Answer

4.6.11 Programming In C: GATE2002-1.17 [top](#)

<http://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

[Answer](#)

4.6.12 Programming In C: GATE2002-2.18 [top](#)

<http://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

[Answer](#)

4.6.13 Programming In C: GATE2002-2.8 [top](#)

<http://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

[Answer](#)

4.6.14 Programming In C: GATE2003-2 [top](#)

<http://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

[Answer](#)

4.6.15 Programming In C: GATE2003-89 [top](#)

<http://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](#)

[Answer](#)

4.6.16 Programming In C: GATE2004-33 [top](#)

<http://gateoverflow.in/1030>

Consider the following C program segment:

```
char p[20];
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](#)

[Answer](#)

4.6.17 Programming In C: GATE2004-IT-58 [top](#)

<http://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.

```
#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 (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](#)

Answer

4.6.18 Programming In C: GATE2004-IT-59 [top](#)

<http://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](#)

Answer

4.6.19 Programming In C: GATE2004-IT-60 [top](#)

<http://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();
    printf ("\n");
    return 0;
}
void wrt_it (void)
{
    int c;
    if (?1)

```

```
wrt_it();  
?2  
}
```

- 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);

gate2004-it | programming | programming-in-c | normal

Answer

4.6.20 Programming In C: GATE2004-IT-61 [top](#)

<http://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

Answer

4.6.21 Programming In C: GATE2005-1, ISRO2017-55 [top](#)

<http://gateoverflow.in/1343>

What does the following C-statement declare?

```
int (*f) (int * );
```

- 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](#)

[Answer](#)

4.6.22 Programming In C: GATE2005-32 [top](#)

<http://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](#)

[Answer](#)

4.6.23 Programming In C: GATE2005-IT-53 [top](#)

<http://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 *, char *) {
    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](#)

[Answer](#)

4.6.24 Programming In C: GATE2005-IT-58 [top](#)

<http://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");
```

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](#)

Answer

4.6.25 Programming In C: GATE2006-57 [top](#)

<http://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](#)

Answer

4.6.26 Programming In C: GATE2006-IT-49 [top](#)

<http://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. etter, k, 6, jungle
 D. etter, u, 8, ncestor

gate2006-it | programming | programming-in-c | normal

Answer

4.6.27 Programming In C: GATE2006-IT-50 [top](#)

<http://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);
    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

Answer

4.6.28 Programming In C: GATE2006-IT-51 [top](#)

<http://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](#)
Answer

4.6.29 Programming In C: GATE2007-IT-31 [top](#)

<http://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](#)
Answer

4.6.30 Programming In C: GATE2008-18 [top](#)

<http://gateoverflow.in/418>

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](#)
Answer

4.6.31 Programming In C: GATE2008-60 [top](#)

<http://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

Answer

4.6.32 Programming In C: GATE2008-IT-49 [top](#)<http://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. dlrlid
- D. worow

gate2008-it programming programming-in-c normal

Answer

4.6.33 Programming In C: GATE2008-IT-50 [top](#)<http://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

Answer

4.6.34 Programming In C: GATE2008-IT-51 [top](#)<http://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

Answer

4.6.35 Programming In C: GATE2008-IT-52 [top](#)

<http://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
 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

Answer

4.6.36 Programming In C: GATE2010-11 [top](#)

<http://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

[Answer](#)

4.6.37 Programming In C: GATE2011_22 [top](#)

<http://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) GATE2011
(B) E2011
(C) 2011
(D) 011

gate2011 | programming | programming-in-c | normal

[Answer](#)

4.6.38 Programming In C: GATE2012-3 [top](#)

<http://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
(B) Choice A
(C) Choice A
 Choice B No Choice
(D) Program gives no output as it is erroneous

gate2012 | programming | easy | programming-in-c

[Answer](#)

4.6.39 Programming In C: GATE2012-48 [top](#)

<http://gateoverflow.in/2176>

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?

(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](#)

Answer

4.6.40 Programming In C: GATE2012-49 [top](#)

<http://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

Answer

4.6.41 Programming In C: GATE2014-1-10 [top](#)

<http://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.
- 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

Answer**4.6.42 Programming In C: GATE2014-2-11** [top](#)<http://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](#)**Answer****4.6.43 Programming In C: GATE2014-2-42** [top](#)<http://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](#)**Answer****4.6.44 Programming In C: GATE2015-1_11** [top](#)<http://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](#)**Answer****4.6.45 Programming In C: GATE2015-1_35** [top](#)<http://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);
}

```

- A. 2036, 2036, 2036
 B. 2012, 4, 2204
 C. 2036, 10, 10
 D. 2012, 4, 6

[gate2015-1](#) | [programming](#) | [programming-in-c](#) | [normal](#)

[Answer](#)

4.6.46 Programming In C: GATE2015-2_15 [top](#)

<http://gateoverflow.in/8088>

Consider the following function written in the C programming language :

```

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](#)

[Answer](#)

4.6.47 Programming In C: GATE2015-3_26 [top](#)

<http://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](#)

[Answer](#)

4.6.48 Programming In C: GATE2015-3_30 [top](#)

<http://gateoverflow.in/8488>

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

[Answer](#)

4.6.49 Programming In C: GATE2015-3_48 [top](#)

<http://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

[Answer](#)

4.6.50 Programming In C: GATE2015-3_54 [top](#)

<http://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

[Answer](#)

4.6.51 Programming In C: GATE2015-3_7 [top](#)

<http://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](#)

[Answer](#)

4.6.52 Programming In C: GATE2017-1-13 [top](#)

<http://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](#)

[Answer](#)

4.6.53 Programming In C: GATE2017-1-36 [top](#)

<http://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.
- (C) Abnormal termination and infinite loop respectively.
- (D) Both terminating abnormally.

[gate2017-1](#) | [programming-in-c](#) | [programming](#) | [normal](#)

[Answer](#)

4.6.54 Programming In C: GATE2017-1-53 [top](#)

<http://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](#)

[Answer](#)

4.6.55 Programming In C: GATE2017-1-55 [top](#)

<http://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](#)

[Answer](#)

4.6.56 Programming In C: GATE2017-2-14 [top](#)

<http://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](#)

Answer

4.6.57 Programming In C: GATE2017-2-2 [top](#)

<http://gateoverflow.in/11871>

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](#)

Answer

4.6.58 Programming In C: GATE2017-2-54 [top](#)

<http://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](#)

Answer

4.6.59 Programming In C: GATE2017-2-55 [top](#)

<http://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](#)

[Answer](#)

4.6.60 Programming In C: GateBook_MockTest2(Programming) [top](#)

<http://gateoverflow.in/116969>

```
# include < stdio.h >
main ( )
{
    float total 0.0, q = 1.0, p = 2.0;
    while (p/q > 0.001)
    {
        q = q + q ;
        total = total + p/q ;
        print f ("% f\n", total) ;
    }
}
```

Which one of the following is the integer that best approximates the last number printed ?

- (A) 0
- (B) 1
- (C) 2
- (D) 3

[programming-in-c](#) [gatebook_mt2](#)

[Answer](#)

Answers: Programming In C

4.6.1 Programming In C: GATE 2016-1-12 [top](#)

<http://gateoverflow.in/39638>



Selected Answer

Option 1: 1st argument is short and not int, 2nd is type error(since s is a short variable).. so WRONG
 Option 2: return type is not void.. so WRONG

Option 3: 1 st argument is int, second is again syntax error.. so WRONG

Option 4: Both the arguments and return type match..p is a pointer to short, so *p is value of short.. So ANSWER

26 votes

-- Abhilash Panicker (8.7k points)

4.6.2 Programming In C: GATE 2016-1-15 [top](#)

<http://gateoverflow.in/39642>



Selected Answer

The mystery about mystery function is it does not affect values in main. As in C, parameters are passed by value- even if they are pointer. So, here the pointer values are exchanged within the function only. (we can use * operator to exchange the values at the location of the pointers and this will affect the values in main).

So NO CHANGES in a,b,c,d.

And ANSWER is 2016

34 votes

-- Abhilash Panicker (8.7k points)

4.6.3 Programming In C: GATE 2016-1-34 [top](#)

<http://gateoverflow.in/39704>



Selected Answer

Answer - d,

Hint : Given in the question itself that we start comparing the contents of an array from $a[0]$ and $a[n-1]$ (converging from both side) then condition must be till both meet at a point and that point will be $a=b$.
Hence loop condition should be $a!=b$.

Option C fails for $n = 2, p = [1, 2]$.

18 votes

-- sukanyac (171 points)

4.6.4 Programming In C: GATE 2016-2-12 [top](#)

<http://gateoverflow.in/39565>



Selected Answer

i is called by reference and j is called by value.

so in function f() only value of i might change,
Now in function f(*p,m)
 $*p$ is pointing to i
thus $*p$ is 5.
m is 10 because of call by value of j.

1. $m=10+5$ hence $m=15$
2. $*p=5 + 15$ hence $*p=20$, that is, value of variable i is now 20
3. returns nothing

Now, back to main
 $i=20$ and j is as it is 10

Hence output of printf will be $i+j=20+10 = 30$

Check code in action => <http://codepad.org/mDLsWYp5>

18 votes

-- Shashank Chavan (3.3k points)

4.6.5 Programming In C: GATE 2016-2-37 [top](#)

<http://gateoverflow.in/39602>



Selected Answer

$f(a, 5)$

$p, n = 5$

3	5	2	6	4
---	---	---	---	---

$\max(f(p+1, 5-1), 3-5)$ or $\max(f(p+1, 4), -2)$

$p, n = 4$

5	2	6	4
---	---	---	---

$\max(\max(f(p+1, 4-1), 5-2), -2)$ or $\max(\max(f(p+1, 3), 3), -2)$

$p, n = 3$

--	--	--

$\max(\max(\max(f(p+1, 3 - 1), 2 - 6), 3), -2)$ or $\max(\max(\max(f(p+1, 2), -4), 3), -2)$

$p, n = 2$

6	4
---	---

$\max(\max(\max(\max(f(p+1, 1), 2), -4), 3), -2)$

$n = 1$, return 0

$\max(\max(\max(\max(0, 2), -4), 3), -2)$

$\max(\max(\max(2, -4), 3), -2)$

$\max(\max(2, 3), -2)$

$\max(3, -2)$

3

Check this code => <http://codepad.org/vVicmkN7>

1 21 votes

-- Praveen Saini (53k points)

4.6.6 Programming In C: GATE2000-1.11 [top](#)

<http://gateoverflow.in/634>



Selected Answer

(a) is the answer. [] has greater precedence than * in C. So, s becomes an array of pointers.

1 15 votes

-- gatecse (13.1k points)

4.6.7 Programming In C: GATE2000-1.12 [top](#)

<http://gateoverflow.in/635>



Selected Answer

ans: d)

X: m = NULL; makes the pointer m point to NULL. But the memory created using malloc is still there and cannot be used as we don't have a link to it. Hence, lost memory

Y: n is freed and so pointer n is now pointing to an invalid memory making it a Dangling pointer.

Z: p is not initialized. p = malloc(sizeof(char)); should have been used before assigning 'a' to *p.

1 16 votes

-- Aditi Dan (5.3k points)

4.6.8 Programming In C: GATE2000-1.17, ISRO2015-79 [top](#)

<http://gateoverflow.in/640>



Selected Answer

answer is (c)

Here structure creates the memory for 'array and union', but union only creates the memory for only 'long z' which is the largest size data type inside it.

hence,

short x [5] = 5*2 = 10 bytes [shorts take 2 bytes]

long z = 8 bytes

so (10+8) = 18 bytes

1 19 votes

-- Kalpana Bhargav (3.2k points)

4.6.9 Programming In C: GATE2000-2.20 [top](#)

<http://gateoverflow.in/667>

Selected Answer

Answer: A

At i=0, j=0
At i=1, j=1
At i=2, j=3
At i=3, j=6
At i=4, j=10

6 votes

-- Rajarshi Sarkar (34.4k points)

4.6.10 Programming In C: GATE2001-2.18 [top](#)

<http://gateoverflow.in/738>

Selected Answer

[P1] may cause an error because function is returning the address of locally declared variable.

[P2] will cause a problem because px is an int pointer that is not assigned with any address and we are doing dereferencing.

[P3] will work because memory in bytes of size of int will be reserved and its address will be stored in px that can be further used, once function execution completes, this m/m will still exist in Heap until we free it using free() function.

hence answer is C

22 votes

-- Manu Thakur (5.8k points)

4.6.11 Programming In C: GATE2002-1.17 [top](#)

<http://gateoverflow.in/822>

Selected Answer

(A) Each function call starts a new activation record and since C allows nested function calls more than one activation record can exist between the current activation record and main.

(B) TRUE

(C) Since, C uses static scoping, the actual function calling sequence has no impact on the visibility of global variables. If a variable is not found in the current activation record, it is looked in global address space and this is independent of the calling sequence.

(D) All function calls- whether recursive or not uses the same stack for saving the activation record. There is no need for a different stack as for C compiler a recursive function call and a normal function call make no difference.

20 votes

-- Arjun Suresh (287k points)

4.6.12 Programming In C: GATE2002-2.18 [top](#)

<http://gateoverflow.in/848>

Selected Answer

Answer: (B)

Ref: <http://trevorjim.com/c-and-cplusplus-are-not-context-free/>

6 votes

-- Rajarshi Sarkar (34.4k points)

4.6.13 Programming In C: GATE2002-2.8 [top](#)

<http://gateoverflow.in/838>

Selected Answer

answer is (b)

Formula to evaluate 2-d array's location is:----

$$\text{loc}(a[i][j]) = BA + [(i-lb1)*NC+(j-lb2))*c$$

where BA= Base Address

NC= no. of columns

C= memory size allocated to data type of array

$a[lb1.....ub1] [lb2..... ub2]$

here BA=0 , NC =100, c=1, a[0.....99][0.....99] so lb1=0 , lb2=0

$$\text{loc}(a[40][50])= 0 + [(40-0)*100 + (50-0)]*1$$

$$= 0+[4000+50]*1 = 4050$$

12 votes

-- Kalpana Bhargav (3.2k points)

4.6.14 Programming In C: GATE2003-2 [top](#)



A is an array of pointers to int, and B is a 2-D array.

A[2] =
can take a pointer

A[2][3] =
can take an int

B[1] =
B[1] is the base address of array and it cannot be changed as array in C is a constant pointer.

B[2][3] =
can take an integer

So, A is the answer.

24 votes

-- Arjun Suresh (287k points)

4.6.15 Programming In C: GATE2003-89 [top](#)



Selected Answer

main: x = 5; //Global x becomes 5

P: int x = *y + 2; //local x in P becomes 5+2 = 7

Q: z+=x; //local z in Q becomes 7 + 5 = 12

Q: print(z); //prints 12

P: *y = x - 1; //content of address of local variable y (same as global variable x) becomes 7 - 1 =

P: print(x); //prints local variable x in P = 7

main: print(x); //prints the global variable x = 6

it should be option D

```
?1 is (c = getchar()) != '\n'
?2 is putchar(c);
```

8 votes

-- sumit kumar singh dixit (2.3k points)

4.6.20 Programming In C: GATE2004-IT-61 [top](#)



Selected Answer

first print A B

f1 is call by value the changes applicable only for local

from f1 U V is printed

back in main A B is printed

then in f2 V W is printed

hence answer is B

12 votes

-- Sankaranarayanan P.N (11k points)

4.6.21 Programming In C: GATE2005-1, ISRO2017-55 [top](#)



Selected Answer

1. A function that takes an integer pointer as argument and returns an integer -> int f (int *)
2. A function that takes an integer as argument and returns an integer pointer -> int * f (int)
3. A pointer to a function that takes an integer pointer as argument and returns an integer =>

```
int (*f) (int * );
```

So Answer is C

14 votes

-- Akash (41.5k points)

4.6.22 Programming In C: GATE2005-32 [top](#)



Selected Answer

ans is d)

When a function is called without being defined, C compiler assumes it to return "int" but here foo is returning "double" and hence the compiler will throw type mis-match error.

From C99 on ward, implicit declaration of functions is not even allowed.

15 votes

-- Aditi Dan (5.3k points)

4.6.23 Programming In C: GATE2005-IT-53 [top](#)



Selected Answer

The answer is D

```
#include <stdio.h>
int main(void) {
```

```

return 0;
}

int anagram (char *a, char *b) {
/*
ASCII characters are of 7-bits
so we use count array to represent all the ASCII characters
(ranging 0-127)
*/
int count [128], j;

/*
so this loop will initialize count of all the ASCII characters to be
0 (zero)
*/
for (j = 0; j < 128; j++) count[j] = 0;

j = 0;
/*
"a[j] && b[j]" ensures that anagram returns 0 (false) in case both
strings have different length. Because different length strings cannot
be anagram of each other
*/

/*
Logic:
Below while loop increments ASCII equivalent position for its occurrence
in array 'a'in count array; and decrements ASCII equivalent position
for its occurrence in array 'b'in count array.

Example: a = "ISS" and b = "SIS"
ASCII equivalent of:
I - 73
S - 83

j = 0: Statement A will increment count[ASCII of 'I'] ==> count[73]
count[73] = 0 --> 1
Statement B will decrement count[ASCII of 'S'] ==> count[83]
count[83] = 0 --> -1 and will increment j j = 0 --> 1

j = 1: Statement A will increment count[ASCII of 'S'] ==> count[83]
count[83] = -1 --> 0
Statement B will decrement count[ASCII of 'I'] ==> count[73]
count[73] = 1 --> 0 and will increment j j = 1 --> 2

j = 2: Statement A will increment count[ASCII of 'S'] ==> count[83]
count[83] = 0 --> 1
Statement B will decrement count[ASCII of 'S'] ==> count[83]
count[83] = 1 --> 0 and will increment j j = 2 --> 3

*** END OF LOOP ***

*/
while (a[j] && b[j]) {

A; //count [a[j]]++

/*
Note: j will be increment after count[]-- will execute
Resource: http://www.c4learn.com/c-programming/increment-operator-inside printf
*/
B; //count[b[j++]]--
}

/*
This loop checks that the number of occurrences of the individual ASCII
characters is same or not.
If count[i] = 0 ---- same number of occurrences for ASCII character i
----> return 1 (true)

if count[i] != 0 ---- different number of occurrences for ASCII character i
----> return 0 (false)
*/
for (j = 0; j < 128; j++) if (count [j]) return 0;
return 1;
}

```

14 votes

-- Sohil Ladhani (183 points)

4.6.24 Programming In C: GATE2005-IT-58 [top](#)

<http://gateoverflow.in/3819>



Answer is (B)

For some 'i' if we find that difference of ($A[j] - A[i] < S$) we increment 'j' to make this difference wider so that it becomes equal to S .

If at times difference becomes greater than S we know that it wont reduce further for same 'i' and so we increment the 'i'.

We do it for each 'i' if not found in previous iteration. until i=

12 votes

-- Sandeep_Uniyal (7.3k points)

4.6.25 Programming In C: GATE2006-57 [top](#)



S1 is false.

S2 is true, depending on the argument passed it may generate segmentation fault.

S3 is false because implementation is having some problem. Let x=3 and I want to implement SWAP[x,x]. Now ans would be 0 but that must be x. Problem is because we are not checking whether both pointer are pointing the same address or different so S4 is true.

S5 is obviously false so Option(c) is right

24 votes

-- Kalpana Bhargav (3.2k points)

4.6.26 Programming In C: GATE2006-IT-49 [top](#)



code :

```
#include <stdio.h>

struct test {
    int i;
    char *c;
}st[] = {5, "become", 4, "better", 6, "jungle", 8, "ancestor", 7, "brother"};

int main () {
    //printf("size = %d\n", sizeof(struct test));
    struct test *p = st;
    p += 1;
    ++p->c;           // +(p->c)
    printf("%s,", p++->c);   // (p++)->c
    printf("%c,", *(p->c)); // *(+(p->c))
    printf("%d,", p[0].i);
    printf("%s \n", p->c);
}
```

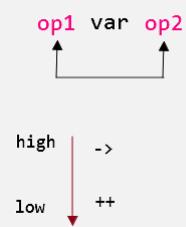
We will assume few things:

- Size of integer 4 Bytes.
- Size of a pointer 4 Bytes.

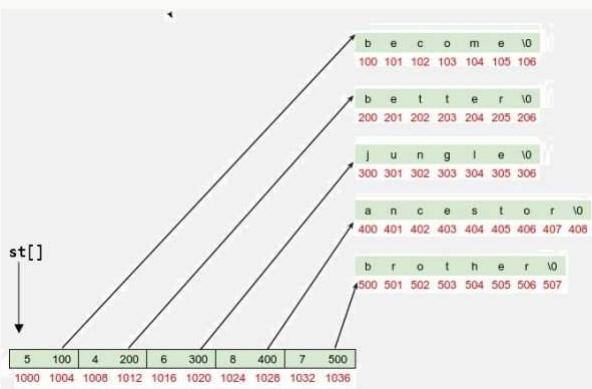
Neglecting any [alignment issues](#) with the storage of this structure we will have 8 Bytes per structure.

And one precedence rule we need to use:

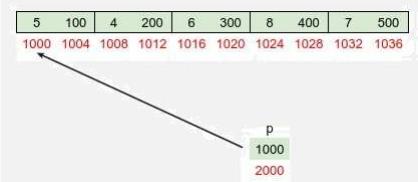
precedence rules



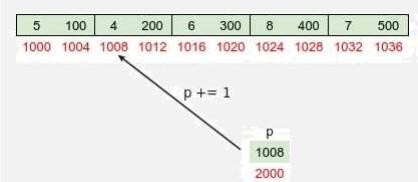
Initial situation :



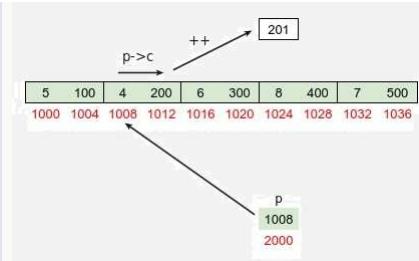
struct test *p = st;



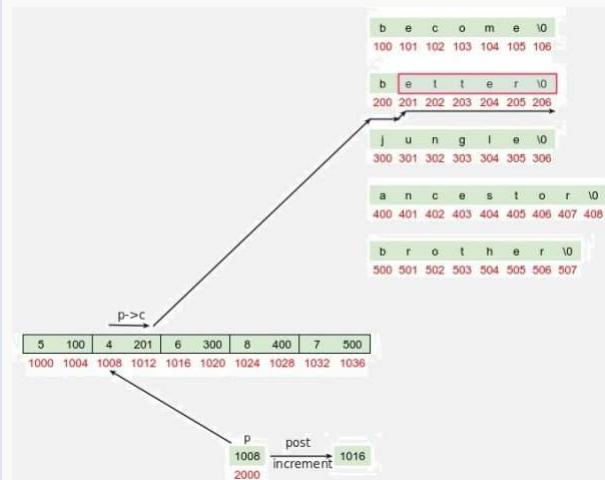
p += 1;

We know that if `ptr` is a pointer then, `ptr + x = ptr + x*sizeof(*ptr);`

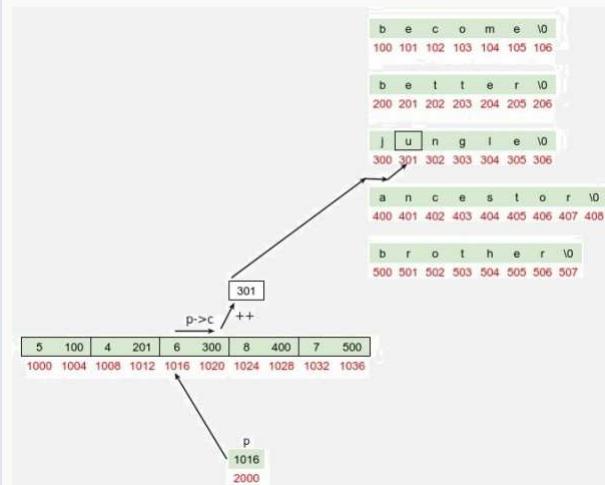
++p->c;



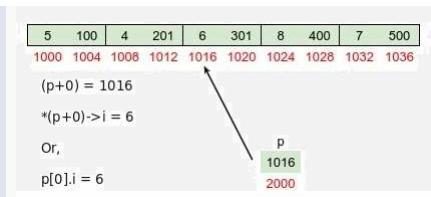
```
printf("%s,", p++->c); // (p++)->c
```



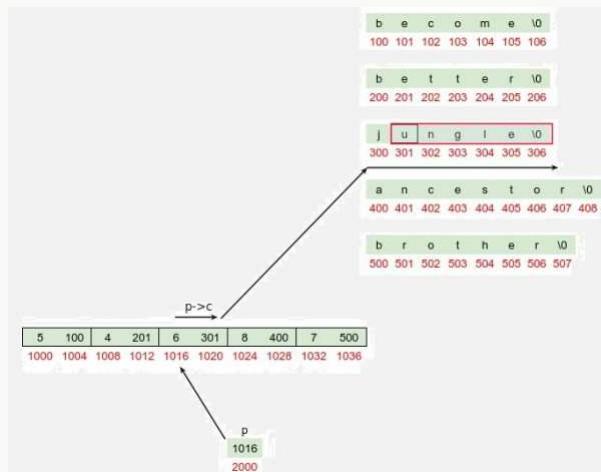
```
printf("%c,", *++p->c); // * (++(p->c))
```



```
printf("%d,", p[0].i);
```



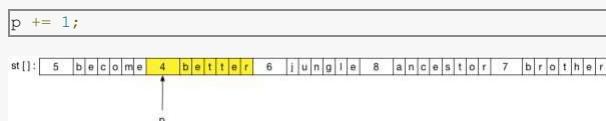
```
printf("%s \n", p->c);
```



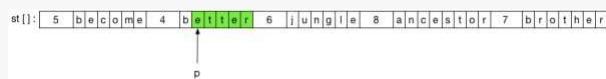
7 votes

-- Debashish Deka (49.1k points)

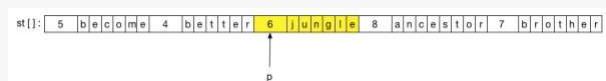
```
#include <stdio.h>
struct test {
    int i;
    char *c;
}st[] = {5, "become", 4, "better", 6, "jungle", 8, "ancestor", 7, "brother"};
main ()
{
    struct test *p = st;
```

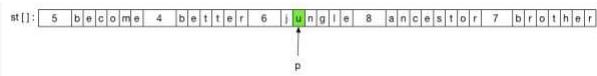


```
++p -> c;
printf("%s", p++ -> c);
```

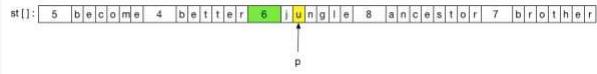


Prints value and increments the pointer. Pointing to the next structure

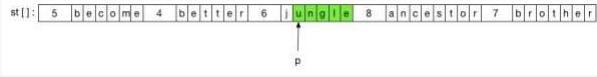




```
printf("%d", p[0].i); //Prints the integer value associated with struct pointed by p
```



```
printf("%s \n", p -> c); //Prints the character string pointed by p
```



PS : Thank you @monanshi for the explanation

24 votes

-- Salman (1k points)

4.6.27 Programming In C: GATE2006-IT-50 [top](#)

<http://gateoverflow.in/3593>



Selected Answer

First of all the swap function just swaps the pointers inside the function and has no effect on the variables being passed.

Inside printab, a and b are added odd integers from 1-5, i.e., $1+3+5 = 9$. So, in first call to printab, $a = -3 + 9 = 6$ and $b = -6 + 9 = 3$.

Static variables have one memory throughout program run (initialized during program start) and they keep their values across function calls. So, during second call to printab, $a = 6 + 9 = 15$, $b = 3 + 9 = 12$.

Hence (D) choice.

19 votes

-- Arjun Suresh (287k points)

4.6.28 Programming In C: GATE2006-IT-51 [top](#)

<http://gateoverflow.in/3594>



Selected Answer

```
a = {a1, a2, a3};
```

```
printf("%d", a[0][2]);
```

$a[0]$ is a_1 . So, this will print $a_1[2] = 8$;

```
printf("%d", *a[2]);
```

$a[2]$ is a_3 . So, this will print $*a_3 = a_3[0] = -12$ ($[]$ has greater precedence than $*$)

```
printf("%d", *++a[0]);
```

$a[0]$ which is a_1 is incremented. a_1 is a pointer to int (base address of an integer array) and so increment means adding `sizeof(int)` and hence a_1 now points to the second element in the array. So, $*++a[0]$ prints second element of a_1 which is 7 and now a_1 starts from 7.

```
printf("%d", * (++a)[0]);
```

$++a$ will increment a , which being an array of pointers (to int) will add `sizeof(pointer)` to a . So, a now contains $\{a_2, a_3\}$ and $a[0]$ will be a_2 and $*a_2$ will be the first element in a_2 which is 23

```
printf("%d\n", a[-1][+1]);
```

$a[-1]$ will subtract a size of pointer from the base address of a . Normally this results in invalid memory access, but since we have incremented a previously, $a[-1]$ is valid and will point to a_1 . So, $a[-1][+1]$ will be $a_1[1]$ which has the value 8.

(a1 was incremented in 3rd printf and hence starts from 7 and not 6. +1 is same as 1, just given to create confusion)

30 votes

-- Arjun Suresh (287k points)

4.6.29 Programming In C: GATE2007-IT-31 [top](#)

<http://gateoverflow.in/3464>



Answer: C

I have tried to explain this question in a better way and the explanation is here...

i	A[i]	for-> if ----- satisfied?	maxsum	sum
-	-	-	0	0
0	A[0] = 2	Yes (i == 0)	0	2
1	A[1] = -2	Yes (a[i] < 0)	2	0
2	A[2] = -1	Yes (a[i] < 0) (for->if->if -----not satisfied)	--	0
3	A[3] = 3	No (else executed)	-- (for->if->if -----not satisfied)	3
4	A[4] = 4	No (else executed)	-- (for->if->if -----not satisfied)	7
5	A[5] = 2	Yes (a[i] < a[i - 1])	7	2

[End of for loop]

```
If (sum (i.e. 2) > maxsum(i.e. 7)) // No
    maxsum = sum; // Not Executed
printf will output maxsum = 7
```

8 votes

-- MKT (191 points)

4.6.30 Programming In C: GATE2008-18 [top](#)

<http://gateoverflow.in/416>



Using option A : x=3, y=4, z=2

a=(3>4)? No

therefore don't evaluate the first part and check second part ((y>z)?y:z)

(4>2)? Yes

a= value of y =4

Answer is (A) $x=3, y=4, z=2$

8 votes

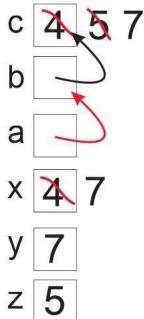
-- Keith Kr (6.2k points)

4.6.31 Programming In C: GATE2008-60 [top](#)

<http://gateoverflow.in/483>



Selected Answer



return $x+y+z = \text{return } 7 + 7 + 5 = \text{return } 19$

so **option B = 19** is correct

10 votes

-- Amar Vashishth (27.9k points)

Answer is **B**.

$z = *ppz$ is a typo and it must be $z = **ppz;$

$**ppz = 1;$ modifies the value of c to 5. $\therefore z = 5.$
 $*py = 2;$ modifies the value of c to 7. $\therefore y = 7.$
 But x will be called as $x = 4,$ $\therefore x = 7.$
 Answer: $7+7+5 = 19.$

13 votes

-- Sona Praneeth Akula (4k points)

4.6.32 Programming In C: GATE2008-IT-49 [top](#)

<http://gateoverflow.in/3359>



Selected Answer

char a[6] =

w	o	r	i	d	\0
---	---	---	---	---	----

a[0] 1 2 3 4 5

After the loop will execute first time,

$a[0] = a[5]$

$a[0] = '\0'$

Next two more iterations of the loop till $i < j$ condition becomes false are not important for the output as the first position is ' $\0$ ';

`printf("%s", a)`, the string starting at address a prints the string starting with ' $\0$ ' and it indicates the end of string, so it will print null string.

so option (B)

16 votes

-- Mitali (225 points)

4.6.33 Programming In C: GATE2008-IT-50 [top](#)

<http://gateoverflow.in/3360>



Answer: C

Only

```
if (num1 >= num2) {swap3 (&num1, &num2);}
```

statement works, which in turn swaps num1 and num2.

11 votes

-- Rajarshi Sarkar (34.4k points)

4.6.34 Programming In C: GATE2008-IT-51 [top](#)

<http://gateoverflow.in/3361>



Answer is (c) 3,2

First 2 variable integer type declared named i and j

Then **int type array** a[8] declared and initialized.

a[0] = 1 , a[1] = 2, a[2] = 3, a[3] = 4, a[4] = 5, a[5] = 6, a[6] = 7,a[7] = 8

Then for loop started

i=0 , i<3 (true)

```
a[0]=a[0]+1 = 1+1=2
i++(outside for loop) , i++(inside for loop);
```

i=2 ,i<3 (true)

```
a[2]=a[2]+1 = 3+1=4
i++, i++(outside for loop) ,
```

i=4, i<3 (false) //Now come out of loop

i-- ; (so i=3)

Now another for loop started where in loop integer type variable named i declared

Block Scope: A Block is a set of statements enclosed within left and right braces ({ and } respectively). Blocks may be nested in C (a block may contain other blocks inside it). A variable declared in a block is accessible in the block and all inner blocks of that block, but not accessible outside the block.

What if the inner block itself has one variable with the same name?

If an inner block declares a variable with the same name as the variable declared by the outer block, then the visibility of the outer block variable ends at the point of declaration by inner block.

So here inner block **int i** has the scope in this block only and outer block int i visibility is not allowed in that block

```
j=7 j>4(true)
int i = 7/2=3
a[3]=a[3]-1=4-1=3
j=6 j>4(true)
int i = 6/2=3
a[3]=a[3]-1=3-1=2
j=5 j>4(true)
int i = 5/2=2
```

```
a[2]=a[2]-1=4-1=3
```

```
j=4 j>4(false)
```

Now when the for loop ends its variable named **i** scope is also end and the outer block variable now visible. So in printf outer variable **i** is used.

so the output would be: 3,2

36 votes

-- Kalpana Bhargav (3.2k points)

4.6.35 Programming In C: GATE2008-IT-52 [top](#)

<http://gateoverflow.in/3362>



Selected Answer

The correct answer is option (B)

first integer type two variables declared i and j

then an integer type 2-d array a[2][3] is declared and initialized and 2-d array b[3][2] is created but not initialized. i.e

	address	value		address	value
a[0][0]	2000	a	b[0][0]	3000	garbage value
a[0][1]	2001	b	b[0][1]	3001	garbage value
a[0][2]	2002	c	b[1][0]	3002	garbage value
a[1][0]	2003	d	b[1][1]	3003	garbage value
a[1][1]	2004	e	b[2][0]	3004	garbage value
a[1][2]	2005	f	b[2][1]	3005	garbage value

now the char type pointer is declared and the base address of array b is put in it. so p=3000

now the for loop is started where i is initialized to 0 ,so

```
i=0 : i<2 (true)
```

```
j=0; j<3 (true)
```

```
*(3000+2*0+0) =a [0][0] => *(3000) = a
```

```
    j++
```

```
j=1; j<3 (true)
```

```
*(3000+2*1+0) =a [0][1] => *(3002) = b
```

```
    j++
```

```
j=2; j<3 (true)
```

```
*(3000+2*2+0) =a [0][2] => *(3004) = c
```

```
    j++
```

```
j=3; j<3 (false)
```

```
i++
```

```
i=1 : i<2 (true)
```

```
j=0; j<3 (true)
```

```
*(3000+2*0+1) =a [1][0] => *(3001) = d
```

```
    j++
```

```
j=1; j<3 (true)
```

```
*(3000+2*1+1) =a [1][1] => *(3003) = e
```

```
    j++
```

```
j=2; j<3 (true)
*(3000+2*2+1) =a [1][2] => *(3005) = f
j++
j=3; j<3 (false)
i++
```

now the values in array b is

b[0][0]	3000	a
b[0][1]	3001	d
b[1][0]	3002	b
b[1][1]	3003	e
b[2][0]	3004	c
b[2][1]	3005	f

hence the output will be (B) choice.

Note:

* (p + 2*j + i)

p + sizeofinner dimension * j + i, hence is same as p[j][i]. Hence with this statement we can identify that the code is transposing the matrix a and storing in b using pointer p.

19 votes

-- Kalpana Bhargav (3.2k points)

4.6.36 Programming In C: GATE2010-11 [top](#)

<http://gateoverflow.in/2184>



Selected Answer

```
p=q;      -- now p and q are pointing to same address i.e. address of j
*p=2;    -- value of j will be updated to 2
hence answer is (D) 0 2
```

16 votes

-- Manu Thakur (5.8k points)

4.6.37 Programming In C: GATE2011_22 [top](#)

<http://gateoverflow.in/2124>



Selected Answer

2011 is the answer.

In C, there is a rule that whatever character code be used by the compiler, codes of all alphabets and digits must be in order. So, if character code of 'A' is x, then for 'B' it must be x+1.

Now %s means printf takes address and prints all bytes starting from that address as characters till any byte becomes the code for '\0'. Now, the passed value to printf here is
 $p + p[3] - p[1]$

p is the starting address of array c. p[3] = 'E' and p[1] = 'A'. So, p[3] - p[1] = 4, and p + 4 will be pointing to the fourth position in the array c. So, printf starts printing from 2 and prints 2011.

(Here "GATE2011" is a string literal and by default a '\0' is added at the end of it by the compiler).

NB: In this question %s is not required.

printf(p + p[3] - p[1]);

Also gives the same result as first argument to printf is a character pointer and only if we want to pass more arguments we need to use a format string.

24 votes

-- Arjun Suresh (287k points)

4.6.38 Programming In C: GATE2012-3 [top](#)<http://gateoverflow.in/35>

Selected Answer

There is a `space` in between the `/` and `n`. ([see-Q-no.-3](#))

```
case 'A' : printf ("Choice A\\ n");
           ^
```

So, [output](#) of the given program is

```
Choice A nChoice BNo Choice
```

Which includes

```
'n'
```

And there is no new line or spaces between outputs. Hence, there is no option matching.

<http://stackoverflow.com/questions/33694700/im-missing-something>

11 votes

-- Arjun Suresh (287k points)

4.6.39 Programming In C: GATE2012-48 [top](#)<http://gateoverflow.in/2176>

Selected Answer

```
main
a=1
prtFun()
a=2
b=1
a= a + ++b = 2+2 = 4
b = 2
printf --> 4 2
back to main
a = a+1 --> 1+1 -->2 (local static a is taken)
prtFun()
a=4 // previous value in the function is retained bcos of static
b=1
a= a + ++b = 4+2 = 6
b = 2
printf --> 6 2
back to main
a = 2
b = 0 (initial value of global b. in prtFun local b is only updated)
printf --> 2 0
```

Answer: C

17 votes

-- Sankaranarayanan P.N (11k points)

4.6.40 Programming In C: GATE2012-49 [top](#)<http://gateoverflow.in/43314>

Selected Answer

```
49
main
a=1
prtFun()
```

```

a=2
b=1
a= a + ++b = 2+2 = 4
b = 2
printf --> 4 2
back to main
a = a+1 --> 1+1 -->2
prtFun()
a=1 //previous a is lost
b=1
a= a + ++b = 2+2 = 4
b = 2
printf --> 4 2
back to main
a = 2
b = 0 (initial value of global b. in prtFun local b is only updated)
printf --> 2 0

```

Answer: D

10 votes

-- Sankaranarayanan P.N (11k points)

4.6.41 Programming In C: GATE2014-1-10 [top](#)

<http://gateoverflow.in/1770>



Selected Answer

```

int i; //i is declared
int*pi = &i; //pi is a pointer variable assigned the address of i
scanf("%d",pi); //i is overwritten with the value we provided because pi is pointing to i earlier
printf("%d\n", i+5) //it will print the value stored in i+5

```

input=3; output=8

Option D is answer.

14 votes

-- Bhagirathi Nayak (13.1k points)

4.6.42 Programming In C: GATE2014-2-11 [top](#)

<http://gateoverflow.in/1965>



Selected Answer

B)

In c, * and / have the same precedence and are left associative.

Evaluation of $n*(n-1)*(n-2)$ might exceed the unsigned int range.

So a) and d) are eliminated.

$n*(n-1)$ is always divisible by 2.(Gives an integer value). Where as it is not always divisible by 3.(You dont always get an integer..truncation possible, less accuracy)

c) eliminated.

In option b)

$n*(n-1)/2$ gives an integer value.

This integer value multiplied by $(n-2)$ again gives an integer value.

Which when divided by 3 gives an integer value(Sets p correctly).

Reason : $n*(n-1)*(n-2)$ is the multiplication of 3 consecutive numbers. which is divisible by 2 as well as 3.
Hence $(n*(n-1)/2*(n-2))$ is divisible by 3.

45 votes

-- Srinath Jayachandran (3.7k points)

4.6.43 Programming In C: GATE2014-2-42 [top](#)

<http://gateoverflow.in/2008>



Selected Answer

There is no updation for i and j in the function. so if we call function with j = 50 the recursive call will be continued infinitely. There is no terminating condition for recursion. hence answer D

11 votes

-- Sankaranarayanan P.N (11k points)

4.6.44 Programming In C: GATE2015-1_11 [top](#)

<http://gateoverflow.in/8185>



Selected Answer

here f1 will not change any values bcz it is call by value but f2 is call by reference and it swaps values of b and c and changes are also reflected in main function...so $5-4-6 = -5$ hence answer is -5

17 votes

-- target gate (241 points)

4.6.45 Programming In C: GATE2015-1_35 [top](#)

<http://gateoverflow.in/8283>



Selected Answer

Address of x is 2000.

x being a 2 D array,

$x + 3 = x + 3 * \text{sizeof its inner dimension}$

$= 2000 + 3 * 3 * 4$ (as inner dimension is 3 integers of size 4)

$= 2000 + 36 = 2036$.

$*(x+3)$ returns the value at address 2036. But since x is 2-D array, one * will just return the 1D array which is the starting address of it, which is 2036 only.

$(x + 2) = 2000 + 2 * 3 * 4 = 2024$

$*(x + 2) + 3 = 2024 + 3 * 4 = 2036$ (The * changes the data type from 2D to 1D and hence + 3 will add $3*4$ and not $3 * 3 * 4$)

So, A.

36 votes

-- Arjun Suresh (287k points)

4.6.46 Programming In C: GATE2015-2_15 [top](#)

<http://gateoverflow.in/8086>



Selected Answer

Ans D as priority of != is greater than that of && in C. The execution happens as

```
if (( *a) && (*a != ' '))
```

So, the if breaks either when $*a = 0$ (not '0' but ASCII 0 or null character '\0'), or when $*a = '$.

So, the recursive call goes like

'A' - 'B' - 'C' - 'D' - ' ' (breaks) and then starts outputting

DCBA

17 votes

-- Vikrant Singh (13.4k points)

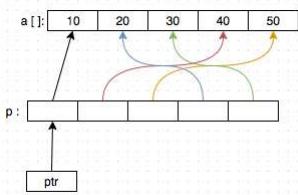
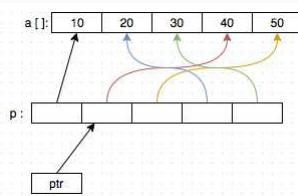
4.6.47 Programming In C: GATE2015-3_26 [top](#)

<http://gateoverflow.in/8478>



Selected Answer

```
static int a[] = {10, 20, 30, 40, 50};
static int *p[] = {a, a+3, a+4, a+1, a+2};
int **ptr = p;
```

`ptr++;`

$$\text{ptr} - p = \frac{\text{address of } \text{ptr} - \text{address of } p}{\text{sizeof}(*\text{ptr})} = 1$$

$$**\text{ptr} = p[2] = *(\text{a} + 3) = 40$$

```
printf("%d%d", ptr-p, **ptr); // 140
```

1 23 votes

-- Salman (1k points)

4.6.48 Programming In C: GATE2015-3_30 [top](#)

<http://gateoverflow.in/8488>

Selected Answer

option C. Only S1 and S2 are correct because Y have same element in both code and in code1

```
Y[i] += X[0][i];
```

this row major order (In C, arrays are stored in row-major order) which gives address of each element in sequential order(1,2,3,...,n) means we cross single element each time to move next shows contiguous in main memory but in code2 for

```
Y[i] += X[i][0];
```

we are crossing n element (row crossing with n element)to move next

1 16 votes

-- Anoop Sonkar (4.8k points)

4.6.49 Programming In C: GATE2015-3_48 [top](#)

<http://gateoverflow.in/8557>

Selected Answer

```
j=2 * 3 / 4 + 2.0 / 5 + 8 / 5;
```

`j = (((2 * 3) / 4) + (2.0 / 5)) + (8/5);` //As associativity of +,* and / are from left to right and + has less precedence than * and /.

`j = ((6/4) + 0.4) + 1;` //2.0 is double value and hence 5 is implicitly typecast to double and we get 0.4. But 8 and 5 are integers and hence 8/5 gives 1 and not 1.6

`j = (1 + 0.4) + 1;` // 6/4 also gives 1 as both are integers

`j = 1.4 + 1;` //1 + 0.4 gives 1.4 as 1 will be implicitly typecast to 1.4

`j = 2.4; // since j is integer when we assign 2.4 to it, it will be implicitly typecast to int.`

`So, j = 2;`

`k -= --j;`

This makes `j = 1` and `k = -1`.

The variables `j` and `k` have values 1 and -1 respectively before the for loop. Inside the for loop, the variable `i` is initialized to 0 and the loop runs from 0 to 4.

`i = 0, k = -1, i + k = -1, default case is executed, printf count = 1`

`i = 1, k = -1, i + k = 0, default case is executed, printf count = 2`

`i = 2, k = -1, i + k = 1, case 2, case 3 and default case is executed, printf count = 5`

`i = 3, k = -1, i + k = 2, case 2, case 3 and default case is executed, printf count = 8`

`i = 4, k = -1, i + k = 3, case 3 and default case is executed, printf count = 10`

`i = 5, loop exits and the control returns to main`

Answer: 10

33 votes

-- Shyam Singh (1.5k points)

4.6.50 Programming In C: GATE2015-3_54 [top](#)

<http://gateoverflow.in/8563>



Selected Answer

The variable `x` is initialized to 1. First and only call to `f1()` returns 26. First call to `f2()` returns 51. First and only call to `f3()` returns 100. Second call to `f2()` returns 52 (The value of local static variable `x` in `f2()` retains its previous value 51 and is incremented by 1).

$$x = 1 + 26 + 51 + 100 + 52 = 230$$

Answer: 230

28 votes

-- Shyam Singh (1.5k points)

4.6.51 Programming In C: GATE2015-3_7 [top](#)

<http://gateoverflow.in/8401>



Selected Answer

`p = s1 + 2;`

`p` now points to the third element in `s1`.

`*p = '0';`

The third element in `s1` is made 0. So, 1234 becomes 1204. C choice.

25 votes

-- Arjun Suresh (287k points)

4.6.52 Programming In C: GATE2017-1-13 [top](#)

<http://gateoverflow.in/118293>

Option (D).

`malloc()` return a void type address (like `1201x`), so no need for typecasting and now `x` contain a location. now `x=assignval(x,0)` is called that assign 0 on memory location pointed by `x` and return `x` (value of `x`, not value at address pointed by `x`). now again `x` contain same address (`1201x`). if (`x`) is evaluated to true because `x` contain a non zero value (`1201x`) and on `x=(int*)malloc(sizeof(int));` assign a new location to `x`, so we lost the previous memory location because now we have no reference to that location and we can't free that location, so execution may result in memory leak.

points:

`malloc()` return a void type pointer so no need to typecasting.

`x==NULL` or `NULL==x` both are valid.

dangling pointer example

```
{
    char *dp = NULL;
    /* ... */
    {
        char c;
        dp = &c;
    }
    /* c falls out of scope */
    /* dp is now a dangling pointer */
}
```

reference :

<http://www.geeksforgeeks.org/what-is-memory-leak-how-can-we-avoid/>

3 votes

-- Niraj Raghuvanshi (199 points)

4.6.53 Programming In C: GATE2017-1-36 [top](#)

<http://gateoverflow.in/118319>



Selected Answer

Answer should be C

Consider foo

Initially val=3

`foo(val--)`

is equivalent to

1.....`foo(val)`

2..... `val=val-1`

so

1.....`foo(3)`

2..... `val=2`

`foo(3)` calls `foo(3)` which in turn calls `foo(3)` this goes on

so here we can see that `foo(3)` is called infinite number of times which causes memory overflow and abrupt termination

and 1 more thing to observe is infinite loop is not there since the `val` is decrement in the first iteration.

consider bar

Here we can see the `val` is not decrement in the loop

so

1. `bar(3)` will call `bar(2)`
2. `bar(2)` will call `bar(1)`
3. `bar(1)` will call `bar(0)`----- Here `bar(0)` return 0
4. `bar(1)` will call `bar(0)`
5. `bar(1)` will call `bar(0)`

This will go on so here there is a problem of infinite loop but not abrupt termination since it doesn't cause memory overflow

4 votes

-- (points)

4.6.54 Programming In C: GATE2017-1-53 [top](#)

<http://gateoverflow.in/116473>

Selected Answer

$$(\text{strlen}(s) - \text{strlen}(t)) = 3 - 5 = -2$$

But in C, when we do operation with two unsigned integers, result is also unsigned. (`strlen` returns `size_t` which is unsigned in most systems). So, this result "-2" is treated as unsigned and its value is `INT_MAX - 2` (not sure about all systems, but at least on systems using 2's complement representation). Now, the comparison is between this large number and another unsigned number `c` which is 0. So, the comparison return TRUE here.

Even if '`c`' is declared as "int", while doing an operation with an unsigned int, it gets promoted to unsigned int and we get the same result.

Hence $(\text{strlen}(s) - \text{strlen}(t)) > 0$ will return 1 on execution thus the conditional operator will return the true statement which is `strlen(abc) = 3`.

Ans should be 3.

10 votes

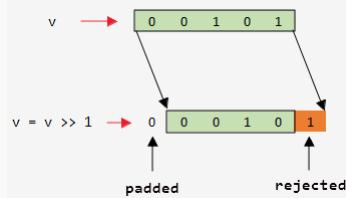
-- Arnabi (6k points)

4.6.55 Programming In C: GATE2017-1-55 [top](#)

<http://gateoverflow.in/116442>

Selected Answer

```
// inside total()
while(v) {
    count += v&1;    \\ check the lowest bit of v
    v >>= 1;        \\ or v = v >> 1 : right shift the bit pattern of v
}
```



This piece of code will count the no of set bits in `v`.

In the `main` function, `i` values goes from 5 to 1, So there will be 5 calls to `total()`.

Each call to `total(i)` counts the no of set bits in `i`. But the `count` is a `static` variable,

So, `total(i) = Total no of set bits in all i ≤ 5.`

```

x = 0

i = 5 :: x = 0 + total(5); // x becomes = 2 :: 5 = 0101 // total(5) = 2
i = 4 :: x = 2 + total(4); // x becomes = 5 :: 4 = 0100 // total(4) = 1 + 2 = 3
i = 3 :: x = 5 + total(3); // x becomes = 10 :: 3 = 0011 // total(3) = 2 + 3 = 5
i = 2 :: x = 10 + total(2); // x becomes = 16 :: 2 = 0010 // total(2) = 1 + 5 = 6
i = 1 :: x = 16 + total(1); // x becomes = 23 :: 1 = 0001 // total(1) = 1 + 6 = 7

```

$$x = \sum_{i=1}^5 [\text{total}(i)] = 2 + 3 + 5 + 6 + 7 = 23$$

8 votes

-- Debashish Deka (49.1k points)

4.6.56 Programming In C: GATE2017-2-14 [top](#)

<http://gateoverflow.in/118245>



Selected Answer

at first in loop we are giving $x=0$ then ptr is pointing to X.

so $*\text{ptr}=0$

now we copying the value of ptr to y ,so $Y=0$

```

x=0;           //value of x = 0
ptr= &x;        // ptr points to variable x
y= *ptr;       // Y contain value pointed by ptr i.e. x= 0;

```

now value of ptr is changed to 1. so the location of X itself got modified

```
*ptr=1;
```

as it is pointing to x so x will also be changed to 1

so 1,0 will be the value

C is correct answer here

11 votes

-- Aboveallplayer (18.5k points)

4.6.57 Programming In C: GATE2017-2-2 [top](#)

<http://gateoverflow.in/118171>



Selected Answer

static char var = Avariable located in data section of memory

$m = \text{malloc}(10); m = \text{null}$; **Here Free(m) missing : So** .A lost memory which cannot be freed

Char * Ptr[10]; Sequence of **10 memory location** to store addresses

register int var1; = request to alloacte a **CPU register** to store data

Answer is A

6 votes

-- Prashant Singh (47.6k points)

4.6.58 Programming In C: GATE2017-2-54 [top](#)

<http://gateoverflow.in/118272>

Selected Answer

hence answer is 0.

5 votes

-- Arnabi (6k points)

4.6.59 Programming In C: GATE2017-2-55 [top](#)

<http://gateoverflow.in/118335>

Selected Answer

```
char c[]="GATECSIT2017";
char *p=c;
printf("%d",strlen(c+2[p]-6[p]-1));
```

2[p] = *(2+p) = p[2]
6[p] = *(6+p) = p[6]
c + 2[p] - 6[p] - 1 = c + 'T' - 'I' - 1 = c + 11 - 1 = c + 10 (In any character coding all alphabet letters are assigned consecutive int values as per C)
printf will print 2 which is the length of "17".

17 votes

-- Arjun Suresh (287k points)

4.6.60 Programming In C: GateBook_MockTest2(Programming) [top](#)

<http://gateoverflow.in/116969>

By looking at first we may find that There is mistake in solution given by them but if we read question carefully than they have given that, best integer that **approximates**

Solution goes like this,

$2/2 + 2/4 + 2/8 + 2/16 + \dots + 2/1024$

Which is nothing but, $1 + (1023/1024) = 1.9990234375 \approx 2.00$

P.S. My solution is so quick for now but I just want to show that significance of word 'Approximate' because I just read 'integer' and I written '1' as an answer.

1 2 votes

-- Meet Samirbhai Shah (3.2k points)

4.7

Programming Paradigms(3) [top](#)

4.7.1 Programming Paradigms: GATE2004-1 [top](#)

<http://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](#)

Answer

4.7.2 Programming Paradigms: GATE2004-90 [top](#)

<http://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 Q-3 R-4 S-1
- B. P-4 Q-3 R-2 S-1
- C. P-3 Q-4 R-1 S-2
- D. P-3 Q-4 R-2 S-1

[gate2004](#) [programming](#) [normal](#) [programming-paradigms](#)

Answer

4.7.3 Programming Paradigms: GATE2008-IT-13 [top](#)

<http://gateoverflow.in/3273>

Match the programming paradigms and languages given in the following table.

	Paradigms	Languages
(I)	Imperative	(a) Prolog
(II)	Object Oriented	(b) Lisp
(III)	Functional	(c) C, Fortran 77, Pascal
(IV)	Logic	(d) C++, Smalltalk, Java

- A. I-c, II-d, III-b, IV-a
- B. I-a, II-d, III-c, IV-b
- C. I-d, II-c, III-b, IV-a
- D. I-c, II-d, III-a, IV-b

[gate2008-it](#) [programming](#) [programming-paradigms](#) [easy](#)

Answer

Answers: Programming Paradigms

4.7.1 Programming Paradigms: GATE2004-1 [top](#)

<http://gateoverflow.in/998>



Selected Answer

Answer is (c) The goal of structured programming is to able to infer the flow of control from the program text . It means user can execute the code according to his requirement. C and Pascal are good example of structured programming. In structured programming control passes one instruction to another instruction in sequential manner.

Avoiding the use of GOTO statements is not the goal of structured programming, it (avoiding the use of GOTO) is one of the requirements for a program to be structured.

12 votes

-- Kalpana Bhargav (3.2k points)

4.7.2 Programming Paradigms: GATE2004-90 [top](#)

<http://gateoverflow.in/1084>



Selected Answer

D should be the answer.

6 votes

-- sumit kumar singh dixit (2.3k points)

4.7.3 Programming Paradigms: GATE2008-IT-13 [top](#)

<http://gateoverflow.in/3273>



Selected Answer

A is correct. Lisp is a pure functional language and Prolog is a logic language. Other languages are well known.

12 votes

-- Arjun Suresh (2.87k points)

4.8

Pseudo Code(1) [top](#)

4.8.1 Pseudo Code: GATE2008-61 [top](#)

<http://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);

Answer

Answers: Pseudo Code

4.8.1 Pseudo Code: GATE2008-61 [top](#)



Selected Answer

Here we are using the '=' operator which has less priority than '!=>' operator. So (c=getchar()) has to be in brackets and after reversing the string we use function putchar(c) for printing the character So option (d) is the right answer

11 votes

-- Kalpana Bhargav (3.2k points)

4.9

Recursion(18) [top](#)

4.9.1 Recursion: GATE 2016-1-35 [top](#)

<http://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. 312213444
- B. 312111222
- C. 3122134
- D. 3121112

[gate2016-1](#) [programming-in-c](#) [recursion](#) [normal](#)
Answer

4.9.2 Recursion: GATE1991_01,x [top](#)

<http://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](#)
Answer

4.9.3 Recursion: GATE1994_21 [top](#)

<http://gateoverflow.in/251>

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

[Answer](#)

4.9.4 Recursion: GATE1995-2.9 [top](#)

<http://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(concat(tail(tail(s)))))
```

- A. ac
- B. bc
- C. ab
- D. cc

gate1995 algorithms normal recursion

[Answer](#)

4.9.5 Recursion: GATE2000-16 [top](#)

<http://gateoverflow.in/687>

A recursive program to compute Fibonacci numbers is shown below. Assume you are also given an array $f [0.....m]$ with all elements initialized to 0

```
fib(n) {
    if (n > M) error ();
    if (n == 0) return 1;
    if (n == 1) return 1;
    if (n == 2) _____ (1)
        return _____ (2)
    t = fib(n - 1) + fib(n - 2);
    _____ (3)
    return t;
}
```

- A. Fill in the boxes with expressions/statements 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

[Answer](#)

4.9.6 Recursion: GATE2001-13 [top](#)

<http://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](#)

[Answer](#)

4.9.7 Recursion: GATE2002-11 [top](#)

<http://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](#)

[Answer](#)

4.9.8 Recursion: GATE2004-31, ISRO2008-40 [top](#)

<http://gateoverflow.in/1028>

Consider the following C function:

```
int f(int n)
{
    static int i = 1;
    if(n >= 5) return n;
    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](#)

[Answer](#)

4.9.9 Recursion: GATE2005-81a [top](#)

<http://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

[Answer](#)

4.9.10 Recursion: GATE2005-81b [top](#)

<http://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 $\text{foo}()$ and stores the value of $\text{foo}(i)$ $0 \leq i < n$, as and when they are computed. With this modification the time complexity for function $\text{foo}()$ 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

[Answer](#)

4.9.11 Recursion: GATE2007-42 [top](#)

<http://gateoverflow.in/1240>

Consider the following C function:

```
int f(int n)
{
    static int r = 0;
    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

[Answer](#)

4.9.12 Recursion: GATE2007-IT-27 [top](#)

<http://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](#)

Answer

4.9.13 Recursion: GATE2014-2-40 [top](#)

<http://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](#)

Answer

4.9.14 Recursion: GATE2017-1-35 [top](#)

<http://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 $\text{fun1}(5)$ is called is

- (A) 53423122233445
- (B) 53423120112233
- (C) 53423122132435
- (D) 53423120213243

[gate2017-1](#) | [programming](#) | [normal](#) | [tricky](#) | [recursion](#)

Answer

4.9.15 Recursion: Gate Sample Questions [top](#)

<http://gateoverflow.in/130912>

```
int fun (int n)
{
6
int x=1, k;
if (n==1) return x;
```

```
for (k=1; k < n; ++k)
x = x + fun(k) * fun(n - k);
}
```

The return value of $\text{fun}(5)$ is _____.

- (a) 0 (b) 26
- (c) 51 (d) 71

[gate](#) [programming](#) [isisamplepapers](#) [recursion](#)

[Answer](#)

4.9.16 Recursion: ISICAL MTech 2014 CS [top](#)

<http://gateoverflow.in/27885>

How many asterisks (*) in terms of k will be printed by the following C function, when called as $\text{count}(m)$ where $m = 3^k$? Justify your answer.

Assume that 4 bytes are used to store an integer in C and k is such that 3^k can be stored in 4 bytes.

```
void count(int n) {
    printf("*");
    if(n>1) {
        count(n/3);
        count(n/3);
        count(n/3);
    }
}
```

[programming-in-c](#) [recursion](#) [isi-2014](#)

[Answer](#)

4.9.17 Recursion: TIFR2010-B-31 [top](#)

<http://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 - 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](#)

[Answer](#)

4.9.18 Recursion: TIFR2011-B-38 [top](#)

<http://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 | programming-in-c | recursion

[Answer](#)

Answers: Recursion

4.9.1 Recursion: GATE 2016-1-35 [top](#)

<http://gateoverflow.in/39730>



Selected Answer

Here **d** is Static, so the value of **d** will persists between the function calls.

1. count(3) will print the value of n and d and increments d and callcount(2) = prints **3 1**.
2. count(2) will print the value of n and d and increments d and callcount(1) = prints **2 2**.
3. count(1) will print the value of n and d and increments d => prints **1 3**.

Now it will return and prints the final incremented value of **d** which is **4**, 3 times.
 So, option A is correct = 3 1 2 2 1 3 4 4 4

23 votes

-- Monanshi Jain (8.1k points)

4.9.2 Recursion: GATE1991_01,x [top](#)

<http://gateoverflow.in/507>



Selected Answer

The recurrence relation for the no. of calls is

$$T(n) = T(n - 1) + T(n - 2) + 2$$

$T(0) = T(1) = 0$ (for fib(0) and fib(1), there are no recursive calls).

$$T(2) = 2$$

$$T(3) = 4$$

$$T(4) = 8$$

$$T(5) = 14$$

$$T(6) = 24$$

$$T(7) = 40.$$

Counting the initial call we get $40 + 1 = 41$.

7 votes

-- Arjun Suresh (287k points)

4.9.3 Recursion: GATE1994_21 [top](#)

<http://gateoverflow.in/2517>



Selected Answer

Size of an activation record = $2 + 2 = 4$ bytes.

So, no. of possible activation records which can be live at a time = $64/4 = 16$.

So, we can have 16 function calls live at a time (recursion depth = 16), meaning the maximum value for n without stack overflow is 16 (calls from 1-16). For $n = 17$, stack will overflow.

This is different from the total no. of recursive calls which will be as follows:

n	No. of calls
1	1
2	3
3	5
4	9
5	15
6	25

13 votes

-- Arjun Suresh (287k points)

4.9.4 Recursion: GATE1995-2.9 [top](#)

<http://gateoverflow.in/261>

Selected Answer

C.

concat(a,head(tail(tail(acbc))))

concat(a,head(tail(cbc)))

concat(a,head(bc))

concat(a,b)

ab.

9 votes

-- Gate Keeda (18.8k points)

4.9.5 Recursion: GATE2000-16 [top](#)

<http://gateoverflow.in/687>

Selected Answer

Array f is used to store the fib() values calculated in order to save repeated calls. Since n = 0 and n = 1 are special cases we can store fib(2) to f[0], fib(3) to f[1] and so on. The missing code completed would be:

```
if (f[n - 2] != 0) {
    return f[n-2];
}
t = fib(n-1) + fib(n-2);
f[n-2] = t;
return t;
```

In this code, fib(i) will do a recursion only once as once fib(i) is calculated it is stored in array. So, the time complexity for fib(n) would be $\Theta(n)$.

PS: We can also store fib(n) in f(n), the above code just saves 2 elements' space in the array.

9 votes

-- Arjun Suresh (287k points)

4.9.6 Recursion: GATE2001-13 [top](#)

<http://gateoverflow.in/754>

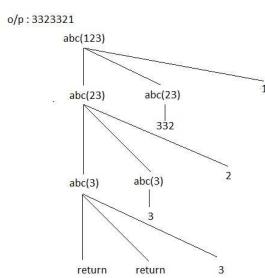
Selected Answer

answer a) 3323321

answer b) $2^n - 1$

Q. a) O/p :

3 3 2 3 3 2 1

**b)**

$$T(n) = 2 * T(n-1) + 1; n > 0$$

= 0 ; n=0 [Since for length zero string no character will be printed]

After solving it by substitution,

$$\begin{aligned} T(n) &= 2 * T(n-1) + 1 \\ &= 2 \times (2 \times T(n-2) + 1) + 1 \\ &= 2^2 \times T(n-2) + 2 + 1 \\ &= 2^2 \times (2 * T(n-3) + 1) + 2 + 1 \\ &= 2^3 \times T(n-3) + 2^2 + 2 + 1 \end{aligned}$$

Finally it will expand like

$$\begin{aligned} &= 2^n \times T(n-n) + 2^{n-1} + 2^{n-2} + \dots + 2^2 + 2 + 1 \\ &= 2^n \times T(0) + 2^{n-1} + 2^{n-2} + \dots + 2^2 + 2 + 1 \\ &= 1 \times (2^{n-1}) / (2 - 1) \\ &= 2^{n-1} \end{aligned}$$

10 votes

-- jayendra (7.7k points)

4.9.7 Recursion: GATE2002-11 [top](#)

<http://gateoverflow.in/864>

Selected Answer

```
void move(int n, char A, char B, char C) {
    if (n > 0) {
        move (n-1, A, C, B);
        printf("Move disk %d from pole %c to pole %c\n", n, A, C);
        move (n-1, B, A, C);
    }
}
```

9 votes

-- sonam vyas (12.8k points)

4.9.8 Recursion: GATE2004-31, ISRO2008-40 [top](#)

<http://gateoverflow.in/1028>

Selected Answer

answer is 7.as,

f(1):n=2,i=2

f(2):n=4,i=3

f(4):n=7,i=4

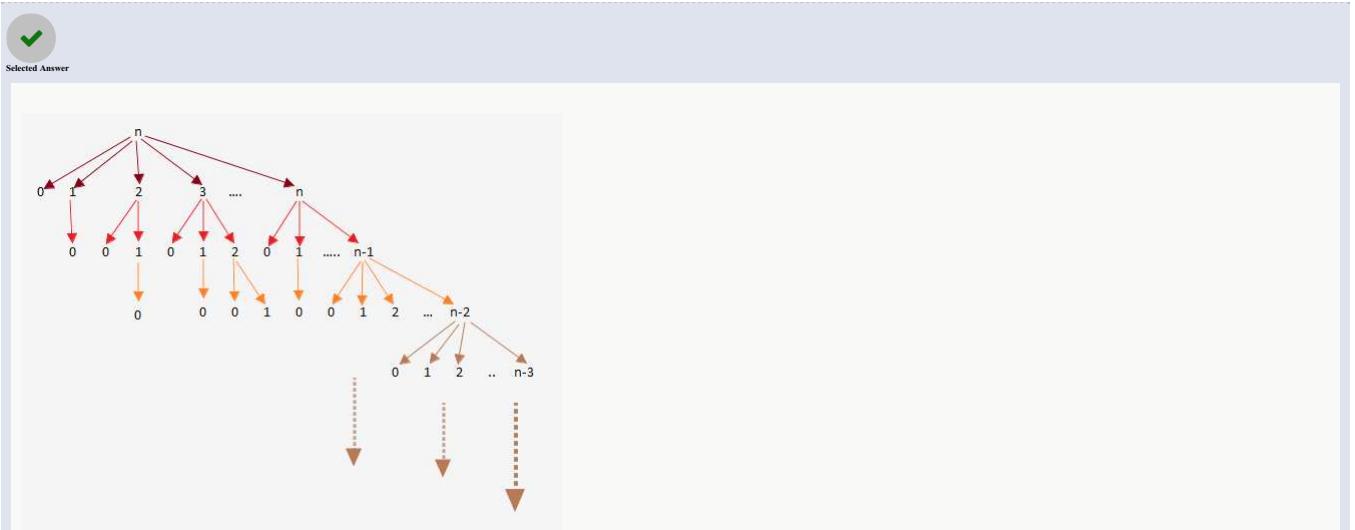
f(7):print(n)==>>> 7<ans>

12 votes

-- sumit kumar singh dixit (2.3k points)

4.9.9 Recursion: GATE2005-81a [top](#)

<http://gateoverflow.in/1403>



A. The code here is storing only local variables. So, the space complexity will be the recursion depth- maximum happening for the last iteration of the for loop- $\text{foo}(n-1) - \text{foo}(n-2) - \dots - \text{foo}(0)$ all live giving space complexity $O(n)$.

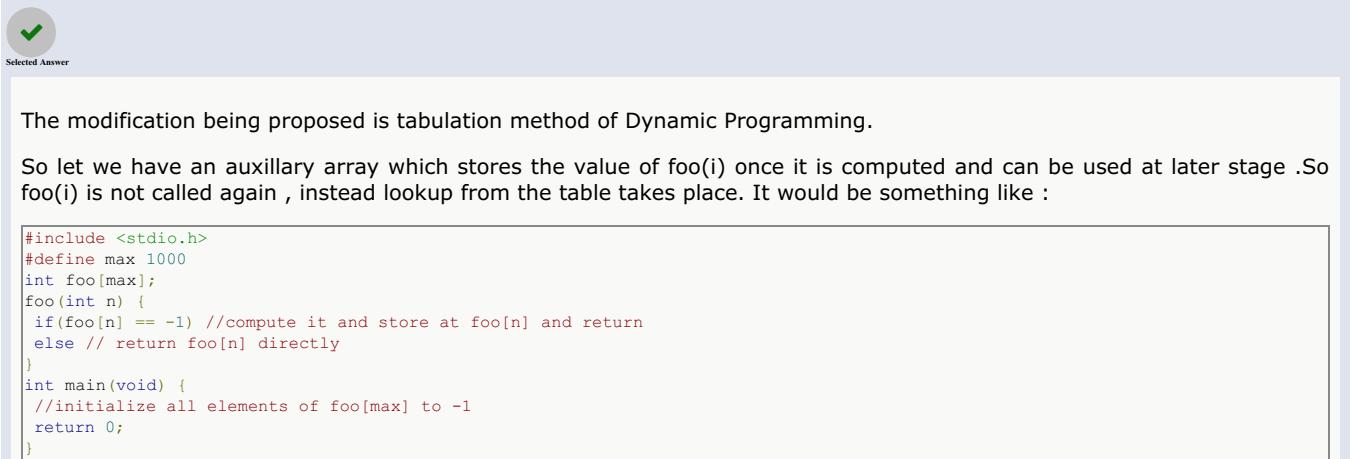
B. To store the n values we need space complexity $O(n)$. So, the space complexity won't change and will be $O(n)$.

18 votes

-- Arjun Suresh (287k points)

4.9.10 Recursion: GATE2005-81b [top](#)

<http://gateoverflow.in/82146>



Using this approach , we need an auxillary array of size n to store $\text{foo}(i)$ where i ranges from 0 to $n - 1$.

So space complexity of this method = $O(n)$ And function $\text{foo}(n)$ computes 2^{n-1}

Hence the correct option should be B)

11 votes

-- HABIB MOHAMMAD KHAN (66.6k points)

4.9.11 Recursion: GATE2007-42 [top](#)

<http://gateoverflow.in/1240>



$f(1) + 5 = 6 + 5 = 11$

$f(0) + 5 = 1+5 = 6$

Consider from last to first. Since it is recursive function.

10 votes

-- Gate Keeda (18.8k points)

4.9.12 Recursion: GATE2007-IT-27 [top](#)

<http://gateoverflow.in/3460>



The function terminates for all powers of 2 (which is infinite), hence (i) is false and (ii) is TRUE.

Let $n = 5$.

Now, recursive calls will go like 5 - 14 - 7 - 20 - 10 - 5 -

And this goes into infinite recursion. And if we multiply 5 with any power of 2, also result will be infinite recursion. Since, there are infinite powers of 2 possible, there are infinite recursions possible (even considering this case only). So, (iv) is TRUE and (iii) is false.

So, correct answer is (D)

16 votes

-- Arjun Suresh (287k points)

4.9.13 Recursion: GATE2014-2-40 [top](#)

<http://gateoverflow.in/2009>



(We can directly go to the "if" part to get one answer, but we need to solve "else" part too to get all possible answers which though is not asked in question)

Solving the else part:

$$\frac{x}{2} + \frac{3}{2x} = \frac{x^2+3}{2x}$$

So, the new value of x will be $\frac{x^2+3}{2x}$ and we need it equal to x .

$$\frac{x^2+3}{2x} = x \implies x^2 + 3 = 2x^2 \implies x^2 = 3 \implies x = 1.732$$

Now solving the if part.

```
abs(x*x - 3) < 0.01
```

So, $x^2 - 3 < 0.01$ and $- (x^2 - 3) < 0.01 \implies x^2 < 3.01$ and $x^2 > 2.99 \implies x < 1.735$ and $x > 1.729$

Corrected to 2 decimal places answer should be 1.73 or 1.74.

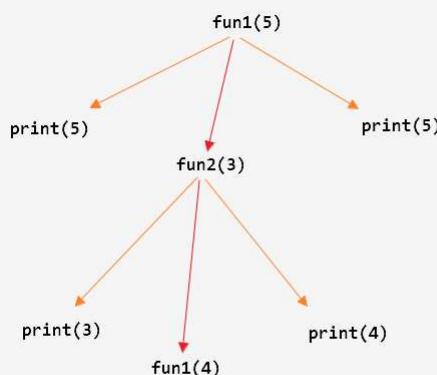
20 votes

-- Arjun Suresh (287k points)

4.9.14 Recursion: GATE2017-1-35 [top](#)

<http://gateoverflow.in/118317>





- Unroll recursion up to a point where we can distinguish the given options and choose the correct one!
- Options **B** and **D** are eliminated.
- **A** is the answer.

7 votes

-- Debashish Deka (49.1k points)

4.9.15 Recursion: Gate Sample Questions [top](#)

<http://gateoverflow.in/130912>



Selected Answer

don't use tree method of recursion when they asked embedded function call inside loop . let's do step by step

$$f(1) \Rightarrow 1$$

$$f(2) \Rightarrow x=1$$

$$1 + f(1)*f(1) = 2$$

$$f(3) \Rightarrow x=1$$

$$k=1 : 1 + f(1)*f(2) = 3$$

$$k=2 : 3 + f(2)*f(1) = 5$$

$$f(4) \Rightarrow x=1$$

$$k=1 : 1 + f(1)*f(3) = 6$$

$$k=2 : 6 + f(2)*f(2) = 10$$

$$k=3 : 10 + f(3)*f(1) = 15$$

$$f(5) \Rightarrow x=1$$

$$k=1 : 1 + f(1)*f(4) = 16$$

$$k=2 : 16 + f(2)*f(3) = 26$$

$$k=3 : 26 + f(3)*f(2) = 36$$

$$k=4 : 36 + f(4)*f(1) = 51$$

so 51

1 votes

-- pawan kumar (4.2k points)

4.9.16 Recursion: ISICAL MTech 2014 CS [top](#)

<http://gateoverflow.in/27885>



I tried it for 3^5 ,

in every call it prints "*" one time

and each function call 3 functions for $n/3$ recursively till get 1 at each node. It becomes a ternary tree

call for 3^5 -----1 = 3^0

call for 3^4 -----3 = 3^1

call for 3^3 -----9 = 3^2

call for 3^2 -----27 = 3^3

call for 3^1 -----81 = 3^4

call for 3^0 -----243 = 3^5

so one "*" prints at each function call

no of "*" printed = $3^0 + 3^1 + 3^2 + \dots + 3^k$ [that makes GP series]

$$= 1 * (1 - 3^{k+1})/(1-3)$$

$$= (3^{k+1} - 1)/2$$

[Note: I am not sure about it, feel free to edit.]

3 votes

-- Praveen Saini (53k points)

4.9.17 Recursion: TIFR2010-B-31 [top](#)

<http://gateoverflow.in/26484>

Answer is A) w and 0

If we solve using parallel innermost rule

$$F(1,0) = F(2,F(1,0)) * F(0,F(1,0))$$

$$= F(2, F(2,F(1,0)) * F(0,F(1,0))) * F(0, F(2,F(1,0)) * F(0,F(1,0)))$$

Since computation of $F(1,0)$ goes on

we assign $F(1,0)$ to w

So $F(1,0) = w$

Using parallel outermost rule

$$F(1,0) = F(2,F(1,0)) * F(0,F(1,0))$$

$$= F(2,F(1,0)) * 0$$

$$= 0$$

2 votes

-- zambus (253 points)

4.9.18 Recursion: TIFR2011-B-38 [top](#)

<http://gateoverflow.in/20923>



Selected Answer

answer = option E

Computable function: those which can be incorporated in a program using for/while loops.

Total Function: Defined for all possible inputs

Well Defined: if its definition assigns it a unique value.

It was a belief in early 1900s that every Computable function was also Primitively Recursive. But the discovery of Ackermann function provided a counter to it.

The class of primitive recursive functions is a small subclass of the class of recursive functions. This means that there are some functions which are Well-Defined Total Functions and are Computable BUT **Not** primitively recursive; eg. Ackermann function.

This makes all options from **option A** to **option D** as True.

But **option E** as

FALSE. As iterative programs are equivalent to only Primitively Recursive class.

5 votes

-- Amar Vashishth (27,9k points)

4.10

Runtime Environments(9) [top](#)

4.10.1 Runtime Environments: CMI2014-A-06 [top](#)

<http://gateoverflow.in/46972>

Suppose we are working with a programming language that supports automatic garbage collection. This means that:

- A. Uninitialized variables are assigned null values.
- B. Unreferenced dynamically allocated memory is added back to free space.
- C. Unreachable *if – then – else* branches are pruned.
- D. Expressions where array indices exceed array bounds are flagged.

[cmi2014](#) [programming](#) [runtime-environments](#)

Answer

4.10.2 Runtime Environments: GATE1990-2-v [top](#)

<http://gateoverflow.in/83980>

Match the pairs in the following questions:

- | | |
|-----------------------|---------------------------|
| (a) Pointer type | (p) Type conversion |
| (b) Activation record | (q) Dynamic structure |
| (c) Repeat-until | (r) Recursion |
| (d) Coercion | (s) Nondeterministic loop |

[gate1990](#) [match-the-following](#) [programming](#) [runtime-environments](#) [recursion](#)

Answer

4.10.3 Runtime Environments: GATE1991_09b [top](#)

<http://gateoverflow.in/43603>

For the following code, indicate the output if

- a. static scope rules
- b. dynamic scope rules

are used

```
var a,b : integer;

procedure P;
  a := 5;
  b := 10;
end {P};

procedure Q;
  var a, b : integer;
  P;
end {Q};
```

```

begin
  a := 1;
  b := 2;
  Q;
  Write ('a = ', a, 'b = ', b);
end

```

gate1991 runtime-environments normal

Answer

4.10.4 Runtime Environments: GATE1997-1.10 [top](#)

<http://gateoverflow.in/2226>

Heap allocation is required for languages.

- A. that support recursion
- B. that support dynamic data structure
- C. that use dynamic scope rules
- D. None of the above

gate1997 programming easy runtime-environments

Answer

4.10.5 Runtime Environments: GATE2003-73 [top](#)

<http://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 programming normal runtime-environments

Answer

4.10.6 Runtime Environments: GATE2003-74 [top](#)

<http://gateoverflow.in/43575>

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 dynamic scoping and call by name parameter passing mechanism, the values printed by the above program are

- A. 115, 220

- B. 25, 220
 C. 25, 15
 D. 115, 105

gate2003 programming runtime-environments normal

[Answer](#)

4.10.7 Runtime Environments: GATE2008-54 [top](#)

<http://gateoverflow.in/477>

Which of the following are true?

- I. A programming language which does not permit global variables of any kind and has no nesting of procedures/functions, but permits recursion can be implemented with static storage allocation
 - II. Multi-level access link (or display) arrangement is needed to arrange activation records only if the programming language being implemented has nesting of procedures/functions
 - III. Recursion in programming languages cannot be implemented with dynamic storage allocation
 - IV. Nesting procedures/functions and recursion require a dynamic heap allocation scheme and cannot be implemented with a stack-based allocation scheme for activation records
 - V. Programming languages which permit a function to return a function as its result cannot be implemented with a stack-based storage allocation scheme for activation records
- A. II and V only
 B. I, III and IV only
 C. I, II and V only
 D. II, III and V only

gate2008 programming difficult runtime-environments

[Answer](#)

4.10.8 Runtime Environments: GATE2010-14 [top](#)

<http://gateoverflow.in/218>

Which languages necessarily need heap allocation in the runtime environment?

- A. Those that support recursion.
 B. Those that use dynamic scoping.
 C. Those that allow dynamic data structure.
 D. Those that use global variables.

gate2010 programming easy runtime-environments

[Answer](#)

4.10.9 Runtime Environments: GATE2012_36 [top](#)

<http://gateoverflow.in/1758>

Consider the program given below, in a block-structured pseudo-language with lexical scoping and nesting of procedures permitted.

```
Program main;
  Var ...

  Procedure A1;
    Var ...
    Call A2;
  End A1

  Procedure A2;
    Var ...

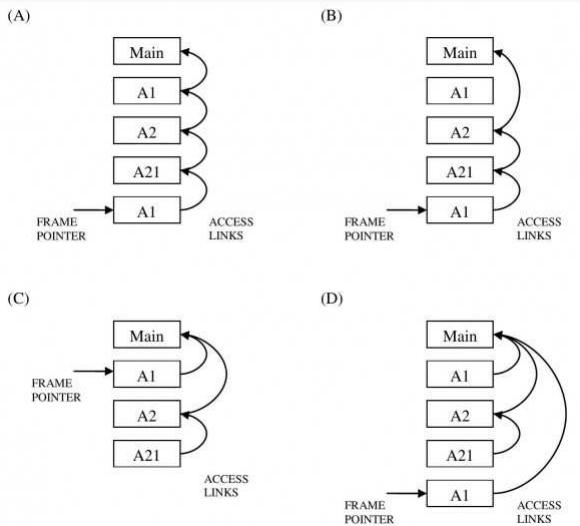
    Procedure A21;
      Var ...
      Call A1;
    End A21

    Call A21;
  End A2

  Call A1;
End main.
```

Consider the calling chain: Main -> A1 -> A2 -> A21 -> A1

The correct set of activation records along with their access links is given by



gate2012 programming runtime-environments normal

[Answer](#)

Answers: Runtime Environments

4.10.1 Runtime Environments: CMI2014-A-06 [top](#)



Selected Answer

Garbage collection also known as automatic memory management, is the automatic recycling of dynamically allocated memory.

Garbage collection is performed by a garbage collector which **recycles memory** that it can prove will **never be used again**.

Systems and languages which use garbage collection can be described as garbage-collected.

As long as address space is available in the memory, the runtime continues to allocate space for new objects. However, memory is not infinite. Eventually the garbage collector must perform a collection in order to free some memory. The garbage collector's optimizing engine determines the best time to perform a collection, based upon the allocations being made.

So option B is Correct.

Unreferenced dynamically allocated memory is added back to free space.

1 votes

-- Manoj Kumar (37k points)

4.10.2 Runtime Environments: GATE1990-2-v [top](#)



Selected Answer

- | | |
|----------------------|---------------------------|
| A. Pointer data type | Q. Dynamic Data Structure |
| B. Activation method | R. Recursion |
| C. Repeat until | S. Nondeterministic loop |
| D. Coercion | P. Type Conversion |

8 votes

-- Digvijay (46k points)

4.10.3 Runtime Environments: GATE1991_09b [top](#)

<http://gateoverflow.in/43603>



Selected Answer

In static scoping, if a variable is not found (variable definition - memory allocation) in the local scope, it is looked upon in global scope. In dynamic scoping, if a variable is not found in local scope, it is looked upon in the function which called the current executing one.

1. $a = 5, b = 10$. main is using global variables. P is also using global variables.

2. $a = 1, b = 2$. main is using global variables. P is using the local variables defined in Q.

(The modification in Q, happens to the variables in P but in main we use the global variables)

1 upvote

-- Arjun Suresh (287k points)

4.10.4 Runtime Environments: GATE1997-1.10 [top](#)

<http://gateoverflow.in/2226>



Selected Answer

memory is taken from heap for dynamic allocation

so option b is correct

1 upvote

-- Sankaranarayanan P.N (11k points)

4.10.5 Runtime Environments: GATE2003-73 [top](#)

<http://gateoverflow.in/960>



Selected Answer

Call-by-name : Is a lazy evaluation (expression passed as an argument evaluated only when used) technique.

Call-by-need: Is a version of call-by-name but when an expression is evaluated during first use, is saved and reused for all later uses.

```
global int i=100, j=5;
//memory created for i and j variable and 100 and 5 stored in them respectively (1)

void P(x) { // p(i+j) (3)
    int i=10;
    // new variable created i with value 10 store in it. (4)

    print(x+10);
    // print(x+10); = print(i+j +10);= 10 + 5 +10 = 25
    // here need of i+j so i+j replaced by 15 everywhere .(5)

    i=200;
    // local i value changed to 200.

    j=20;
    //global j value changed to 20 as there is no local j.
    //if dynamic scoping used, then the scope of j comes from scope of j in main as
    //main called this function. Here, main also uses global j and hence no change.

    print (x);
    // print (x);= print (i+j); = printf(15) = 15 (7)
    // due to call by need. If call by name used answer is 200+20 = 220
}

main() {P(i+j);}
// 1st function call since it is call by need no evaluation done
//here i and j value refer to global values(2)
```

Ans is C

Refer : <https://www.cs.bgu.ac.il/~comp161/wiki.files/ps9.pdf>

1 upvote

-- Prashant Singh (47.6k points)

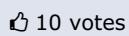
4.10.6 Runtime Environments: GATE2003-74 [top](#)

<http://gateoverflow.in/43575>



Selected Answer

```
global int i=100, j=5; // memory created for i and j variable and 100 and 5 store in it respectively (1)
void P(x) { // p(i+j) (3)
    int i=10; // new variable created i with value 10 store in it. (4)
    print(x+10); // print(x+10); = print(i+j +10);= 10 +5 +10 = 25 (5)
    i=200; // local i and global j value changed to 200 and 20 respectively (6)
    j=20;
    print (x); // print (x);= print (i +j); = 200 +20 = 220 (7)
}
main() {P(i+j);} // 1st function call since it is call by name dont calculate value just send i+j : here i and j value reffer to global values(2)
```

Ans is BReffer: <http://stackoverflow.com/questions/838079/what-is-pass-by-name-and-how-does-it-work-exactly>

10 votes

-- Prashant Singh (47.6k points)

4.10.7 Runtime Environments: GATE2008-54 [top](#)

<http://gateoverflow.in/477>

Answer is A.

I. Recursion can never be implemented with Static Storage Allocation.

II, Is TRUE.

III. Recursion can be implemented with Dynamic Storage Allocation but not with Static Storage Allocation.

IV. Can be done with Stack based allocation scheme.

V. Is TRUE as with a stack based allocation once a function returns its activation record is no longer valid- so we cannot return a function as a result.



10 votes

-- Gate Keeda (18.8k points)

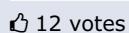
4.10.8 Runtime Environments: GATE2010-14 [top](#)

<http://gateoverflow.in/2187>

Selected Answer

Those that allow dynamic data structure.

malloc etc uses memory from heap area



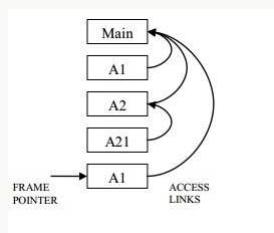
12 votes

-- Sankaranarayanan P.N (11k points)

4.10.9 Runtime Environments: GATE2012_36 [top](#)

<http://gateoverflow.in/1758>

Selected Answer



Since, Activation records are created at procedure entry time and destroyed at procedure exit time.
therefore here Calling sequence is given as ,
Main->A1->A2->A21->A1

now A1 and A2 are defined under Main...So A1,A2 Access link are pointed to main
A21 is Defined under A2 hence its Access link will point to A2

22 votes

-- Kalpish Singhal (2k points)

4.11

Type Checking(1) top

4.11.1 Type Checking: GATE2003-24 top

<http://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

Answer

Answers: Type Checking

4.11.1 Type Checking: GATE2003-24 top

<http://gateoverflow.in/914>



Selected Answer

Answer is C. In dynamically typed languages variables do have type- just that it is inferred during runtime.

5 votes

-- Arjun Suresh (287k points)

5

Theory of Computation (268) [top](#)

5.1

Closure Property(9) [top](#)

5.1.1 Closure Property: GATE1989-3-ii [top](#)

<http://gateoverflow.in/87117>

Context-free languages and regular languages are both closed under the operation (s) of :

- A. Union
- B. Intersection
- C. Concatenation
- D. Complementation

[gate1989](#) [easy](#) [theory-of-computation](#) [closure-property](#)[Answer](#)

5.1.2 Closure Property: GATE1992-16 [top](#)

<http://gateoverflow.in/595>

Which of the following three statements are true? Prove your answer.

- (i) The union of two recursive languages is recursive.
- (ii) The language $\{O^n \mid n \text{ is a prime}\}$ is not regular.
- (iii) Regular languages are closed under infinite union.

[gate1992](#) [theory-of-computation](#) [normal](#) [closure-property](#)[Answer](#)

5.1.3 Closure Property: GATE2002-2.14 [top](#)

<http://gateoverflow.in/844>

Which of the following is true?

- A. The complement of a recursive language is recursive
- B. The complement of a recursively enumerable language is recursively enumerable
- C. The complement of a recursive language is either recursive or recursively enumerable
- D. The complement of a context-free language is context-free

[gate2002](#) [theory-of-computation](#) [easy](#) [closure-property](#)[Answer](#)

5.1.4 Closure Property: GATE2006-IT-32 [top](#)

<http://gateoverflow.in/357>

Let L be a context-free language and M a regular language. Then the language $L \cap M$ is

- A. always regular
- B. never regular
- C. always a deterministic context-free language
- D. always a context-free language

[gate2006-it](#) [theory-of-computation](#) [closure-property](#) [easy](#)[Answer](#)

5.1.5 Closure Property: GATE2013-17 [top](#)

<http://gateoverflow.in/1439>

Which of the following statements is/are **FALSE**?

1. For every non-deterministic Turing machine, there exists an equivalent deterministic Turing machine.
2. Turing recognizable languages are closed under union and complementation.
3. Turing decidable languages are closed under intersection and complementation.
4. Turing recognizable languages are closed under union and intersection.

- (A) 1 and 4 only (B) 1 and 3 only (C) 2 only (D) 3 only

gate2013 theory-of-computation normal closure-property

[Answer](#)

5.1.6 Closure Property: GATE2016-2-18 [top](#)

<http://gateoverflow.in/39574>

Consider the following types of languages: L_1 : Regular, L_2 : Context-free, L_3 : Recursive, L_4 : Recursively enumerable. Which of the following is/are **TRUE** ?

I. $\bar{L}_3 \cup L_4$ is recursively enumerable.

II. $\bar{L}_2 \cup L_3$ is recursive.

III. $L_1^* \cap L_2$ is context-free.

IV. $L_1 \cup \bar{L}_2$ is context-free.

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

gate2016-2 theory-of-computation regular-languages context-free-language closure-property normal

[Answer](#)

5.1.7 Closure Property: GATE2017-2-04 [top](#)

<http://gateoverflow.in/118143>

Let L_1, L_2 be any two context-free languages and R be any regular language. Then which of the following is/are CORRECT?

I. $\underline{L_1} \cup L_2$ is context-free

II. $\underline{L_1}$ is context-free

III. $L_1 - R$ is context-free

IV. $L_1 \cap L_2$ is context-free

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

gate2017-2 theory-of-computation closure-property

[Answer](#)

5.1.8 Closure Property: TIFR2013-B-11 [top](#)

<http://gateoverflow.in/25772>

Which of the following statements is FALSE?

- a. The intersection of a context free language with a regular language is context free.
- b. The intersection of two regular languages is regular.
- c. The intersection of two context free languages is context free
- d. The intersection of a context free language and the complement of a regular language is context free.
- e. The intersection of a regular language and the complement of a regular language is regular.

tifr2013 theory-of-computation closure-property

[Answer](#)

5.1.9 Closure Property: TIFR2014-B-14 [top](#)

<http://gateoverflow.in/27321>

Which the following is FALSE?

- a. Complement of a recursive language is recursive.
- b. A language recognized by a non-deterministic Turing machine can also be recognized by a deterministic Turing machine.

- c. Complement of a context free language can be recognized by a Turing machine.
 d. If a language and its complement are both recursively enumerable then it is recursive.
 e. Complement of a non-recursive language can never be recognized by any Turing machine.

[tifr2014](#) | [theory-of-computation](#) | [closure-property](#)

[Answer](#)

Answers: Closure Property

5.1.1 Closure Property: GATE1989-3-ii [top](#)



Selected Answer

Regular language is closed under **Union , Intersection, Concatenation, Complementation**.

CFL is closed under **Union Concatenation but not under Intersection or Complementation**

So Both closed under Union Concatenation.

5 votes

-- Prashant Singh (47.6k points)

5.1.2 Closure Property: GATE1992-16 [top](#)



Selected Answer

(i) True. Recursive languages are closed under union.

(ii) True. The language is Turing Machine acceptable.

(iii) False. Regular languages are closed under finite union.

5 votes

-- Rajarshi Sarkar (34.4k points)

5.1.3 Closure Property: GATE2002-2.14 [top](#)



Selected Answer

Complement of recursive language is always recursive

7 votes

-- Bhagirathi Nayak (13.1k points)

5.1.4 Closure Property: GATE2006-IT-32 [top](#)



Selected Answer

Let $\Sigma = \{a, b\}$

Let $L_1 = \Sigma^*$ be a regular language

Let $L_2 = \{ww^R \mid w \in (a+b)^*\}$ be a context free language.

$L_1 \cap L_2 = \{ww^R \mid w \in (a+b)^*\}$ which is clearly context free language and not DCFL or Regular
 Hence the answer is option D

9 votes

-- Mari Ganesh Kumar (2.2k points)

5.1.5 Closure Property: GATE2013-17 [top](#)



Selected Answer

Recursive enumerable languages are not closed under complement . while recursive languages are.

both Recursive and Recursive enumerable languages are closed under intersection, union, and kleene star.

http://gatecse.in/wiki/Closure_Property_of_Language_Families

Non-Deterministic TM is equivalent to DTM

only 2 is false. Option C is correct.

Note: Turing decidable language mean Recursive language and Turing recognizable language mean recursive enumerable language.

14 votes

-- Praveen Saini (53k points)

5.1.6 Closure Property: GATE2016-2-18 [top](#)

<http://gateoverflow.in/39574>



Selected Answer

I.
 $\overline{L_3} \cup L_4$

L_3 is recursive, so

$\overline{L_3}$ is also recursive (closed under complement),

So,

$\overline{L_3}$ is recursive enumerable.

L_4 is recursive enumerable,

so,

$\overline{L_3} \cup L_4$ is also recursive enumerable (closed under union).

II.
 $\overline{L_2} \cup L_3$

L_2 is Context-free, so

$\overline{L_2}$, may or may not be Context-free (not closed under complement), but definitely

$\overline{L_2}$ is Recursive.

L_3 is recursive.

so

$\overline{L_2} \cup L_3$ is also recursive (closed under union).

III.
 $L_1^* \cap L_2$

L_1 is Regular, so

L_1^* is also regular (closed under kleene star)

L_2 is Context-free

so,

$L_1^* \cap L_2$ is also context-free (closed under intersection with regular).

IV.
 $L_1 \cup \overline{L_2}$

L_1 is regular.

L_2 is context-free, so

$\overline{L_2}$ may or may not be Context-free (not closed under complement).

so,
 $L_1 \cup \overline{L_2}$ may or may not be Context-free.

Here answer is D

29 votes

-- Praveen Saini (53k points)

5.1.7 Closure Property: GATE2017-2-04 [top](#)

<http://gateoverflow.in/118143>



Selected Answer

Q.11 Let L_1, L_2 be any two context-free languages and R be any regular language. Then which of the following is/are CORRECT?

- I. $L_1 \cup L_2$ is context-free
- II. $\overline{L_1}$ is context-free
- III. $L_1 \rightarrow R$ is context-free
- IV. $L_1 \cap L_2$ is context-free

(A) I, II and IV only (B) I and III only (C) II and IV only (D) I only

Question ID : 2178752729

Status : Answered

Chosen Option : 2

Options

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

this is correct Q,

intersection of two CFL are not closed operation, and CFL also not closed under complimentation,

CFL-reg we can write like this CFL $CFL \cap \text{regular}$ so regular language closed under complimentation and intersection of CFL with regular is always CFL.

2 votes

-- 2018 (4.9k points)

5.1.8 Closure Property: TIFR2013-B-11 [top](#)

<http://gateoverflow.in/25772>



Selected Answer

Context language are not closed under intersection so option C

6 votes

-- Umang Raman (14.5k points)

5.1.9 Closure Property: TIFR2014-B-14 [top](#)

<http://gateoverflow.in/27321>



Selected Answer

A) True. For a recursive language, we have a TM which says for any string "w", "yes" if it belongs to the language and "no" if it does not. For the complement of a recursive language we just have to reverse 'yes' and 'no' conditions and this is possible with a slight modification to the original TM. So, the new language also recursive

B) True. Non-determinism does not add any recognizing power to a Turing machine though it can affect the time complexity in solving a problem.

C) True. Complement of CFL is recursive. and recursive language is recognized by Turing machine. So, complement of language can be recognized by TM

D) True. If a language is r.e., we have a TM which says "yes" whenever a given string belongs to L. Similarly, if its complement is r.e., we have a TM which says "yes", whenever a string does not belong to the language. So, combining both we can always say "yes" if a string belongs to the language and "no" if a string does not belong to L => L is recursive.

E) False.

Non-recursive Language means:- a) RE or b) Non RE

Now **Non RE complement** may be RE which **Can be recognized by TM...so option E is false.**

Example:-> $L = \{ \langle M, w \rangle \mid M \text{ is a TM and it does not halt on string } w \}.$ //Non RE (complement of halting problem)

Now $L^c = \{ \langle M, w \rangle \mid M \text{ is a TM and it halts on string } w \}.$ //RE but not Recursive //recognized by TM (halting problem)

So answer is (E)

4 votes

-- srestha (55.2k points)

5.2**Context Free Language(32) [top](#)****5.2.1 Context Free Language: GATE 2016-1-16 [top](#)**<http://gateoverflow.in/39640>

Which of the following languages is generated by the given grammar?

$$S \rightarrow aS \mid bS \mid \epsilon$$

- A. $\{a^n b^m \mid n, m \geq 0\}$
- B. $\{w \in \{a, b\}^* \mid w \text{ has equal number of } a's \text{ and } b's\}$
- C. $\{a^n \mid n \geq 0\} \cup \{b^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}$
- D. $\{a, b\}^*$

[gate2016-1](#) | [theory-of-computation](#) | [context-free-language](#) | [normal](#)

[Answer](#)

5.2.2 Context Free Language: GATE 2016-1-42 [top](#)<http://gateoverflow.in/39705>

Consider the following context-free grammars;

$$G_1 : S \rightarrow aS \mid B, B \rightarrow b \mid bB$$

$$G_2 : S \rightarrow aA \mid bB, A \rightarrow aA \mid B \mid \epsilon, B \rightarrow bB \mid \epsilon$$

Which one of the following pairs of languages is generated by G_1 and G_2 , respectively?

- A. $\{a^m b^n \mid m > 0 \text{ or } n > 0\}$ and $\{a^m b^n \mid m > 0 \text{ and } n > 0\}$
- B. $\{a^m b^n \mid m > 0 \text{ and } n > 0\}$ and $\{a^m b^n \mid m > 0 \text{ or } n \geq 0\}$
- C. $\{a^m b^n \mid m \geq 0 \text{ or } n > 0\}$ and $\{a^m b^n \mid m > 0 \text{ and } n > 0\}$
- D. $\{a^m b^n \mid m \geq 0 \text{ and } n > 0\}$ and $\{a^m b^n \mid m > 0 \text{ or } n > 0\}$

[gate2016-1](#) | [theory-of-computation](#) | [context-free-language](#) | [normal](#)

[Answer](#)

5.2.3 Context Free Language: GATE 2016-2-43 [top](#)<http://gateoverflow.in/39605>

Consider the following languages:

$$L_1 = \{a^n b^n c^{n+m} : m, n \geq 1\}$$

$$L_2 = \{a^n b^n c^{2n} : n \geq 1\}$$

Which one of the following is TRUE?

- A. Both L_1 and L_2 are context-free.
- B. L_1 is context-free while L_2 is not context-free.
- C. L_2 is context-free while L_1 is not context-free.
- D. Neither L_1 nor L_2 is context-free.

[gate2016-2](#) | [theory-of-computation](#) | [context-free-language](#) | [normal](#)

[Answer](#)

5.2.4 Context Free Language: GATE1987-1-xii [top](#)<http://gateoverflow.in/80281>

A context-free grammar is ambiguous if

- A. The grammar contains useless non-terminals.
- B. It produces more than one parse tree for some sentence.
- C. Some production has two non terminals side by side on the right-hand side.
- D. None of the above.

gate1987 theory-of-computation context-free-language ambiguous

Answer

5.2.5 Context Free Language: GATE1987-2k [top](#)

<http://gateoverflow.in/8059>

State whether the following statements are TRUE or FALSE:

The intersection of two CFL's is also a CFL.

gate1987 theory-of-computation context-free-language

Answer

5.2.6 Context Free Language: GATE1992_02,xix [top](#)

<http://gateoverflow.in/572>

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

(xix) Context-free languages are

- a. closed under union
- b. closed under complementation
- c. closed under intersection
- d. closed under Kleene closure

gate1992 context-free-language theory-of-computation normal

Answer

5.2.7 Context Free Language: GATE1992_02,xviii [top](#)

<http://gateoverflow.in/576>

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

If G is a context free grammar and w is a string of length l in $L(G)$, how long is a derivation of w in G , if G is in Chomsky normal form?

- (a). $2l$
- (b). $2l + 1$
- (c). $2l - 1$
- (d). l

gate1992 theory-of-computation context-free-language easy

Answer

5.2.8 Context Free Language: GATE1995-2.20 [top](#)

<http://gateoverflow.in/2632>

Which of the following definitions below generate the same language as L , where $L = \{x^n y^n \text{ such that } n \geq 1\}$?

- I. $E \rightarrow xEy \mid xy$
- II. $xy \mid (x^+ xyy^+)$
- III. $x^+ y^+$
 - A. I only
 - B. I and II
 - C. II and III
 - D. II only

gate1995 theory-of-computation easy context-free-language

Answer**5.2.9 Context Free Language: GATE1996_2.8** [top](#)<http://gateoverflow.in/2737>

If L_1 and L_2 are context free languages and R a regular set, one of the languages below is not necessarily a context free language. Which one?

- A. $L_1 \cdot L_2$
- B. $L_1 \cap L_2$
- C. $L_1 \cap R$
- D. $L_1 \cup L_2$

[gate1996](#) [theory-of-computation](#) [context-free-language](#) [easy](#)
Answer**5.2.10 Context Free Language: GATE1996_2.9** [top](#)<http://gateoverflow.in/2738>

Define a context free languages $L \in \{0,1\}^*$, $\text{init}(L) = \{u \mid uv \in L \text{ for some } v \text{ in } \{0,1\}^*\}$ (in other words, $\text{init}(L)$ is the set of prefixes of L)

Let $L = \{w \mid w \text{ is nonempty and has an equal number of 0's and 1's}\}$

Then $\text{init}(L)$ is

- A. the set of all binary strings with unequal number of 0's and 1's
- B. the set of all binary strings including null string
- C. the set of all binary strings with exactly one more 0 than the number of 1's or one more 1 than the number of 0's
- D. None of the above

[gate1996](#) [theory-of-computation](#) [context-free-language](#) [normal](#)
Answer**5.2.11 Context Free Language: GATE1999_1.5** [top](#)<http://gateoverflow.in/1459>

Context-free languages are closed under:

- A. Union, intersection
- B. Union, Kleene closure
- C. Intersection, complement
- D. Complement, Kleene closure

[gate1999](#) [theory-of-computation](#) [context-free-language](#) [easy](#)
Answer**5.2.12 Context Free Language: GATE1999_7** [top](#)<http://gateoverflow.in/1506>

Show that the language

$$L = \{x c x \mid x \in \{0,1\}^* \text{ and } c \text{ is a terminal symbol}\}$$

is not context free. c is not 0 or 1.

[gate1999](#) [theory-of-computation](#) [context-free-language](#) [normal](#)
Answer

5.2.13 Context Free Language: GATE2000-7 [top](#)

<http://gateoverflow.in/678>

- Construct as minimal finite state machine that accepts the language, over $\{0,1\}$, of all strings that contain neither the sub string 00 nor the sub string 11.
- Consider the grammar

```

S → aSAb
S → ε
A → bA
A → ε
  
```

where S, A are non-terminal symbols with S being the start symbol; a, b are terminal symbols and ϵ is the empty string. This grammar generates strings of the form $a^i b^j$ for some $i, j \geq 0$, where i and j satisfy some condition. What is the condition on the values of i and j?

[gate2000](#) [theory-of-computation](#) [descriptive](#) [regular-languages](#) [context-free-language](#)

[Answer](#)

5.2.14 Context Free Language: GATE2001-1.5 [top](#)

<http://gateoverflow.in/698>

Which of the following statements is true?

- If a language is context free it can always be accepted by a deterministic push-down automaton
- The union of two context free languages is context free
- The intersection of two context free languages is a context free
- The complement of a context free language is a context free

[gate2001](#) [theory-of-computation](#) [context-free-language](#) [easy](#)

[Answer](#)

5.2.15 Context Free Language: GATE2003-51 [top](#)

<http://gateoverflow.in/940>

Let $G = (\{S\}, \{a,b\}, R, S)$ be a context free grammar where the rule set R is $S \rightarrow aSb \mid SS \mid \epsilon$

Which of the following statements is true?

- G is not ambiguous
- There exist $x, y \in L(G)$ such that $xy \notin L(G)$
- There is a deterministic pushdown automaton that accepts $L(G)$
- We can find a deterministic finite state automaton that accepts $L(G)$

[gate2003](#) [theory-of-computation](#) [context-free-language](#) [normal](#)

[Answer](#)

5.2.16 Context Free Language: GATE2005-57 [top](#)

<http://gateoverflow.in/1380>

Consider the languages:

- $L_1 = \{ww^R \mid w \in \{0,1\}^*\}$
- $L_2 = \{w\#w^R \mid w \in \{0,1\}^*\}$, where # is a special symbol
- $L_3 = \{ww \mid w \in \{0,1\}^*\}$

Which one of the following is TRUE?

- L_1 is a deterministic CFL
- L_2 is a deterministic CFL
- L_3 is a CFL, but not a deterministic CFL
- L_3 is a deterministic CFL

[gate2005](#) [theory-of-computation](#) [context-free-language](#) [easy](#)

[Answer](#)

5.2.17 Context Free Language: GATE2006-19 [top](#)

<http://gateoverflow.in/980>

Let

$$\begin{aligned}L_1 &= \{0^{n+m}1^n0^m \mid n,m \geq 0\}, \\L_2 &= \{0^{n+m}1^{n+m}0^m \mid n,m \geq 0\} \text{ and} \\L_3 &= \{0^{n+m}1^{n+m}0^{n+m} \mid n,m \geq 0\}.\end{aligned}$$

Which of these languages are NOT context free?

- A. L_1 only
- B. L_3 only
- C. L_1 and L_2
- D. L_2 and L_3

[gate2006](#) [theory-of-computation](#) [context-free-language](#) [normal](#)

[Answer](#)

5.2.18 Context Free Language: GATE2006-IT-34 [top](#)

<http://gateoverflow.in/3573>

In the context-free grammar below, S is the start symbol, a and b are terminals, and ϵ denotes the empty string.

$$\begin{aligned}S &\rightarrow aSAb \mid \epsilon \\A &\rightarrow bA \mid \epsilon\end{aligned}$$

The grammar generates the language

- A. $((a + b)^* b)$
- B. $(a^m b^n \mid m \leq n)$
- C. $(a^m b^n \mid m = n)$
- D. $a^* b^*$

[gate2006-it](#) [theory-of-computation](#) [context-free-language](#) [normal](#)

[Answer](#)

5.2.19 Context Free Language: GATE2006-IT-4 [top](#)

<http://gateoverflow.in/3543>

In the context-free grammar below, S is the start symbol, a and b are terminals, and ϵ denotes the empty string

$$S \rightarrow aSa \mid bSb \mid a \mid b \mid \epsilon$$

Which of the following strings is NOT generated by the grammar?

- A. aaaa
- B. baba
- C. abba
- D. babaabab

[gate2006-it](#) [theory-of-computation](#) [context-free-language](#) [easy](#)

[Answer](#)

5.2.20 Context Free Language: GATE2007-IT-46 [top](#)

<http://gateoverflow.in/3481>

The two grammars given below generate a language over the alphabet $\{x, y, z\}$

$$\begin{array}{ll}G1 : & S \rightarrow x \mid z \mid xS \mid zS \mid yB \\ & B \rightarrow y \mid z \mid yB \mid zB \\G2 : & S \rightarrow y \mid z \mid yS \mid zS \mid xB \\ & B \rightarrow y \mid yS\end{array}$$

Which one of the following choices describes the properties satisfied by the strings in these languages?

- A. G1 : No y appears before any x
G2 : Every x is followed by at least one y
- B. G1 : No y appears before any x
G2 : No x appears before any y

- C. G1 : No y appears after any x
 G2 : Every x is followed by at least one y
 D. G1 : No y appears after any x
 G2 : Every y is followed by at least one x

[gate2007-it](#) [theory-of-computation](#) [normal](#) [context-free-language](#)

[Answer](#)

5.2.21 Context Free Language: GATE2007-IT-48 [top](#)

<http://gateoverflow.in/3490>

Consider the grammar given below

$$\begin{array}{l} S \rightarrow xB \mid yA \\ A \rightarrow x \mid xS \mid yAA \\ B \rightarrow y \mid yS \mid xBB \end{array}$$

Consider the following strings.

- i. xxxyyx
- ii. xxxyxy
- iii. xyxy
- iv. yxxxy
- v. yxx
- vi. xyx

Which of the above strings are generated by the grammar ?

- A. i, ii and iii
- B. ii, v and vi
- C. ii, iii and iv
- D. i, iii and iv

[gate2007-it](#) [theory-of-computation](#) [context-free-language](#) [normal](#)

[Answer](#)

5.2.22 Context Free Language: GATE2007-IT-49 [top](#)

<http://gateoverflow.in/3491>

Consider the following grammars. Names representing terminals have been specified in capital letters.

$$\begin{array}{ll} \text{G1} & \text{stmt} \rightarrow \text{WHILE } (\text{expr}) \\ : & \text{stmt} \rightarrow \text{stmnt} \\ & \text{stmt} \rightarrow \text{OTHER} \\ & \text{expr} \rightarrow \text{ID} \\ \text{G2} & \text{stmt} \rightarrow \text{WHILE } (\text{expr}) \\ : & \text{stmt} \rightarrow \text{stmnt} \\ & \text{stmt} \rightarrow \text{OTHER} \\ & \text{expr} \rightarrow \text{expr} + \text{expr} \\ & \text{expr} \rightarrow \text{expr} * \text{expr} \\ & \text{expr} \rightarrow \text{ID} \end{array}$$

Which one of the following statements is true?

- A. G₁ is context-free but not regular and G₂ is regular
- B. G₂ is context-free but not regular and G₁ is regular
- C. Both G₁ and G₂ are regular
- D. Both G₁ and G₂ are context-free but neither of them is regular

[gate2007-it](#) [theory-of-computation](#) [context-free-language](#) [normal](#)

[Answer](#)

5.2.23 Context Free Language: GATE2008-IT-34 [top](#)

<http://gateoverflow.in/3344>

Consider a CFG with the following productions.

$$\begin{array}{l} S \rightarrow AA \mid B \\ A \rightarrow 0A \mid A0 \mid 1 \\ B \rightarrow 0B00 \mid 1 \end{array}$$

S is the start symbol, A and B are non-terminals and 0 and 1 are the terminals. The language generated by this grammar is

- A. $\{0^n 1 0^{2n} \mid n \geq 1\}$
- B. $\{0^i 1 0^j 1 0^k \mid i, j, k \geq 0\} \cup \{0^n 1 0^{2n} \mid n \geq 0\}$
- C. $\{0^i 1 0^j \mid i, j \geq 0\} \cup \{0^n 1 0^{2n} \mid n \geq 0\}$
- D. The set of all strings over $\{0, 1\}$ containing at least two 0's

gate2008-it theory-of-computation context-free-language normal

Answer

5.2.24 Context Free Language: GATE2008-IT-78 [top](#)

<http://gateoverflow.in/3392>

A CFG G is given with the following productions where S is the start symbol, A is a non-terminal and a and b are terminals.

$$S \rightarrow aS \mid AA \rightarrow aAb \mid bAa \mid \epsilon$$

Which of the following strings is generated by the grammar above?

- A. aabbaba
- B. aabaaba
- C. abababb
- D. aabbaab

gate2008-it theory-of-computation normal context-free-language

Answer

5.2.25 Context Free Language: GATE2009-12, ISRO2016-37 [top](#)

<http://gateoverflow.in/1304>

$$S \rightarrow aSa \mid bSb \mid a \mid b$$

The language generated by the above grammar over the alphabet $\{a, b\}$ is the set of

- A. all palindromes
- B. all odd length palindromes
- C. strings that begin and end with the same symbol
- D. all even length palindromes

gate2009 theory-of-computation context-free-language easy isro2016

Answer

5.2.26 Context Free Language: GATE2015-3_32 [top](#)

<http://gateoverflow.in/8489>

Which of the following languages are context-free?

$$L_1 : \{a^m b^n a^n b^m \mid m, n \geq 1\}$$

$$L_2 : \{a^m b^n a^m b^n \mid m, n \geq 1\}$$

$$L_3 : \{a^m b^n \mid m = 2n + 1\}$$

- A. L_1 and L_2 only
- B. L_1 and L_3 only
- C. L_2 and L_3 only
- D. L_3 only

gate2015-3 theory-of-computation context-free-language normal

Answer

5.2.27 Context Free Language: GATE2017-1-10 [top](#)

<http://gateoverflow.in/118290>

Consider the following context-free grammar over the alphabet $\Sigma = \{a, b, c\}$ with S as the start symbol:

$$S \rightarrow abScT \mid abcT$$

$$T \rightarrow bT \mid b$$

Which one of the following represents the language generated by the above grammar?

- (A) $\{(ab)^n(cb)^n \mid n \geq 1\}$
- (B) $\{(ab)^n cb^{m_1} cb^{m_2} \dots cb^{m_n} \mid n, m_1, m_2, \dots, m_n \geq 1\}$
- (C) $\{(ab)^n(cb^m)^n \mid m, n \geq 1\}$
- (D) $\{(ab)^n(cb^n)^m \mid m, n \geq 1\}$

gate2017-1 theory-of-computation context-free-language normal

[Answer](#)

5.2.28 Context Free Language: GATE2017-1-34 [top](#)

<http://gateoverflow.in/118316>

If G is a grammar with productions

$$S \rightarrow SaS | aSb | bSa | SS | \in$$

where S is the start variable, then which one of the following strings is not generated by G ?

- (A) $abab$
- (B) $aaab$
- (C) $abbaa$
- (D) $babba$

gate2017-1 theory-of-computation context-free-language normal

[Answer](#)

5.2.29 Context Free Language: GATE2017-1-38 [top](#)

<http://gateoverflow.in/118321>

Consider the following languages over the alphabet $\Sigma = \{a, b, c\}$. Let $L_1 = \{a^n b^n c^m \mid m, n \geq 0\}$ and $L_2 = \{a^m b^n c^n \mid m, n \geq 0\}$.

Which of the following are context-free languages?

- I. $L_1 \cup L_2$
- II. $L_1 \cap L_2$
- (A) I only
- (B) II only
- (C) I and II
- (D) Neither I nor II

gate2017-1 theory-of-computation context-free-language normal

[Answer](#)

5.2.30 Context Free Language: GATE2017-2-16 [top](#)

<http://gateoverflow.in/118243>

Identify the language generated by the following grammar, where S is the start variable.

- $S \rightarrow XY$
 - $X \rightarrow aX \mid a$
 - $Y \rightarrow aYb \mid \epsilon$
- A. $\{a^m b^n \mid m \geq n, n > 0\}$
 B. $\{a^m b^n \mid m \geq n, n \geq 0\}$
 C. $\{a^m b^n \mid m > n, n \geq 0\}$
 D. $\{a^m b^n \mid m > n, n > 0\}$

gate2017-2 theory-of-computation context-free-language

[Answer](#)

5.2.31 Context Free Language: ISI2013-CS-4a [top](#)

<http://gateoverflow.in/47637>

Give a context-free grammar G that generates $L = \{0^i 1^j 0^k \mid i + k = j\}$. Prove that $L = L(G)$.

descriptive | isi2013 | context-free-language | grammar | theory-of-computation

Answer

5.2.32 Context Free Language: ISI2015-CS-5b [top](#)

<http://gateoverflow.in/47339>

Construct a context free grammar (CFG) to generate the following language:

$L = \{a^n b^m c^{n+m} : n, m \text{ are integers, and } n \geq 1, m \geq 1\}$

descriptive | isi2015 | theory-of-computation | context-free-language

Answer

Answers: Context Free Language

5.2.1 Context Free Language: GATE 2016-1-16 [top](#)

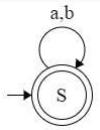
<http://gateoverflow.in/39640>



Selected Answer

$$S \rightarrow aS \mid bS \mid \epsilon$$

is $(a+b)^*$



20 votes

-- Praveen Saini (53k points)

5.2.2 Context Free Language: GATE 2016-1-42 [top](#)

<http://gateoverflow.in/39705>



Selected Answer

Ans (D)

G1 results in strings b, ab, bb, aab, abb, bbb, ... i.e $a^m b^n$, $m \geq 0$ and $n > 0$ (and because only a's are not possible but only b's are)

G2 result in strings a, b, aa, ab, bb, aaa, aab, abb, bbb ... i.e $a^m b^n$, $m > 0$ or $n > 0$ (or because only b's is possible b,bb,bbb, , only a's are possible)

20 votes

-- juxtapose (519 points)

5.2.3 Context Free Language: GATE 2016-2-43 [top](#)

<http://gateoverflow.in/39605>



Selected Answer

$$L_1 = \{a^n b^m c^{n+m} : m, n \geq 1\} \text{ is Context-free language}$$

(push $a's$ into stack, then push $b's$ into stack, read $c's$ and pop $b's$, when no $b's$ left on stack, keep reading $c's$ and pop $a's$, when no $c's$ left in input, and stack is empty, then accepted).

$L_2 = \{a^n b^n c^{2n} : n \geq 1\}$ is Context-sensitive language and not context-free. (cannot implemented by one stack)

So answer is option B

28 votes

-- Praveen Saini (53k points)

I think only L1 is context free .

Since L1 is explained by others I will take L2

In L2 we can push all a's and all b's onto the stack and on seeing each c we can pop out 1 b and after all b's are popped off we can pop all a's .

that way $a^n b^n$ will be same in number as c^{2n} and ultimately we can attain empty string or bottom of the stack .

But the problem with this approach is that it will accept strings which are not in the language like abbbcccc.

OPTION : B

13 votes

-- ravitheja990 (119 points)

5.2.4 Context Free Language: GATE1987-1-xii [top](#)

<http://gateoverflow.in/8029>

Selected Answer

an ambiguous grammar produces more than one parse tree for any string.. that's B..

3 votes

-- kirti singh (3.3k points)

5.2.5 Context Free Language: GATE1987-2k [top](#)

<http://gateoverflow.in/8059>

Selected Answer

No intersection of two CFLs may or may not be a CFL i.e. CFL is not closed under intersection operation.

Example:

$L_1: \{ a^n b^n c^m \mid n, m \geq 1 \} \cap L_2: \{ a^n b^m c^m \mid n, m \geq 1 \}$

$L_3 = \{ a^m b^m c^m \mid m \geq 1 \}$ Which is CSL

6 votes

-- Prashant Singh (47.6k points)

5.2.6 Context Free Language: GATE1992_02,xix [top](#)

<http://gateoverflow.in/572>

Selected Answer

Answer: A, D

Context Free languages are not closed under intersection and complementation.

6 votes

-- Rajarshi Sarkar (34.4k points)

5.2.7 Context Free Language: GATE1992_02,xviii [top](#)

<http://gateoverflow.in/576>

Selected Answer

Chomsky Normal Form (If all of its production rules are of the form):

$A \rightarrow BC$ or

$A \rightarrow a$ or

$S \rightarrow \epsilon$

where A, B and C are nonterminal symbols, a is a terminal symbol (a symbol that represents a constant value), S is the start symbol, and ϵ is the empty string. Also, neither B nor C may be the start symbol, and the third production rule can only appear if ϵ is in $L(G)$, namely, the language produced by the context-free grammar G.

Applying productions of the first form will increase the number of nonterminals from k to $k+1$, since you replace one nonterminal (-1) with two nonterminals (+2) for a net gain of +1 nonterminal. Since you start with one nonterminal, this means you need to do $l-1$ productions of the first form. You then need l more of the second form to convert the nonterminals to terminals, giving a total of $l+(l-1)=2l-1$ productions.

10 votes

-- Rajarshi Sarkar (34.4k points)

5.2.8 Context Free Language: GATE1995-2.20 [top](#)



Selected Answer

A.

In the other two you can have any number of x and y. There is no such restriction over the number of both being equal.

11 votes

-- Gate Keeda (18.8k points)

5.2.9 Context Free Language: GATE1996_2.8 [top](#)



Selected Answer

B.

CFL's are not closed under intersection.

8 votes

-- Gate Keeda (18.8k points)

5.2.10 Context Free Language: GATE1996_2.9 [top](#)



Selected Answer

(b) is the answer. Because for any binary string of 0's and 1's we can append another string to make it contain equal number of 0's and 1's. i.e., any string over {0,1} is a prefix of a string in L.

Example:

01111 - is prefix of 011110000 which is in L.
1111- is prefix of 11110000 which is in L.
01010- is prefix of 010101 which is in L.

8 votes

-- Arjun Suresh (287k points)

5.2.11 Context Free Language: GATE1999_1.5 [top](#)



Selected Answer

Cfl are not closed under intersection and complement now choose the correct option so b)union and klenne closure

9 votes

-- Bhagirathi Nayak (13.1k points)

5.2.12 Context Free Language: GATE1999_7 [top](#)



Selected Answer

language contains strings where sub string on left of c is same as that on right of c

say 01100c01100

sub string on left of c and right of c cannot be matched with one Stack

while that can be done using two stack

if we push all 0's and 1's on left of c in stack 1 , and all 0's and 1' on right of c in stack 2

then **Top of stack** of both stack will have same symbol .. that can be matched

9 votes

-- Praveen Saini (53k points)

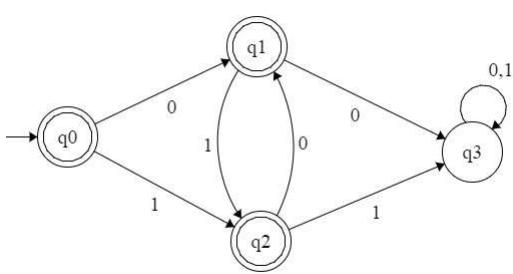
5.2.13 Context Free Language: GATE2000-7 [top](#)

<http://gateoverflow.in/678>



Selected Answer

(a) language $L = (0+1)^* - (0+1)^*(00+11)(0+1)^*$ is it true ?? DFA contains 4 states , 3 are final , 1 is dead state



(b) $i \leq j$

as $S \rightarrow aSAb$

there will be always for one a in left and minimum one b in right and $A \rightarrow bA \mid^*$ can generate any no of b's including null , if A is \mid^* then $i=j$ and if A is generate any b then $j>i$ so condition is $i \leq j$

11 votes

-- Mithlesh Upadhyay (5.3k points)

5.2.14 Context Free Language: GATE2001-1.5 [top](#)

<http://gateoverflow.in/698>



Selected Answer

Answer is (B)

(A) is wrong as a language can be context free even if it is being accepted by non deterministic PDA for ex- { $WW^r : W \in \Sigma^*(a,b)$ and W^r is reverse}

(C) and (D) not always true as Context free languages are not closed under Complement and Intersection.

10 votes

-- Prateeksha Keshari (2.1k points)

5.2.15 Context Free Language: GATE2003-51 [top](#)

<http://gateoverflow.in/940>



Selected Answer

It will be easy to analyze the problem if we replace terminal a and b by (and) respectively.

$S \rightarrow (S) \mid SS \mid \epsilon$

$L(G) =$ balanced parentheses [each left parenthesis has a matching right parenthesis and are well nested]

example () , (()), (()), (() ()).

a) $S \Rightarrow (S) \Rightarrow ()$

$S \Rightarrow SS \Rightarrow S \Rightarrow (S) \Rightarrow ()$

$S \Rightarrow SS \Rightarrow S \Rightarrow (S) \Rightarrow ()$

String () can be derived by above three way each having different derivation tree.

So G is Ambiguous

b) Concatenation of two balance parenthesis will be balanced also . eq. $x = (())$ $y = ()$ $xy = (())()$.

c) We can design Deterministic PDA for L. put left parenthesis (only) in stack and pop with right parenthesis.

d) We cannot design DFA for L because we need a stack to match left parenthesis with right parenthesis.

only option C is true.

13 votes

-- Praveen Saini (53k points)

5.2.16 Context Free Language: GATE2005-57 [top](#)

<http://gateoverflow.in/1380>



Selected Answer

B. http://gatcse.in/wiki/Identify_the_class_of_the_language

7 votes

-- Gate Keeda (18.8k points)

5.2.17 Context Free Language: GATE2006-19 [top](#)

<http://gateoverflow.in/980>



L1 is context-free. We count the number of 0's and check if the remaining number of 1's followed by 0's count to the initial number of 0's.

L2 is not context-free. Here the number of 0's and the following 1's must be same, which can be checked using a PDA. But after that we must also ensure that the following number of 0's must be less than the previous count of 0's and 1's (otherwise $n < 0$, which violates the condition for acceptance) and we cannot do these two checks using a single PDA.

L3 is again not context-free as it is nothing but equal number of 0's followed by equal number of 1's followed by equal number of 0's.

14 votes

-- Arjun Suresh (287k points)

5.2.18 Context Free Language: GATE2006-IT-34 [top](#)

<http://gateoverflow.in/3573>



Selected Answer

$$A \rightarrow bA \mid \epsilon$$

$$\therefore A = b^*$$

$$S \rightarrow aSAb \mid \epsilon$$

$$\equiv S \rightarrow aSb^*b \mid \epsilon$$

$$\equiv S \rightarrow aSb^+ \mid \epsilon$$

$$S = a^n(b^+)^n, \quad n \geq 0$$

$$S = a^n b^n b^*, \quad n \geq 0$$

$$S = a^m b^n, \quad m \leq n$$

Hence, option B is correct.

18 votes

-- Pragy Agarwal (19.3k points)

5.2.19 Context Free Language: GATE2006-IT-4 [top](#)

<http://gateoverflow.in/3543>



Selected Answer

$L(G) = \text{PALINDROME}$

baba does not belong to palindrome , so B is the answer

9 votes

-- Praveen Saini (53k points)

5.2.20 Context Free Language: GATE2007-IT-46 [top](#)



Selected Answer

from Above grammar

Regular expression for G1: $(x+z)^+ + (x+z)^*y(y+z)^+$

Regular expression for G2 : $(y+z+xy)^+$

Option A is correct

4 votes

-- Praveen Saini (53k points)

5.2.21 Context Free Language: GATE2007-IT-48 [top](#)



Selected Answer

ii), iii) and iv)

so option C is correct

Above grammar is for **equal no of x and y**

from Non-terminal S -> xB

=>xy [as B->y one y for one x]

S->xB

=>xxBB [as B ->yBB one B result in one y for one x]

S->xB

=>xyS [as B->yS one y for one x and start again]

Note :Same applies for string start with y i.e . S->yA

9 votes

-- Praveen Saini (53k points)

5.2.22 Context Free Language: GATE2007-IT-49 [top](#)



Selected Answer

Regular grammar is either right linear or left linear. A left linear grammar is one in which there is at most 1 non-terminal on the right side of any production, and it appears at the left most position. Similarly, in right linear grammar non-terminal appears at the right most position.

Here, we can write a right linear grammar for G1 as

S -> w(E
E -> id)S
S -> o

(w - WHILE, o - OTHER)

So, $L(G_1)$ is regular.

Now for G_2 also we can write a right linear grammar:

$S \rightarrow w(E)$
 $E \rightarrow idS$
 $E \rightarrow id+E$
 $E \rightarrow id^*E$
 $S \rightarrow o$

making its language regular.

So, both G_1 and G_2 have an equivalent regular grammar. But as such both these grammars are neither right linear nor left linear and hence not a regular grammar. So, D must be the answer.

<http://www.cs.odu.edu/~toida/nerzic/390teched/regular/grammar/reg-grammar.html>

12 votes

-- Arjun Suresh (287k points)

5.2.23 Context Free Language: GATE2008-IT-34 [top](#)

<http://gateoverflow.in/3344>



Selected Answer

$B \rightarrow 0B00 \mid 1$

generates $\{0^n 1 0^{2n} \mid n \geq 0\}$

$S \rightarrow AA,$
 $A \rightarrow 0A \mid A0 \mid 1$

generates $0A0A \rightarrow 00A0A \rightarrow 00101$, which is there in only B and D choices. D is not the answer as "00" is not generated by the given grammar. So, only option left is B and if we see carefully, non-terminal B is generating the second part of B choice and AA is generating the first part.

$\{0^i 1 0^j 1 0^k \mid i, j, k \geq 0\} \cup \{0^n 1 0^{2n} \mid n \geq 0\}$

10 votes

-- Arjun Suresh (287k points)

5.2.24 Context Free Language: GATE2008-IT-78 [top](#)

<http://gateoverflow.in/3392>



Selected Answer

$S \rightarrow aS$

$S \rightarrow aA$

$S \rightarrow aaAb$

$S \rightarrow aabAab$

$S \rightarrow aabbAaab$

$S \rightarrow aabbaab$

hence d is d answer

22 votes

-- Shreyans Dhankhar (2.5k points)

5.2.25 Context Free Language: GATE2009-12, ISRO2016-37 [top](#)

<http://gateoverflow.in/1304>



Selected Answer

ans is B..string generated by this language is a,b,aba,bab,aabaa,.....all this strings are odd length palindromes

12 votes

-- neha pawar (4.4k points)

5.2.26 Context Free Language: GATE2015-3_32 [top](#)

<http://gateoverflow.in/8489>

Selected Answer

first check for L1. now look $a^m \& b^m$ and $a^n \& b^n$ must be comparable using one stack for CFL.
now take a stack push all a^m in to the stack then push all b^n in to stack now a^n is coming so pop b^n for each a^n by this b^n and a^n will be comparable. now we have left only a^m in stack and b^m is coming so pop a^m for each b^m by which we can compare a^m to b^m ..we conclude that we are comparing this L1 using a single stack so this is CFG.

now for L2.this can not be done in to a single stack because m and n are not comparable we can not find when to push or pop so this is CSL.

now for L3.push all a's into stack and pop 2a's for every b and at last we left with a single a .
bcz here aaaaabb is a valid string where $m=2n+1$ and $n=2$.So realized using single stack hence L3 is CFG.

so the option is B.. L1 and L3 are CFG

22 votes

-- Anoop Sonkar (4.8k points)

5.2.27 Context Free Language: GATE2017-1-10 [top](#)

<http://gateoverflow.in/116290>

Selected Answer

Answer should be B

Consider the 1st production $S \rightarrow abScT$

This production generates equal number of (ab)'s and c's but after each c there is T which goes to $T \rightarrow bT$

so with each c there can be one or more b's (one because of production $T \rightarrow b$ and more because of $T \rightarrow bT$)
and these b's are independent.

For example

ababcbbccbb is the part of the language

and ababcbbbbcbccbb is also the part of the language so we rule out A and C both say equal number of b's after each c

In option D equal number of (ab)'s and c's is not satisfied . The only option that satisfies these 2 conditions is B

uddipto's edit:

people who says this will be the string at any case

$S \rightarrow abcT$

$\rightarrow abcb$

no problem for i B in this case also here $cm^0 = cm^1$ means n value 1 that's it

and remember in B c is not coming eternally.it is coming until m value reaches n of b..right there

.if $m=n=1$ then cb stops

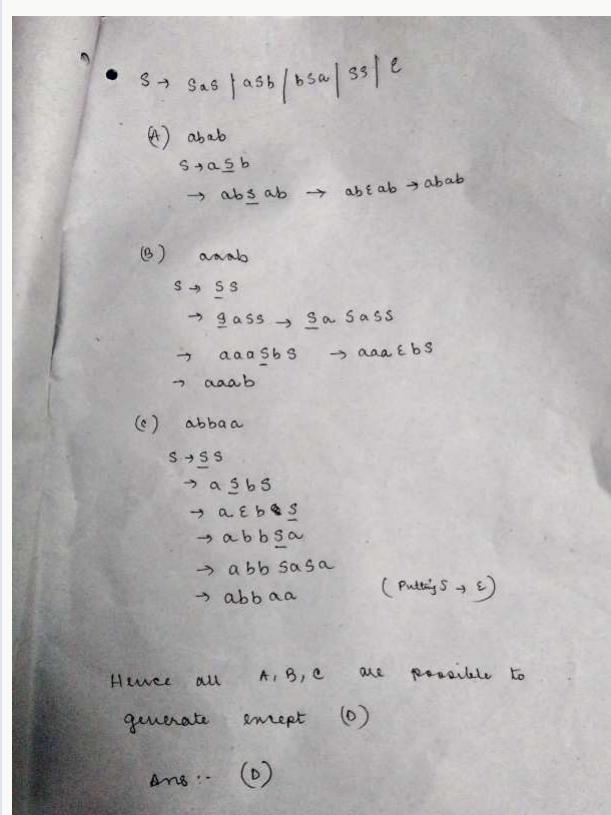
6 votes

-- (points)

5.2.28 Context Free Language: GATE2017-1-34 [top](#)

<http://gateoverflow.in/116316>

Selected Answer



hence ans is d.

6 votes

-- Arnabi (6k points)

5.2.29 Context Free Language: GATE2017-1-38 [top](#)

<http://gateoverflow.in/118321>

Selected Answer

Answer: Option (A)

1). Here let's say $L = L_1 \cup L_2$.

Here as we see , L_1 and L_2 both are DCFLs. And hence CFLs.

And from the Closure Properties of CFLs, we can conclude that CFLs are Closed under Union Operation.

And hence, $L_1 \cup L_2$ is CFL.

2). Now, let's take $L = L_1 \cap L_2$, and that can be expressed as,

$L = \{a^i b^j c^k \mid i=j \text{ AND } j=k, i, j, k \geq 0\}$, in other words

$L = \{a^n b^n c^n \mid n \geq 0\}$

And we know that it comes under CSL(As we will require Two Stacks at a Time). So L is CSL and hence, $L_1 \cap L_2$ is NOT CFL.

3 votes

-- Tilak D. Nanavati (2.6k points)

5.2.30 Context Free Language: GATE2017-2-16 [top](#)

<http://gateoverflow.in/118243>

Selected Answer

$S \rightarrow XY$

$X \rightarrow aX|a$

$Y \rightarrow aYb \mid \epsilon$

X generates atleast one 'a'. While Y generates equal no of a's and b's(including epsilon).

$L = \{ a, ab, aab, aabb, aaabb, \dots \}$

Hence, Ans should be Option C.

3 votes

-- tvkkk (913 points)

5.2.31 Context Free Language: ISI2013-CS-4a [top](#)



Selected Answer

$$L = \{0^i 1^j 0^k \mid j = i + k, i, j, k \geq 0\} = \{\epsilon, 0110, 011100, 001110, \dots\}$$

$$\text{Now } L = \{0^i 1^{i+k} 0^k \mid i, k \geq 0\}$$

Here CFG will be

$$S \rightarrow S_1 S_2$$

$$S_1 \rightarrow 0 S_1 1 \mid \epsilon$$

$$S_2 \rightarrow 1 S_2 0 \mid \epsilon$$

7 votes

-- srestha (55.2k points)

5.2.32 Context Free Language: ISI2015-CS-5b [top](#)



Selected Answer

Given Context free language is :

$$L = \{ a^n b^m c^{n+m} : n, m \text{ are integers, } n \geq 1, m \geq 1 \}.$$

For clarity given language can be interpreted as

$$L = \{ a^n b^m c^m c^n : n, m \text{ are integers, } n \geq 1, m \geq 1 \}.$$

Now the given language is clearly **DCFL**.

And context free grammar (CFG) of the following language:

$$S \rightarrow aSc \mid aAc$$

$$A \rightarrow bAc \mid bc$$

5 votes

-- Manoj Kumar (37k points)

5.3

Countable Set(1) [top](#)

5.3.1 Countable Set: GATE1997-3.4 [top](#)

<http://gateoverflow.in/2235>

Given $\Sigma = \{a, b\}$, which one of the following sets is not countable?

- (a) Set of all strings over Σ

- (b) Set of all languages over Σ
 (c) Set of all regular languages over Σ
 (d) Set of all languages over Σ accepted by Turing machines

gate1997 | theory-of-computation | normal | countable-set

Answer

Answers: Countable Set

5.3.1 Countable Set: GATE1997-3.4 [top](#)



Selected Answer

- (b) Set of all languages over Σ is uncountable.

Ref: http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-045j-automata-computability-and-complexity-spring-2011/lecture-notes/MIT6_045JS11_lec05.pdf

Power set of an infinite set is uncountable. Set of languages over Σ is the power set of set of strings over Σ which is an infinite set. Hence the set of languages becomes an uncountable set.

12 votes

-- Arjun Suresh (287k points)

5.4

Decidability(22) [top](#)

5.4.1 Decidability: GATE 2016-1-17 [top](#)

<http://gateoverflow.in/39651>

Which of the following decision problems are undecidable?

- I. Given NFAs N_1 and N_2 , is $L(N_1) \cap L(N_2) = \Phi$
 - II. Given a CFG $G = (N, \Sigma, P, S)$ and a string $x \in \Sigma^*$, does $x \in L(G)$?
 - III. Given CFGs G_1 and G_2 , is $L(G_1) = L(G_2)$?
 - IV. Given a TM M , is $L(M) = \Phi$?
- A. I and IV only
 B. II and III only
 C. III and IV only
 D. II and IV only

gate2016-1 | theory-of-computation | decidability | easy

Answer

5.4.2 Decidability: GATE1987-2I [top](#)

<http://gateoverflow.in/80603>

State whether the following statement are TRUE or FALSE.

A is recursive if both A and its complement are accepted by Turing machines.

gate1987 | decidability

Answer

5.4.3 Decidability: GATE1989-3-iii [top](#)

<http://gateoverflow.in/87123>

Answer the following questions:

Which of the following problems are undecidable?

- A. Membership problem in context-free languages.
- B. Whether a given context-free language is regular.
- C. Whether a finite state automation halts on all inputs.
- D. Membership problem for type 0 languages.

[gate1989](#) [normal](#) [theory-of-computation](#) [decidability](#)
[Answer](#)

5.4.4 Decidability: GATE1990-3-vii [top](#)

<http://gateoverflow.in/84835>

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

It is undecidable whether:

- A. An arbitrary Turing machine halts after 100 steps.
- B. A Turing machine prints a specific letter.
- C. A Turing machine computes the products of two numbers
- D. None of the above.

[gate1990](#) [normal](#) [theory-of-computation](#) [decidability](#)
[Answer](#)

5.4.5 Decidability: GATE1995_11 [top](#)

<http://gateoverflow.in/2647>

Let L be a language over Σ i.e., $L \subseteq \Sigma^*$. Suppose L satisfies the two conditions given below.

- i. L is in NP and
- ii. For every n , there is exactly one string of length n that belongs to L .

Let L^c be the complement of L over Σ^* . Show that L^c is also in NP.

[gate1995](#) [theory-of-computation](#) [normal](#) [decidability](#)
[Answer](#)

5.4.6 Decidability: GATE1996_1.9 [top](#)

<http://gateoverflow.in/2713>

Which of the following statements is false?

- A. The Halting Problem of Turing machines is undecidable
- B. Determining whether a context-free grammar is ambiguous is undecidable
- C. Given two arbitrary context-free grammars G_1 and G_2 it is undecidable whether $L(G_1) = L(G_2)$
- D. Given two regular grammars G_1 and G_2 it is undecidable whether $L(G_1) = L(G_2)$

[gate1996](#) [theory-of-computation](#) [decidability](#) [easy](#)
[Answer](#)

5.4.7 Decidability: GATE1997_6.5 [top](#)

<http://gateoverflow.in/2261>

Which one of the following is not decidable?

- A. Given a Turing machine M , a string s and an integer k , M accepts s within k steps
- B. Equivalence of two given Turing machines
- C. Language accepted by a given finite state machine is not empty
- D. Language generated by a context free grammar is non-empty

[gate1997](#) [theory-of-computation](#) [decidability](#) [easy](#)
[Answer](#)

5.4.8 Decidability: GATE2000-2.9 [top](#)

<http://gateoverflow.in/655>

Consider the following decision problems:

(P1): Does a given finite state machine accept a given string?

(P2): Does a given context free grammar generate an infinite number of strings?

Which of the following statements is true?

- A. Both (P1) and (P2) are decidable
- B. Neither (P1) nor (P2) is decidable
- C. Only (P1) is decidable
- D. Only (P2) is decidable

gate2000 theory-of-computation decidability normal

Answer

5.4.9 Decidability: GATE2001-2.7 [top](#)

<http://gateoverflow.in/725>

Consider the following problem X.

Given a Turing machine M over the input alphabet Σ , any state q of M and a word $w \in \Sigma^*$, does the computation of M on w visit the state of q?

Which of the following statements about X is correct?

- A. X is decidable
- B. X is undecidable but partially decidable
- C. X is undecidable and not even partially decidable
- D. X is not a decision problem

gate2001 theory-of-computation decidability normal

Answer

5.4.10 Decidability: GATE2003-52 [top](#)

<http://gateoverflow.in/355>

Consider two languages L_1 and L_2 each on the alphabet Σ . Let $f : \Sigma^* \rightarrow \Sigma^*$ be a polynomial time computable bijection such that $(\forall x)[x \in L_1 \text{ iff } f(x) \in L_2]$. Further, let f^{-1} be also polynomial time computable.

Which of the following **CANNOT** be true?

- A. $L_1 \in P$ and L_2 is finite
- B. $L_1 \in NP$ and $L_2 \in P$
- C. L_1 is undecidable and L_2 is decidable
- D. L_1 is recursively enumerable and L_2 is recursive

gate2003 theory-of-computation normal decidability

Answer

5.4.11 Decidability: GATE2005-45 [top](#)

<http://gateoverflow.in/1375>

- Consider three decision problems P_1 , P_2 and P_3 . It is known that P_1 is decidable and P_2 is undecidable. Which one of the following is TRUE?

- A. P_3 is decidable if P_1 is reducible to P_3
- B. P_3 is undecidable if P_3 is reducible to P_2
- C. P_3 is undecidable if P_2 is reducible to P_3
- D. P_3 is decidable if P_3 is reducible to P_2 's complement

gate2005 theory-of-computation decidability normal

Answer

5.4.12 Decidability: GATE2007-6 [top](#)

<http://gateoverflow.in/1204>

Which of the following problems is undecidable?

- A. Membership problem for CFGs
- B. Ambiguity problem for CFGs
- C. Finiteness problem for FSAs
- D. Equivalence problem for FSAs

[gate2007](#) [theory-of-computation](#) [decidability](#) [normal](#)

[Answer](#)

5.4.13 Decidability: GATE2008-10 [top](#)

<http://gateoverflow.in/408>

Which of the following are decidable?

- I. Whether the intersection of two regular languages is infinite
 - II. Whether a given context-free language is regular
 - III. Whether two push-down automata accept the same language
 - IV. Whether a given grammar is context-free
-
- A. I and II
 - B. I and IV
 - C. II and III
 - D. II and IV

[gate2008](#) [theory-of-computation](#) [decidability](#) [easy](#)

[Answer](#)

5.4.14 Decidability: GATE2012_24 [top](#)

<http://gateoverflow.in/1608>

Which of the following problems are decidable?

1. Does a given program ever produce an output?
2. If L is a context-free language, then, is \bar{L} also context-free?
3. If L is a regular language, then, \bar{L} is also regular?
4. If L is a recursive language, then, is \bar{L} also recursive?

- (A) 1, 2, 3, 4
- (B) 1, 2
- (C) 2, 3, 4
- (D) 3, 4

[gate2012](#) [theory-of-computation](#) [decidability](#) [normal](#)

[Answer](#)

5.4.15 Decidability: GATE2013_41 [top](#)

<http://gateoverflow.in/1553>

Which of the following is/are undecidable?

1. G is a CFG. Is $L(G) = \phi$?
 2. G is a CFG. Is $L(G) = \Sigma^*$?
 3. M is a Turing machine. Is $L(M)$ regular?
 4. A is a DFA and N is an NFA. Is $L(A) = L(N)$?
- (A) 3 only (B) 3 and 4 only (C) 1, 2 and 3 only (D) 2 and 3 only

[gate2013](#) [theory-of-computation](#) [decidability](#) [normal](#)

[Answer](#)

5.4.16 Decidability: GATE2014-3-35 [top](#)

<http://gateoverflow.in/2069>

Which one of the following problems is undecidable?

- A. Deciding if a given context-free grammar is ambiguous.
- B. Deciding if a given string is generated by a given context-free grammar.

- C. Deciding if the language generated by a given context-free grammar is empty.
 D. Deciding if the language generated by a given context-free grammar is finite.

gate2014-3 | theory-of-computation | context-free-language | decidability | normal

[Answer](#)

5.4.17 Decidability: GATE2015-2_21 [top](#)

<http://gateoverflow.in/8111>

Consider the following statements.

- I. The complement of every Turing decidable language is Turing decidable
 II. There exists some language which is in NP but is not Turing decidable
 III. If L is a language in NP, L is Turing decidable

Which of the above statements is/are true?

- A. Only II
 B. Only III
 C. Only I and II
 D. Only I and III

gate2015-2 | theory-of-computation | decidability | easy

[Answer](#)

5.4.18 Decidability: GATE2015-3_53 [top](#)

<http://gateoverflow.in/8562>

Language L_1 is polynomial time reducible to language L_2 . Language L_3 is polynomial time reducible to language L_2 , which in turn polynomial time reducible to language L_4 . Which of the following is/are true?

- I. if $L_4 \in P$, then $L_2 \in P$
 II. if $L_1 \in P$ or $L_3 \in P$, then $L_2 \in P$
 III. $L_1 \in P$, if and only if $L_3 \in P$
 IV. if $L_4 \in P$, then $L_3 \in P$

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

gate2015-3 | theory-of-computation | decidability | normal

[Answer](#)

5.4.19 Decidability: GATE2017-1-39 [top](#)

<http://gateoverflow.in/118322>

Let A and B be finite alphabets and let $\#$ be a symbol outside both A and B . Let f be a total function from A^* to B^* . We say f is *computable* if there exists a Turing machine M which given an input $x \in A^*$, always halts with $f(x)$ on its tape.

Let L_f denote the language $\{x\#f(x) \mid x \in A^*\}$. Which of the following statements is true:

- (A) f is computable if and only if L_f is recursive.
 (B) f is computable if and only if L_f is recursively enumerable.
 (C) If f is computable then L_f is recursive, but not conversely.
 (D) If f is computable then L_f is recursively enumerable, but not conversely.

gate2017-1 | theory-of-computation | decidability | difficult

[Answer](#)

5.4.20 Decidability: GATE2017-2-41 [top](#)

<http://gateoverflow.in/118605>

Let $L(R)$ be the language represented by regular expression R . Let $L(G)$ be the language generated by a context free grammar G . Let $L(M)$ be the language accepted by a Turing machine M . Which of the following decision problems are undecidable?

- I. Given a regular expression R and a string w , is $w \in L(R)$?
 - II. Given a context-free grammar G , is $L(G) = \emptyset$?
 - III. Given a context-free grammar G , is $L(G) = \Sigma^*$ for some alphabet Σ ?
 - IV. Given a Turing machine M and a string w , is $w \in L(M)$?
- A. I and IV only
 - B. II and III only
 - C. II, III and IV only
 - D. III and IV only

[gate2017-2](#) | [theory-of-computation](#) | [decidability](#)

[Answer](#)

5.4.21 Decidability: TIFR2010-B-25 [top](#)

<http://gateoverflow.in/18745>

Which of the following problems is decidable? (Here, CFG means context free grammar and CFL means context free language.)

- A. Give a CFG G , find whether $L(G) = R$, where R is regular set.
- B. Given a CFG G , find whether $L(G) = \{\}$.
- C. Find whether the intersection of two CFLs is empty.
- D. Find whether the complement of CFL is a CFL.
- E. Find whether CFG G_1 and CFG G_2 generate the same language, i.e., $L(G_1) = L(G_2)$.

[tifr2010](#) | [theory-of-computation](#) | [context-free-language](#) | [decidability](#)

[Answer](#)

5.4.22 Decidability: TIFR2011-B-25 [top](#)

<http://gateoverflow.in/20404>

Let A_{TM} be defined as follows:

$$A_{TM} = \{\langle M, w \rangle \mid \text{The Turning machine } M \text{ accepts the word } w\}$$

And let L be some NP-complete language. Which of the following statements is FALSE?

- a. $L \in \text{NP}$
- b. Every problem in NP is polynomial time reducible to L .
- c. Every problem in NP is polynomial time reducible to A_{TM} .
- d. Since L is NP-complete, A_{TM} is polynomial time reducible to L .
- e. $A_{TM} \notin \text{NP}$.

[tifr2011](#) | [theory-of-computation](#) | [decidability](#)

[Answer](#)

Answers: Decidability

5.4.1 Decidability: GATE 2016-1-17 [top](#)

<http://gateoverflow.in/39651>



Selected Answer

- I. is Decidable, we may use cross product of NFA (or by converting them into DFA), if We didn't get final states of both together at any state in it. then $L(N_1) \cap L(N_2) = \emptyset$, Disjoint languages.
- II. Membership in CFG is Decidable (CYK algorithm)
- III. Equivalence of Two context free grammars is Undecidable.
- IV. For TM M , $L(M) = \emptyset$ is Undecidable.

29 votes

-- Praveen Saini (53k points)

5.4.2 Decidability: GATE1987-2I [top](#)

<http://gateoverflow.in/80603>

Yes if A and its complement are accepted by Turing machines.then A is recursive.

Suppose a language A is recursively enumerable. That means there exists a Turing machine T1 that, given any string of the language, halts and accepts that string.

Now let's also suppose that the complement of A, $A' = \{w : w \mid A\}$, is recursively enumerable. That means there is some other Turing machine T2 that, given any string of A' halts and accepts that string. So any string belongs to either A or A' . Hence, any string will cause either T1 or T2 (or both) to halt. We construct a new Turing machine that emulates both T1 and T2, alternating moves between them. When either one stops, we can tell (by whether it accepted or rejected the string) to which language the string belongs.

Thus, we have constructed a Turing machine that, for each input, halts with an answer whether or not the string belongs to A' . Therefore A and A' are recursive languages.

3 votes

-- Prashant Singh (47.6k points)

5.4.3 Decidability: GATE1989-3-iii [top](#)

<http://gateoverflow.in/87123>

1. Membership problem in context-free languages. is **Decidable**.
2. Whether a given context-free language is regular. **Undecidable** [Regularity is decidable till DCFL]
3. Whether a finite state automation halts on all inputs. **Decidable**
4. Membership problem for type 0 languages. **Undecidable**[undecidable for RE or semidecidable]

Reff : <http://gatecse.in/grammar-decidable-and-undecidable-problems/>

3 votes

-- Prashant Singh (47.6k points)

5.4.4 Decidability: GATE1990-3-vii [top](#)

<http://gateoverflow.in/84835>

A.An arbitrary Turing machine halts after 100 steps. **DECIDABLE**,we can run TM for 100 steps and conclude that

B.A Turing machine prints a specific letter. **UNDECIDABLE**,

C.A Turing machine computes the products of two numbers,**UNDECIDABLE**, Even though we can design a TM for calculation product of 2 numbers but here it is asking whether given TM computes product of 2 numbers,so the behaviour of TM unknown hence,Undecidable

4 votes

-- Prajwal Bhat (11.6k points)

5.4.5 Decidability: GATE1995_11 [top](#)

<http://gateoverflow.in/2647>

Selected Answer

Since L is in NP it is decidable (recursive) and so is its complement L^c . Now, L^c may or may not be in NP. But we are given that for any string length n , exactly one string belongs to L , which means for any string length all but one string belongs to L^c .

Now, definition of NP says that all "yes" instances of the problem can be solved in polynomial time using a nondeterministic TM. So, given an instance of $\langle L^c, x \rangle$, we non-deterministically take all words of length n , where n is the length of w , and see if it is in L . As soon as we get the word (such a word is sure to exist as exactly one word of length n belongs to L), we see if this word is same as x . If it is not same (and only if it is not same), $x \in L^c$ and we get this answer in polynomial time making L^c an NP problem.

5 votes

-- Arjun Suresh (287k points)

5.4.6 Decidability: GATE1996_1.9 [top](#)

<http://gateoverflow.in/2713>



Selected Answer

D..

equivalence of Regular languages is decidable.

1. Membership,
2. Emptiness,
3. Finiteness,
4. Equivalence,
5. Ambiguity,
6. Regularity,
7. Everything,
8. Disjointedness...

All are decidable for Regular languages.

First 3 for CFL.

Only 1st for CSL and REC.

None for RE.

8 votes

-- Gate Keeda (18.8k points)

5.4.7 Decidability: GATE1997_6.5 [top](#)

<http://gateoverflow.in/2261>



Selected Answer

Equivalence of two TMs is undecidable

12 votes

-- Bhagirathi Nayak (13.1k points)

5.4.8 Decidability: GATE2000-2.9 [top](#)

<http://gateoverflow.in/656>



Selected Answer

For P1, we just need to give a run on the machine. Finite state machines always halts unlike TM.

For P2, check if the CFG generates any string of length between n and $2n - 1$, where n is the pumping lemma constant. If so, $L(CFG)$ is infinite, else finite. But finding the pumping lemma constant is not trivial - but there are other procedures which can do this - <http://cs.stackexchange.com/questions/52507/is-it-decidable-whether-a-given-context-free-grammar-generates-an-infinite-number/52520>

Hence, both P1 and P2 are decidable - (A).

http://gatcse.in/wiki/Grammar:_Decidable_and_Undecidable_Problems

10 votes

-- Arjun Suresh (287k points)

5.4.9 Decidability: GATE2001-2.7 [top](#)

<http://gateoverflow.in/725>



Selected Answer

X is undecidable but partially decidable.

We have the TM M. Just make the state q the final state and make all other final states non-final and get a new TM M'. Give

input w to M' . If w would have taken M to state q (yes case of the problem), our new TM M' would accept it. So, the given problem is partially decidable.

If M goes for an infinite loop and never reaches state q (no case for the problem), M' cannot output anything. This problem is the state entry problem, which like word accepting problem and halting problem is undecidable.

1 21 votes

-- Arjun Suresh (287k points)

5.4.10 Decidability: GATE2003-52 [top](#)

<http://gateoverflow.in/355>



Selected Answer

Since, f is a polynomial time computable bijection and f^{-1} is also polynomial time computable, L_1 and L_2 should have the same complexity (isomorphic). This is because, given a problem for L_1 , we can always do a polynomial time reduction to L_2 and vice versa. Hence, the answer is 'C', as in 'A', L_1 and L_2 can be finite, in 'B', L_1 and L_2 can be in P and in 'D', L_1 and L_2 can be recursive. Only, in 'C' there is no intersection for L_1 and L_2 , and hence it can't be true.

Alternatively, we can prove 'C' to be false as follows:

Given L_2 is decidable. Now, for a problem in L_1 , we can have a TM , which takes an input x , calculates $f(x)$ in polynomial time, check $f(x)$ is in L_2 (this is decidable as L_2 is decidable), and if it is, then output yes and otherwise no. Thus L_1 must also be decidable.

1 8 votes

-- gatecse (13.1k points)

5.4.11 Decidability: GATE2005-45 [top](#)

<http://gateoverflow.in/1375>



Selected Answer

- (A) If P_1 is reducible to P_3 , then P_3 is at least as hard as P_1 . So, no guarantee if P_3 is decidable.
- (B) If P_3 is reducible to P_2 , then P_3 cannot be harder than P_2 . But P_2 being undecidable, this can't say P_3 is undecidable.
- (C) If P_2 is reducible to P_3 , then P_3 is at least as hard as P_2 . Since, P_2 is undecidable, this means P_3 is also undecidable - hence the answer.
- (D) Complement of an undecidable problem is undecidable. Now, reducing to an undecidable problem can't prove P_3 is decidable.

http://gatecse.in/wiki/Some_Reduction_Inferences

1 15 votes

-- Arjun Suresh (287k points)

5.4.12 Decidability: GATE2007-6 [top](#)

<http://gateoverflow.in/1204>



Selected Answer

Membership problem is decidable as it can be solved by parsers.

Finiteness problem is decidable for FSAs (also for CFGs), as we just need to check for a loop in the DFA.

Equivalence problem for FSAs is decidable as we can take the complement of one FSA (complement of FSA is another FSA), and do an intersection with the other (FSAs are closed under intersection also), giving a new FSA. If this new FSA accept no string, then the given FSAs are equivalent, else not equivalent.

Only ambiguity problem for CFGs are undecidable.

http://gatecse.in/wiki/Grammar:_Decidable_and_Undecidable_Problems

1 12 votes

-- Arjun Suresh (287k points)

5.4.13 Decidability: GATE2008-10 [top](#)

<http://gateoverflow.in/408>



Selected Answer

- (1) Intersection of two regular languages is regular. And checking if a regular language is infinite is decidable.
- (2) Undecidable
- (3) Undecidable
- (4) Decidable as we just have to check if the grammar obeys the rules of CFG. (Obviously undecidable had it been language instead of grammar)

Reference: http://gatecse.in/wiki/Grammar:_Decidable_and_Undecidable_Problems

10 votes

-- gatecse (13.1k points)

5.4.14 Decidability: GATE2012_24 [top](#)



Selected Answer

CFL's are not closed under complementation and a program can loop forever. So, it may not produce any output.

Regular and recursive languages are closed under complementation.

Hence, only 3,4 are decidable.

9 votes

-- Bhagirathi Nayak (13.1k points)

5.4.15 Decidability: GATE2013_41 [top](#)



Selected Answer

It will be D.

First is Emptiness for CFG.

Second is everything for CFG.

Third is Regularity for REC

Fourth is equivalence for regular.

13 votes

-- Gate Keeda (18.8k points)

5.4.16 Decidability: GATE2014-3-35 [top](#)



Selected Answer

(A) is the answer. Proving (A) is undecidable is not so easy. But we can easily prove the other three options given here are decidable.

http://gatecse.in/wiki/Grammar:_Decidable_and_Undecidable_Problems

11 votes

-- Arjun Suresh (287k points)

5.4.17 Decidability: GATE2015-2_21 [top](#)



Selected Answer

1 is true. The solution to a decision problem is either "yes" or "no", and hence if we can decide a problem, we have also decided its complement- just reverse "yes" and "no". (This is applicable for decidability and not for acceptance)

2 is false. Because NP class is defined as the class of languages that can be solved in polynomial time by a non-deterministic Turing machine. So, none of the NP class problems is undecidable.

3 is true for same reason as 2.

So, answer is D.

20 votes

-- Arjun Suresh (287k points)

5.4.18 Decidability: GATE2015-3_53 [top](#)



Selected Answer

1. L1 is polynomial time reducible to L2. So, L2 is at least as hard as L1.
2. L3 is polynomial time reducible to L2. So, L2 is at least as hard as L3.
3. L2 is polynomial time reducible to L4. So, L4 is at least as hard as L2.

If L4 is in P, L3, L2 and L1 must also be in P. So, I and IV are true.

We can have L1 in P and L2 not in P, and none of the given conditions are violated. So, II is false.

Assume L3 not in P. Now, Since L2 must be at least as hard as L3, it must also be not in P. But L1 is less harder than L1 as per condition 1, and it can be in P without violating any given conditions. So, III is false.

Hence C choice.

More Info: http://gatecse.in/wiki/Some_Reduction_Inferences

12 votes

-- Arjun Suresh (287k points)

5.4.19 Decidability: GATE2017-1-39 [top](#)

<http://gateoverflow.in/11832>

f being computable, given x , we get $f(x)$.

L_f is recursive - given any string we get to know if the string is in L or not.

Now lets see the two cases:

- Does f being computable implies L_f is recursive?

We are given x and we need to know if $x \# f(x)$ belongs to L_f . Since f being computable we can calculate $f(x)$ and our problem reduces to 3 string comparisons - first x , followed by $\#$ and then $f(x)$ which can be done by a TM. So, L_f must be recursive.

- Does L_f being recursive implies f be computable?

If L_f is recursive, given any string we can say whether it belongs to L_f or not. Now, to compute $f(x)$, we can do a dovetailing approach starting with strings in lexicographic (or any other order) as follows s_1, s_2, s_3, \dots and forming inputs to the TM as

$x \# s_1, x \# s_2, x \# s_3, \dots$

By dovetailing it means to simulation TM for one step in first iteration, 2 steps in second iteration, 3 steps in third iteration etc., and in each iteration adding a new string to the simulation (TM is parallelly simulating all strings).

https://courses.engr.illinois.edu/cs373/sp2009/lectures/lect_24.pdf

Now, for any x , we should have an $f(x)$ since f is a total function. So, in our simulation we are sure that sometime we will add the string $x \# f(x)$ to the simulation and then the TM will halt for it. And whenever it does halt, we can just take the part of the input string after $\#$, and that is $f(x)$. We have a way to compute $f(x)$ for any x . So, f is computable. The same method would work even if L_f is recursively enumerable and not necessarily recursive.

So, the correct statements are

1. If L_f is recursively enumerable f is computable
2. If f is computable L_f is recursive

So, options A and B are TRUE but official key was only A.

2 votes

-- Arjun Suresh (287k points)

5.4.20 Decidability: GATE2017-2-41 [top](#)

<http://gateoverflow.in/118605>



Selected Answer

1st statement is Membership problem of regular language = **decidable**

2nd statement is Emptyness problem of CFL = **decidable**

3rd statement is accept everthing problem of CFL = **undecidable**

4th statement is Membership problem of RE language = **undecidable**

D is answer

6 votes

-- Prashant Singh (47.6k points)

5.4.21 Decidability: TIFR2010-B-25 [top](#)

<http://gateoverflow.in/16745>



Selected Answer

A) we dont have any standard algo to change CFG into CFL.

from a given CFG deciding a launguage is finite is decidable but regular its undecidable (check out the link provided by Arjun sir in the comment for better clarification.)

B)From a given given CFG we can determine the CFL and CFL emptiness is Decidable.

C) Intersection of two CFL is undecidable coz it is not closed under intersection.

D) CFL is not closed under Complement so its undecidable.

E) CFL is not closed under equivalence so it is undecidale to compare 2 language.

therefore according to me B is decidable and A,C,D and E are undecidable.

6 votes

-- Umang Raman (14.5k points)

5.4.22 Decidability: TIFR2011-B-25 [top](#)

<http://gateoverflow.in/20404>



Selected Answer

A_{TM} is the language of the Halting Problem. It is undecidable, but Recursively Enumerable.

L is NPC

- True. Any language in **NPC** is also in **NP** by definition.
- True. By definition, any problem in **NP** is polynomial time reducible to any **NPC** problem.
- True. A_{TM} is undecidable. Any language that is decidable is polynomial time reducible to A_{TM} !
- False.**
 A_{TM} is undecidable. No Turing Machine can guarantee an answer in a finite time, let alone a polynomial time.
- True. A_{TM} is undecidable. It is certainly not in **NP**.

Hence, the correct answer is option d.

6 votes

-- Pragy Agarwal (19.3k points)

5.5

Finite Automata(63) top

5.5.1 Finite Automata: CMI2010-B-04b top

<http://gateoverflow.in/47073>

Indicate whether the following statement is true or false, providing a short explanation to substantiate your answers.

A DFA that has n states and accepts an infinite language must accept at least one string x such that $2n < |x| < 3n$, where $|x|$ denotes the length of x .

descriptive cmi2010 finite-automata

Answer

5.5.2 Finite Automata: CMI2010-B-04c top

<http://gateoverflow.in/47074>

Indicate whether the following statement is true or false, providing a short explanation to substantiate your answers.

If a language L is accepted by an NFA with n states then there is a DFA with no more than 2^n states accepting L .

descriptive cmi2010 finite-automata

Answer

5.5.3 Finite Automata: GATE 2016-2-42 top

<http://gateoverflow.in/39591>

Consider the following two statements:

I. If all states of an **NFA** are accepting states then the language accepted by the **NFA** is \sum^* .

II. There exists a regular language A such that for all languages B , $A \cap B$ is regular.

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

- A. Only I is true
- B. Only II is true
- C. Both I and II are true
- D. Both I and II are false

gate2016-2 theory-of-computation finite-automata normal

Answer

5.5.4 Finite Automata: GATE 2016-2_16 top

<http://gateoverflow.in/39562>

The number of states in the minimum sized DFA that accepts the language defined by the regular expression.

$(0 + 1)^*(0 + 1)(0 + 1)^*$

is _____.

gate2016-2 theory-of-computation finite-automata normal numerical-answers

Answer

5.5.5 Finite Automata: GATE1987-2j top

<http://gateoverflow.in/80594>

State whether the following statements are TRUE or FALSE:

A minimal DFA that is equivalent to an NDFA with n nodes has always 2^n states.

gate1987 theory-of-computation finite-automata

Answer

5.5.6 Finite Automata: GATE1991_17,b top

<http://gateoverflow.in/544>

Let L be the language of all binary strings in which the third symbol from the right is a 1. Give a non-deterministic finite automaton that recognizes L . How many states does the minimized equivalent deterministic finite automaton have? Justify your answer briefly?

gate1991 theory-of-computation finite-automata normal

Answer

5.5.7 Finite Automata: GATE1993_27 [top](#)

<http://gateoverflow.in/2323>

Draw the state transition of a deterministic finite state automaton which accepts all strings from the alphabet $\{a, b\}$, such that no string has 3 consecutive occurrences of the letter b.

gate1993 theory-of-computation finite-automata easy

Answer

5.5.8 Finite Automata: GATE1994_3.3 [top](#)

<http://gateoverflow.in/2480>

State True or False with one line explanation

A FSM (Finite State Machine) can be designed to add two integers of any arbitrary length (arbitrary number of digits).

gate1994 theory-of-computation finite-automata normal

Answer

5.5.9 Finite Automata: GATE1995_2.23 [top](#)

<http://gateoverflow.in/2636>

A finite state machine with the following state table has a single input x and a single output z .

present state	next state, z	
	$x = 1$	$x = 0$
A	D, 0	B, 0
B	B, 1	C, 1
C	B, 0	D, 1
D	B, 1	C, 0

If the initial state is unknown, then the shortest input sequence to reach the final state C is:

- A. 01
- B. 10
- C. 101
- D. 110

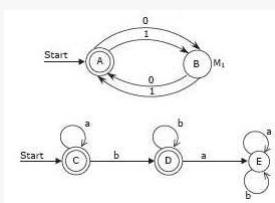
gate1995 theory-of-computation finite-automata normal

Answer

5.5.10 Finite Automata: GATE1996_12 [top](#)

<http://gateoverflow.in/2764>

Given below are the transition diagrams for two finite state machines M_1 and M_2 recognizing languages L_1 and L_2 respectively.



- a. Display the transition diagram for a machine that recognizes $L_1 \cdot L_2$, obtained from transition diagrams for M_1 and M_2 by adding only ϵ transitions and no new states.
- b. Modify the transition diagram obtained in part (a) obtain a transition diagram for a machine that recognizes $(L_1 \cdot L_2)^*$ by adding only ϵ transitions and no new states.

(Final states are enclosed in double circles).

gate1996 theory-of-computation finite-automata normal

Answer

5.5.11 Finite Automata: GATE1996_2.23 [top](#)

<http://gateoverflow.in/2752>

Consider the following state table for a sequential machine. The number of states in the minimized machine will be

		Input	
		0	1
Present State	A	D, 0	B, 1
	B	A, 0	C, 1
	C	A, 0	B, 1
	D	A, 1	C, 1
		Next state, Output	

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

gate1996 theory-of-computation normal finite-automata

Answer

5.5.12 Finite Automata: GATE1997_20 [top](#)

<http://gateoverflow.in/2280>

Construct a finite state machine with minimum number of states, accepting all strings over (a,b) such that the number of a's is divisible by two and the number of b's is divisible by three.

gate1997 theory-of-computation finite-automata normal

Answer

5.5.13 Finite Automata: GATE1997_21 [top](#)

<http://gateoverflow.in/2281>

Given that L is a language accepted by a finite state machine, show that L^P and L^R are also accepted by some finite state machines, where

$$L^P = \{s \mid ss' \in L \text{ some string } s'\}$$

$$L^R = \{s \mid s \text{ obtained by reversing some string in } L\}$$

gate1997 theory-of-computation finite-automata proof

Answer

5.5.14 Finite Automata: GATE1998_1.10 [top](#)

<http://gateoverflow.in/1641>

Which of the following set can be recognized by a Deterministic Finite state Automaton?

- A. The numbers $1, 2, 4, 8, \dots, 2^n, \dots$ written in binary
- B. The numbers $1, 2, 4, 8, \dots, 2^n, \dots$ written in unary
- C. The set of binary string in which the number of zeros is the same as the number of ones.
- D. The set $\{1, 101, 11011, 1110111, \dots\}$

gate1998 theory-of-computation finite-automata normal

Answer**5.5.15 Finite Automata: GATE1998_2.5** [top](#)<http://gateoverflow.in/1677>

Let L be the set of all binary strings whose last two symbols are the same. The number of states in the minimal state deterministic finite state automaton accepting L is

- A. 2
- B. 5
- C. 8
- D. 3

[gate1998](#) [theory-of-computation](#) [finite-automata](#) [normal](#)
Answer**5.5.16 Finite Automata: GATE1998_4** [top](#)<http://gateoverflow.in/1695>

Design a deterministic finite state automaton (using minimum number of states) that recognizes the following language:

$$L = \{w \in \{0,1\}^* \mid w \text{ interpreted as binary number (ignoring the leading zeros) is divisible by five}\}.$$

[gate1998](#) [theory-of-computation](#) [finite-automata](#) [normal](#)
Answer**5.5.17 Finite Automata: GATE1999_1.4** [top](#)<http://gateoverflow.in/1458>

Consider the regular expression $(0+1)(0+1)\dots N$ times. The minimum state finite automaton that recognizes the language represented by this regular expression contains

- A. n states
- B. $n+1$ states
- C. $n+2$ states
- D. None of the above

[gate1999](#) [theory-of-computation](#) [finite-automata](#) [easy](#)
Answer**5.5.18 Finite Automata: GATE2001-1.6** [top](#)<http://gateoverflow.in/699>

Given an arbitrary non-deterministic finite automaton (NFA) with N states, the maximum number of states in an equivalent minimized DFA at least

- A. N^2
- B. 2^N
- C. $2N$
- D. $N!$

[gate2001](#) [finite-automata](#) [theory-of-computation](#) [easy](#)
Answer**5.5.19 Finite Automata: GATE2001-2.5** [top](#)<http://gateoverflow.in/723>

Consider a DFA over $\Sigma = \{a,b\}$ accepting all strings which have number of a's divisible by 6 and number of b's divisible by 8. What is the minimum number of states that the DFA will have?

- A. 8
- B. 14
- C. 15
- D. 48

[gate2001](#) [theory-of-computation](#) [finite-automata](#)

Answer**5.5.20 Finite Automata: GATE2001-5** [top](#)<http://gateoverflow.in/746>

Construct DFA's for the following languages:

- $L = \{w \mid w \in \{a,b\}^*, w \text{ has baab as a substring}\}$
- $L = \{w \mid w \in \{a,b\}^*, w \text{ has an odd number of a's and an odd number of b's}\}$

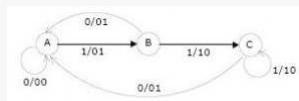
[gate2001](#) [theory-of-computation](#) [easy](#) [descriptive](#) [finite-automata](#)
Answer**5.5.21 Finite Automata: GATE2002-2.13** [top](#)<http://gateoverflow.in/843>

The smallest finite automaton which accepts the language $\{x \mid \text{length of } x \text{ is divisible by 3}\}$ has

- A. 2 states
- B. 3 states
- C. 4 states
- D. 5 states

[gate2002](#) [theory-of-computation](#) [normal](#) [finite-automata](#)
Answer**5.5.22 Finite Automata: GATE2002-2.5** [top](#)<http://gateoverflow.in/835>

The finite state machine described by the following state diagram with A as starting state, where an arc label is and x stands for 1-bit input and y stands for 2-bit output



- A. outputs the sum of the present and the previous bits of the input
- B. outputs 01 whenever the input sequence contains 11
- C. outputs 00 whenever the input sequence contains 10
- D. none of the above

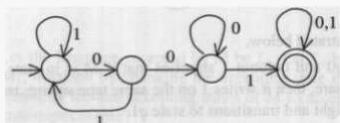
[gate2002](#) [theory-of-computation](#) [normal](#) [finite-automata](#)
Answer**5.5.23 Finite Automata: GATE2002-21** [top](#)<http://gateoverflow.in/874>

We require a four state automaton to recognize the regular expression $(a/b)^*abb$

- Give an NFA for this purpose
- Give a DFA for this purpose

[gate2002](#) [theory-of-computation](#) [finite-automata](#) [normal](#) [descriptive](#)
Answer**5.5.24 Finite Automata: GATE2003-50** [top](#)<http://gateoverflow.in/839>

Consider the following deterministic finite state automaton M.



Let S denote the set of seven bit binary strings in which the first, the fourth, and the last bits are 1. The number of strings in S that are accepted by M is

- A. 1
- B. 5
- C. 7
- D. 8

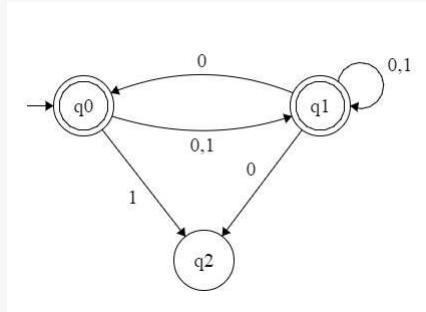
gate2003 theory-of-computation finite-automata normal

[Answer](#)

5.5.25 Finite Automata: GATE2003-55 [top](#)

<http://gateoverflow.in/943>

Consider the NFA M shown below.



Let the language accepted by M be L . Let L_1 be the language accepted by the NFA M_1 obtained by changing the accepting state of M to a non-accepting state and by changing the non-accepting states of M to accepting states. Which of the following statements is true?

- A. $L_1 = \{0,1\}^* - L$
- B. $L_1 = \{0,1\}^*$
- C. $L_1 \subseteq L$
- D. $L_1 = L$

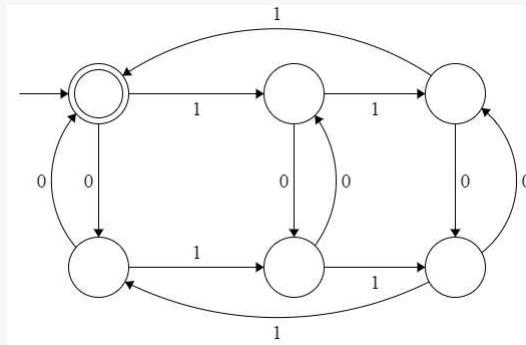
gate2003 theory-of-computation finite-automata normal

[Answer](#)

5.5.26 Finite Automata: GATE2004-86 [top](#)

<http://gateoverflow.in/1089>

The following finite state machine accepts all those binary strings in which the number of 1's and 0's are respectively



- A. divisible by 3 and 2
- B. odd and even
- C. even and odd
- D. divisible by 2 and 3

gate2004 theory-of-computation finite-automata easy

Answer**5.5.27 Finite Automata: GATE2004-IT-41** [top](#)<http://gateoverflow.in/3684>

Let $M = (K, \Sigma, \delta, s, F)$ be a finite state automaton, where

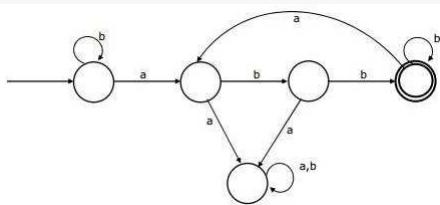
$K = \{A, B\}$, $\Sigma = \{a, b\}$, $s = A$, $F = \{B\}$,
 $\delta(A, a) = A$, $\delta(A, b) = B$, $\delta(B, a) = B$ and $\delta(B, b) = A$

A grammar to generate the language accepted by M can be specified as $G = (V, \Sigma, R, S)$, where $V = K \cup \Sigma$, and $S = A$. Which one of the following set of rules will make $L(G) = L(M)$?

- A. $\{A \rightarrow aB, A \rightarrow bA, B \rightarrow bA, B \rightarrow aA, B \rightarrow \epsilon\}$
- B. $\{A \rightarrow aA, A \rightarrow bB, B \rightarrow aB, B \rightarrow bA, B \rightarrow \epsilon\}$
- C. $\{A \rightarrow bB, A \rightarrow aB, B \rightarrow aA, B \rightarrow bA, B \rightarrow \epsilon\}$
- D. $\{A \rightarrow aA, A \rightarrow bA, B \rightarrow aB, B \rightarrow bA, A \rightarrow \epsilon\}$

[gate2004-it](#) [theory-of-computation](#) [finite-automata](#) [normal](#)**Answer****5.5.28 Finite Automata: GATE2005-53** [top](#)<http://gateoverflow.in/1376>

Consider the machine M :

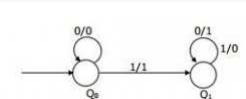


The language recognized by M is:

- A. $\{w \in \{a,b\}^* \mid \text{every } a \text{ in } w \text{ is followed by exactly two } b's\}$
- B. $\{w \in \{a,b\}^* \mid \text{every } a \text{ in } w \text{ is followed by at least two } b's\}$
- C. $\{w \in \{a,b\}^* \mid w \text{ contains the substring 'abb'}\}$
- D. $\{w \in \{a,b\}^* \mid w \text{ does not contain 'aa' as a substring}\}$

[gate2005](#) [theory-of-computation](#) [finite-automata](#) [normal](#)**Answer****5.5.29 Finite Automata: GATE2005-63** [top](#)<http://gateoverflow.in/1386>

The following diagram represents a finite state machine which takes as input a binary number from the least significant bit.



Which of the following is TRUE?

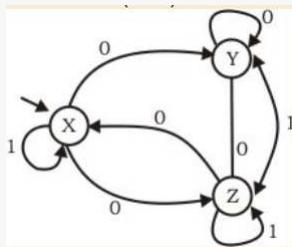
- A. It computes 1's complement of the input number
- B. It computes 2's complement of the input number
- C. It increments the input number
- D. it decrements the input number

[gate2005](#) [theory-of-computation](#) [finite-automata](#) [easy](#)**Answer**

5.5.30 Finite Automata: GATE2005-IT-37 [top](#)

<http://gateoverflow.in/3784>

Consider the non-deterministic finite automaton (NFA) shown in the figure.



State X is the starting state of the automaton. Let the language accepted by the NFA with Y as the only accepting state be L1. Similarly, let the language accepted by the NFA with Z as the only accepting state be L2. Which of the following statements about L1 and L2 is TRUE?

- A. L1 = L2
- B. L1 ⊂ L2
- C. L2 ⊂ L1
- D. None of the above

[gate2005-it](#) [theory-of-computation](#) [finite-automata](#) [normal](#)

[Answer](#)

5.5.31 Finite Automata: GATE2005-IT-39 [top](#)

<http://gateoverflow.in/3785>

Consider the regular grammar:

$$\begin{aligned} S &\rightarrow Xa \mid Ya \\ X &\rightarrow Za \\ Z &\rightarrow Sa \mid \epsilon \\ Y &\rightarrow Wa \\ W &\rightarrow Sa \end{aligned}$$

where S is the starting symbol, the set of terminals is {a} and the set of non-terminals is {S, W, X, Y, Z}. We wish to construct a deterministic finite automaton (DFA) to recognize the same language. What is the minimum number of states required for the DFA?

- A. 2
- B. 3
- C. 4
- D. 5

[gate2005-it](#) [theory-of-computation](#) [finite-automata](#) [normal](#)

[Answer](#)

5.5.32 Finite Automata: GATE2006-34 [top](#)

<http://gateoverflow.in/1291>

Consider the regular language $L = (111 + 11111)^*$. The minimum number of states in any DFA accepting this language is:

- A. 3
- B. 5
- C. 8
- D. 9

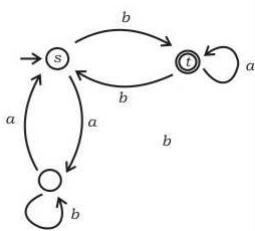
[gate2006](#) [theory-of-computation](#) [finite-automata](#) [normal](#)

[Answer](#)

5.5.33 Finite Automata: GATE2006-IT-3 [top](#)

<http://gateoverflow.in/3542>

In the automaton below, s is the start state and t is the only final state.



Consider the strings $u = abbaba$, $v = bab$, and $w = aabb$. Which of the following statements is true?

- A. The automaton accepts u and v but not w
- B. The automaton accepts each of u , v , and w
- C. The automaton rejects each of u , v , and w
- D. The automaton accepts u but rejects v and w

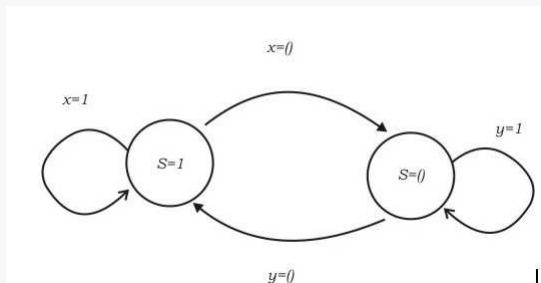
[gate2006-it](#) [theory-of-computation](#) [finite-automata](#) [normal](#)

Answer

5.5.34 Finite Automata: GATE2006-IT-37 [top](#)

<http://gateoverflow.in/3576>

For a state machine with the following state diagram the expression for the next state S^+ in terms of the current state S and the input variables x and y is



- A. $S^+ = S' \cdot y' + S \cdot x$
- B. $S^+ = S \cdot x \cdot y' + S' \cdot y \cdot x'$
- C. $S^+ = x \cdot y'$
- D. $S^+ = S' \cdot y + S \cdot x'$

[gate2006-it](#) [theory-of-computation](#) [finite-automata](#) [normal](#)

Answer

5.5.35 Finite Automata: GATE2007-29 [top](#)

<http://gateoverflow.in/1227>

A minimum state deterministic finite automaton accepting the language
 $L = \{w \mid w \in \{0,1\}^*, \text{ number of } 0s \text{ and } 1s \text{ in } w \text{ are divisible by } 3 \text{ and } 5, \text{ respectively}\}$ has

- A. 15 states
- B. 11 states
- C. 10 states
- D. 9 states

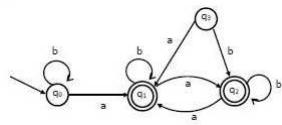
[gate2007](#) [theory-of-computation](#) [finite-automata](#) [normal](#)

Answer

5.5.36 Finite Automata: GATE2007-74 [top](#)

<http://gateoverflow.in/1270>

Consider the following Finite State Automaton:



The language accepted by this automaton is given by the regular expression

- A. $b^*ab^*ab^*ab^*$
- B. $(a+b)^*$
- C. $b^*a(a+b)^*$
- D. $b^*ab^*ab^*$

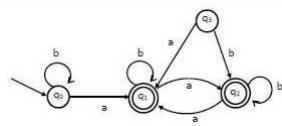
[gate2007](#) [theory-of-computation](#) [finite-automata](#) [normal](#)

[Answer](#)

5.5.37 Finite Automata: GATE2007-75 [top](#)

<http://gateoverflow.in/43514>

Consider the following Finite State Automaton



The minimum state automaton equivalent to the above FSA has the following number of states

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

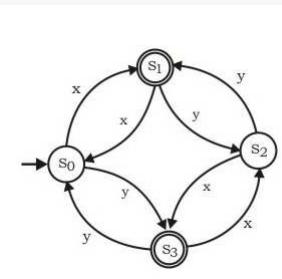
[normal](#) [gate2007](#) [theory-of-computation](#) [finite-automata](#)

[Answer](#)

5.5.38 Finite Automata: GATE2007-IT-47 [top](#)

<http://gateoverflow.in/3489>

Consider the following DFA in which s_0 is the start state and s_1, s_3 are the final states.



What language does this DFA recognize?

- A. All strings of x and y
- B. All strings of x and y which have either even number of x and even number of y or odd number of x and odd number of y
- C. All strings of x and y which have equal number of x and y
- D. All strings of x and y with either even number of x and odd number of y or odd number of x and even number of y

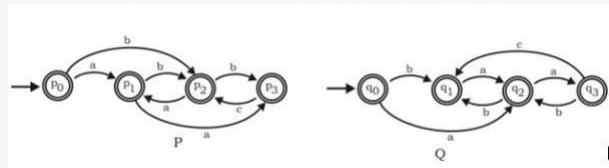
[gate2007-it](#) [theory-of-computation](#) [finite-automata](#) [normal](#)

[Answer](#)

5.5.39 Finite Automata: GATE2007-IT-50 [top](#)

<http://gateoverflow.in/3492>

Consider the following finite automata P and Q over the alphabet $\{a, b, c\}$. The start states are indicated by a double arrow and final states are indicated by a double circle. Let the languages recognized by them be denoted by $L(P)$ and $L(Q)$ respectively.



The automation which recognizes the language $L(P) \cap L(Q)$ is :

- A.
- B.
- C.
- D.

[gate2007-it](#) [theory-of-computation](#) [finite-automata](#) [normal](#)

Answer

5.5.40 Finite Automata: GATE2007-IT-71 [top](#)

<http://gateoverflow.in/3523>

Consider the regular expression $R = (a + b)^* (aa + bb) (a + b)^*$

Which of the following non-deterministic finite automata recognizes the language defined by the regular expression R? Edges labeled λ denote transitions on the empty string.

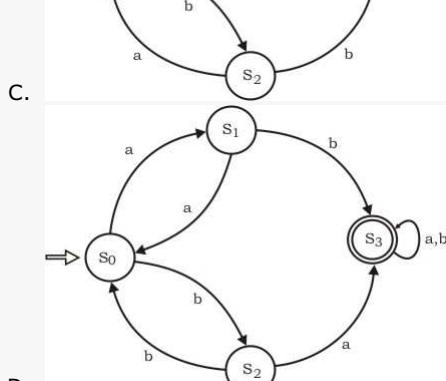
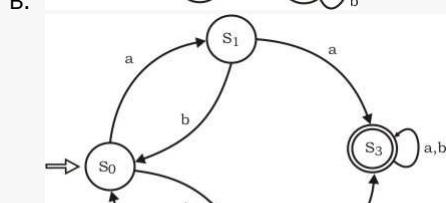
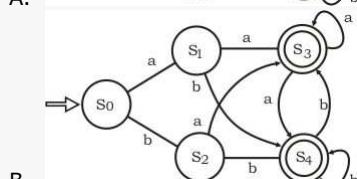
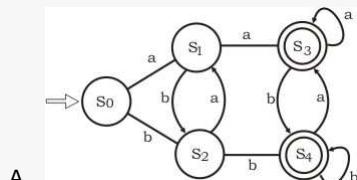
- A.
- B.
- C.
- D.

Answer

5.5.41 Finite Automata: GATE2007-IT-72 [top](#)<http://gateoverflow.in/3524>

Consider the regular expression $R = (a + b)^* (aa + bb) (a + b)^*$

Which deterministic finite automaton accepts the language represented by the regular expression R?



Answer

5.5.42 Finite Automata: GATE2008-49 [top](#)<http://gateoverflow.in/462>

Given below are two finite state automata (\rightarrow indicates the start state and F indicates a final state)

Y:

	a	b
\rightarrow 1	1	2
2 (F)	2	1

Z:

	a	b
\rightarrow 1	2	2
2 (F)	1	1

Which of the following represents the product automaton $Z \times Y$?

A.

	a	b
$\rightarrow P$	S	R
Q	R	S
$R(F)$	Q	P
S	Q	P

B.

	a	b
$\rightarrow P$	S	Q
Q	R	S
$R(F)$	Q	P
S	P	Q

C.

	a	b
$\rightarrow P$	Q	S
Q	R	S
$R(F)$	Q	P
S	Q	P

D.

	a	b
$\rightarrow P$	S	Q
Q	S	R
$R(F)$	Q	P
S	Q	P

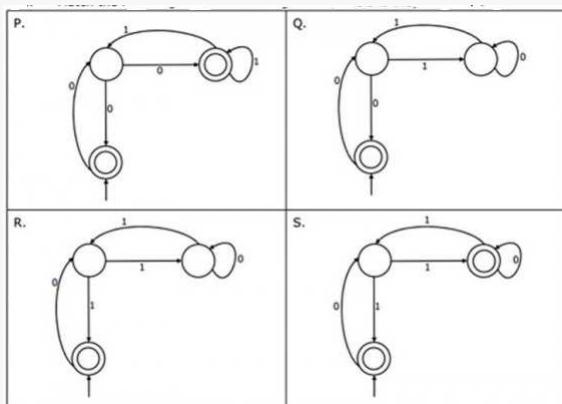
gate2008 normal theory-of-computation finite-automata

Answer

5.5.43 Finite Automata: GATE2008-52 [top](#)

<http://gateoverflow.in/464>

Match the following NFAs with the regular expressions they correspond to

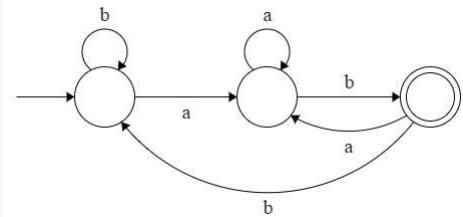


1. $\epsilon + 0(01^*1 + 00)^*01^*$
2. $\epsilon + 0(10^*1 + 00)^*0$
3. $\epsilon + 0(10^*1 + 10)^*1$
4. $\epsilon + 0(10^*1 + 10)^*10^*$

- A. P-2, Q-1, R-3, S-4
- B. P-1, Q-3, R-2, S-4
- C. P-1, Q-2, R-3, S-4
- D. P-3, Q-2, R-1, S-4

[gate2008](#) [theory-of-computation](#) [finite-automata](#) [normal](#)
Answer**5.5.44 Finite Automata: GATE2008-IT-32** [top](#)<http://gateoverflow.in/3342>

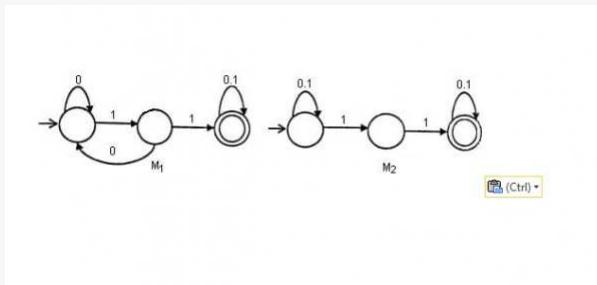
If the final states and non-final states in the DFA below are interchanged, then which of the following languages over the alphabet {a, b} will be accepted by the new DFA?



- A. Set of all strings that do not end with ab
- B. Set of all strings that begin with either an a or a b
- C. Set of all strings that do not contain the substring ab,
- D. The set described by the regular expression $b^*aa^*(ba)^*b^*$

[gate2008-it](#) [theory-of-computation](#) [finite-automata](#) [normal](#)
Answer**5.5.45 Finite Automata: GATE2008-IT-36** [top](#)<http://gateoverflow.in/3346>

Consider the following two finite automata. M_1 accepts L_1 and M_2 accepts L_2 . Which one of the following is TRUE?



- A. $L_1 = L_2$
- B. $L_1 \subset L_2$
- C. $L_1 \cap L_2' = \emptyset$
- D. $L_1 \cup L_2 \neq L_1$

[gate2008-it](#) [theory-of-computation](#) [finite-automata](#) [normal](#)
Answer**5.5.46 Finite Automata: GATE2008-IT-6** [top](#)<http://gateoverflow.in/3266>

Let N be an NFA with n states and let M be the minimized DFA with m states recognizing the same language. Which of the following is NECESSARILY true?

- A. $m \leq 2^n$
- B. $n \leq m$
- C. M has one accept state
- D. $m = 2^n$

[gate2008-it](#) [theory-of-computation](#) [finite-automata](#) [normal](#)
Answer

5.5.47 Finite Automata: GATE2009-27 [top](#)

<http://gateoverflow.in/1313>

Given the following state table of an FSM with two states A and B, one input and one output.

PRESENT STATE A	PRESENT STATE B	Input	Next State A	Next State B	Output
0	0	0	0	0	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	1	0	0
0	0	1	0	1	0
0	1	1	0	0	1
1	0	1	0	1	1
1	1	1	0	0	1

If the initial state is A=0 ,B=0 what is the minimum length of an input string which will take the machine to the state A=0,B=1 with output=1.

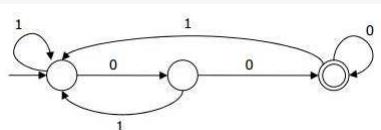
- A. 3
- B. 4
- C. 5
- D. 6

[gate2009](#) [theory-of-computation](#) [finite-automata](#) [normal](#)

[Answer](#)

5.5.48 Finite Automata: GATE2009-41 [top](#)

<http://gateoverflow.in/1327>



The above DFA accepts the set of all strings over {0,1} that

- A. begin either with 0 or 1.
- B. end with 0.
- C. end with 00.
- D. contain the substring 00.

[gate2009](#) [theory-of-computation](#) [finite-automata](#) [easy](#)

[Answer](#)

5.5.49 Finite Automata: GATE2010-41 [top](#)

<http://gateoverflow.in/2342>

Let w be any string of length n in $\{0,1\}^*$. Let L be the set of all substrings of w . What is the minimum number of states in non-deterministic finite automaton that accepts L ?

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

gate2010 theory-of-computation finite-automata normal

Answer

5.5.50 Finite Automata: GATE2011_42 [top](#)

<http://gateoverflow.in/2144>

Definition of a language L with alphabet $\{a\}$ is given as following.

$$L = \{a^{nk} \mid k > 0, \text{ and } n \text{ is a positive integer constant}\}$$

What is the minimum number of states needed in a DFA to recognize L ?

- (A) $k + 1$
- (B) $n + 1$
- (C) 2^{n+1}
- (D) 2^{k+1}

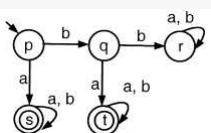
gate2011 theory-of-computation finite-automata normal

Answer

5.5.51 Finite Automata: GATE2011_45 [top](#)

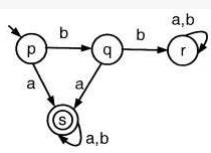
<http://gateoverflow.in/2147>

A deterministic finite automaton (DFA) D with alphabet $\Sigma = \{a,b\}$ is given below.

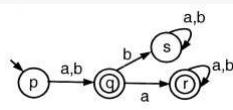


Which of the following finite state machines is a valid minimal DFA which accepts the same language as D ?

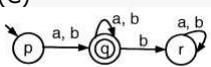
(A)



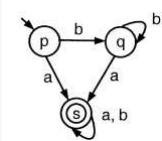
(B)



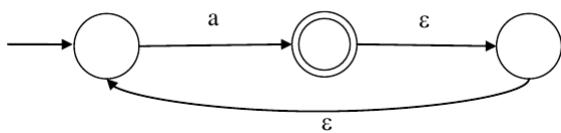
(C)



(D)


[gate2011](#) [theory-of-computation](#) [finite-automata](#) [easy](#)
Answer**5.5.52 Finite Automata: GATE2012_12** [top](#)<http://gateoverflow.in/44>

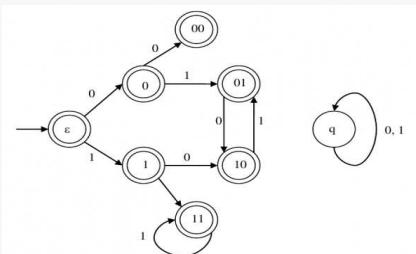
What is the complement of the language accepted by the NFA shown below?
Assume $\Sigma = \{a\}$ and ϵ is the empty string.



- (A) \emptyset
- (B) $\{\epsilon\}$
- (C) a^*
- (D) $\{a, \epsilon\}$

[gate2012](#) [finite-automata](#) [easy](#) [theory-of-computation](#)
Answer**5.5.53 Finite Automata: GATE2012_46** [top](#)<http://gateoverflow.in/2159>

Consider the set of strings on $\{0,1\}$ in which, every substring of 3 symbols has at most two zeros. For example, 001110 and 011001 are in the language, but 100010 is not. All strings of length less than 3 are also in the language. A partially completed DFA that accepts this language is shown below.



The missing arcs in the DFA are

	00	01	10	11	q
00	1	0			
01				1	
10	0				
11			0		

	00	01	10	11	q
00		0			1
01		1			
10				0	
11		0			

	00	01	10	11	q
00		1			0
01		1			
10			0		
11		0			

	00	01	10	11	q
00		1			0
01				1	
10	0				
11		0			

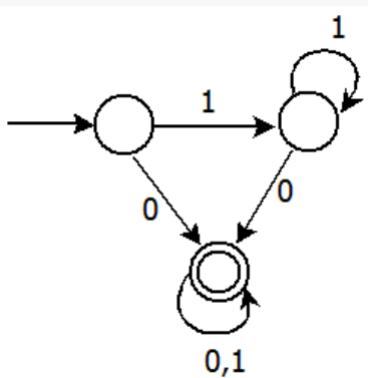
gate2012 theory-of-computation finite-automata normal

Answer

5.5.54 Finite Automata: GATE2013_33 top

<http://gateoverflow.in/1544>

Consider the DFA A given below.



Which of the following are **FALSE**?

1. Complement of $L(A)$ is context-free.
2. $L(A) = L((11^*0 + 0)(0 + 1)^*0^*1^*)$
3. For the language accepted by A, A is the minimal DFA.
4. A accepts all strings over $\{0, 1\}$ of length at least 2.

- (A) 1 and 3 only
 (B) 2 and 4 only
 (C) 2 and 3 only
 (D) 3 and 4 only

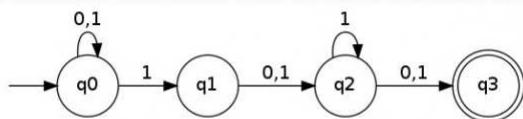
gate2013 theory-of-computation finite-automata normal

[Answer](#)

5.5.55 Finite Automata: GATE2014-1-16 [top](#)

<http://gateoverflow.in/1782>

Consider the finite automaton in the following figure:



What is the set of reachable states for the input string 0011?

- A. $\{q_0, q_1, q_2\}$
 B. $\{q_0, q_1\}$
 C. $\{q_0, q_1, q_2, q_3\}$
 D. $\{q_3\}$

gate2014-1 theory-of-computation finite-automata easy

[Answer](#)

5.5.56 Finite Automata: GATE2015-1_52 [top](#)

<http://gateoverflow.in/8362>



Consider the DFAs M and N given above. The number of states in a minimal DFA that accept the language $L(M) \cap L(N)$ is _____.

gate2015-1 theory-of-computation finite-automata easy numerical-answers

[Answer](#)

5.5.57 Finite Automata: GATE2015-2_53 [top](#)

<http://gateoverflow.in/8256>

The number of states in the minimal deterministic finite automaton corresponding to the regular expression $(0 + 1)^*(10)$ is _____.

gate2015-2 theory-of-computation finite-automata normal numerical-answers

[Answer](#)

5.5.58 Finite Automata: GATE2015-3_18 [top](#)

<http://gateoverflow.in/8415>

Let L be the language represented by the regular expression $\Sigma^*0011\Sigma^*$ where $\Sigma = \{0,1\}$. What is the minimum number of states in a DFA that recognizes \bar{L} (complement of L)?

- A. 4
 B. 5
 C. 6
 D. 8

[gate2015-3](#) [theory-of-computation](#) [finite-automata](#) [normal](#)
Answer**5.5.59 Finite Automata: GATE2017-1-22** [top](#)<http://gateoverflow.in/118302>

Consider the language L given by the regular expression $(a+b)^*b(a+b)$ over the alphabet $\{a,b\}$. The smallest number of states needed in a deterministic finite-state automaton (DFA) accepting L is _____.

[gate2017-1](#) [theory-of-computation](#) [finite-automata](#) [numerical-answers](#)
Answer**5.5.60 Finite Automata: GATE2017-2-25** [top](#)<http://gateoverflow.in/118160>

The minimum possible number of states of a deterministic finite automaton that accepts the regular language $L = \{w_1aw_2 \mid w_1, w_2 \in \{a,b\}^*, |w_1| = 2, |w_2| \geq 3\}$ is _____.

[theory-of-computation](#) [gate2017-2](#) [finite-automata](#) [numerical-answers](#)
Answer**5.5.61 Finite Automata: GATE2017-2-39** [top](#)<http://gateoverflow.in/118384>

Let δ denote the transition function and $\hat{\delta}$ denote the extended transition function of the ϵ -NFA whose transition table is given below:

δ	ϵ	a	b
$\rightarrow q_0$	$\{q_2\}$	$\{q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_2\}$	$\{q_3\}$
$\{q_2\}$	$\{q_0\}$	\emptyset	\emptyset
q_3	\emptyset	\emptyset	$\{q_2\}$

Then $\hat{\delta}(q_2, aba)$ is

- A. \emptyset
- B. $\{q_0, q_1, q_3\}$
- C. $\{q_0, q_1, q_2\}$
- D. $\{q_0, q_2, q_3\}$

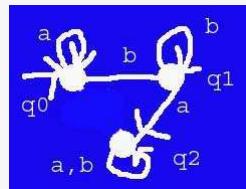
[gate2017-2](#) [theory-of-computation](#) [finite-automata](#)
Answer**5.5.62 Finite Automata: ISI2011-CS-4b** [top](#)<http://gateoverflow.in/48157>

Suppose $M = (Q, \Sigma, \delta, q_0, F)$ is a deterministic finite automaton, and suppose there exists a state $q \in Q$, a string $z \in \Sigma$, and integers $i, j > 0$ such that $\delta(q, z^i) = \delta(q, z^j) = q$. Prove that $\delta(q, z^{\gcd(i,j)}) = q$.

[descriptive](#) [isi2011](#) [theory-of-computation](#) [finite-automata](#)
Answer**5.5.63 Finite Automata: ISI2014-CS-4a** [top](#)<http://gateoverflow.in/47443>

Construct a deterministic finite automaton accepting the following language: $\{w \in \{0,1\}^* : w \text{ has an equal number of } 01\text{'s and } 10\text{'s}\}$. For example, 101 is in the language because it contains one instance of 10 and one instance of 01 as well.

[descriptive](#) [isi2014](#) [theory-of-computation](#) [finite-automata](#)
Answer**Answers: Finite Automata****5.5.1 Finite Automata: CMI2010-B-04b** [top](#)<http://gateoverflow.in/47073>



consider above dfa in which

initial state = q_0

final states = q_0, q_1

trap state = q_2

language generated by above dfa is a^*b^* which is a infinite language

$n = 3$ (number of states)

now consider string $X = \{aaabbbb\}$ i.e a^3b^4

$|X| = 7$

now $2n = 2^3 = 6$

$3n = 3^3 = 9$

$2n < |x| < 3n \implies 6 < 7 < 9$

hence proved

2 votes

-- Tauhin Gangwar (9k points)

5.5.2 Finite Automata: CMI2010-B-04c [top](#)



Selected Answer

If a language L is accepted by an NFA with n states then there is a DFA with no more than 2^n states accepting L .

This is correct statement. with [Proof.](#)

2 votes

-- Manoj Kumar (37k points)

5.5.3 Finite Automata: GATE 2016-2-42 [top](#)



Selected Answer

I , False, as in NFA, it is not necessary that all states have transitions for all symbols.

II True, there exists a regular language

$A = \{\}$, such that for all languages

B ,

$A \cap B = \{\}$ is regular

So answer is option B.

38 votes

-- Praveen Saini (53k points)

5.5.4 Finite Automata: GATE 2016-2_16 [top](#)

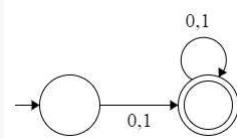
<http://gateoverflow.in/39562>



All strings over $\{0,1\}$ having length ≥ 1

$$(0+1)^*(0+1)(0+1)^* = (0+1)(0+1)^* = (0+1)^*(0+1) = (0+1)^+$$

having DFA



No of states in minimal DFA = 2

26 votes

-- Praveen Saini (53k points)

5.5.5 Finite Automata: GATE1987-2j top

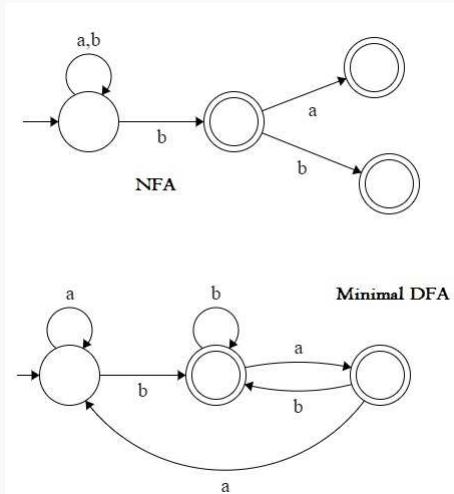
<http://gateoverflow.in/80594>



No a minimal DFA that is equivalent to an NFA with n nodes has always 2^n states.

Correct statement is A minimal DFA that is equivalent to an NFA with n nodes has atmost 2^n states.

Example :



4 votes

-- Prashant Singh (47.6k points)

5.5.6 Finite Automata: GATE1991_17,b top

<http://gateoverflow.in/544>



Answer :-

Check following NFA. I've done subset construction too. 8 States are needed even after minimization..

Every state containing D is final state.



Third symbol from the right is a '1'. So, we can also consider Myhill-Nerode theorem here. Intuitively we need to remember the last 3 bits of the string each of which forms a different equivalence class as per Myhill-Nerode theorem as shown by the following table. Here, for any set of strings (in a row), we distinguish only the rows above it - as the relation is symmetric. Further strings in the language and not in the language are distinguished separately as ϵ distinguishes them.

Last 3 bits	Distinguishing string	In the language?
1000		N
2001	"00" - distinguishes from strings in 1.	N
3010	"0" - distinguishes from strings in 1 and 2. "00" distinguishes from strings in 4.	N
4011	"0" - distinguishes from strings in 1 and 2. "00" distinguishes from strings in 3.	N
5100		Y
6101	"00" distinguishes from strings in 5.	Y
7110	"0" distinguishes from strings in 5 and "00" distinguishes from strings in 6	Y
8111	"00" distinguishes from strings in 5 and 7. "0" distinguishes from strings in 6.	Y

6 votes

-- Akash (41.5k points)

5.5.7 Finite Automata: GATE1993_27 top

<http://gateoverflow.in/2323>

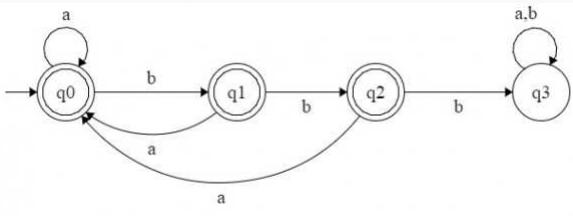


Selected Answer

Design a DFA that accepts all strings contain bbb

regular expression $(a + b)^*bbb(a + b)^*$

then take complement of DFA such that no string has 3 consecutive occurrences of the letter b.



having regular expression $(a + ba + bba)^*(\epsilon + b + bb)$

14 votes

-- Praveen Saini (53k points)

5.5.8 Finite Automata: GATE1994_3.3 [top](#)

<http://gateoverflow.in/2480>

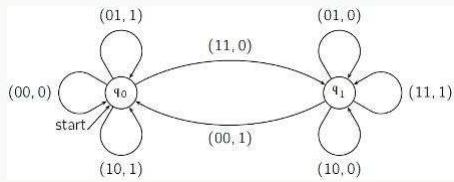


Selected Answer

Finite Automata (FA) or Finite State Machine to add two integers can be constructed using two states:

- q_0 : Start state to represent carry bit is 0
- q_1 : State to represent carry bit is 1

The inputs to FA will be pair of bits i.e. 00, 01, 10, and 11



The FA starts in state 1 (since carry is 0) and inputs a pair of bits. If the pair is 11, the FA outputs a 0 and switches to state 2 (since the carry is 1), where the next pair of bits is input and is added to a carry bit of 1.

Example: Consider the addition of 52 and 21

110100 - (binary representation of 52)

010101 - (binary representation of 21)

Since adding numbers is done from right to left, The first input symbol is 01, representing a 0 in the rightmost (binary) digit of 52 and a 1 in the rightmost digit of 21. The machine enters state q_0 (since there is no carry) and outputs a 1. The next input is 00 because both numbers have zero as the second rightmost digit. The machine enters state q_0 and outputs 0. The next input is 11. The machine enters state q_1 (since the carry is 1) and outputs 0. Being in state q_1 means that there is a carry from this position into the next. And the remaining bits can be worked out to get 1001001 (i.e. 73).

10 votes

-- Saurav Kumar Gupta (2.1k points)

No. Perhaps that wont be possible to add any two arbitrary numbers because that will need a memory element which is not there in a FSM.

11 votes

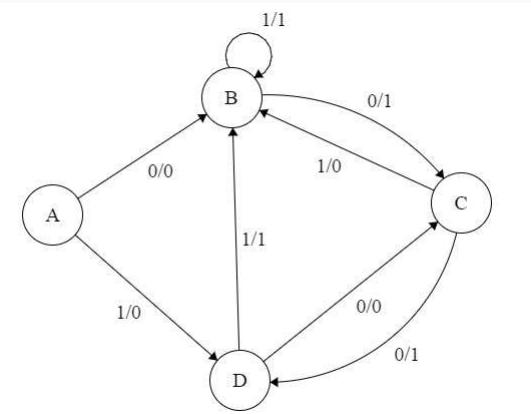
-- Gate Keeda (18.8k points)

5.5.9 Finite Automata: GATE1995_2.23 [top](#)

<http://gateoverflow.in/2638>



Selected Answer



if A is start state , shortest sequence is 10 or 00 to reach C

if B is start state , shortest sequence is 0 to reach C

if C is start state , shortest sequence is 10 or 00 to reach C

if D is start state , shortest sequence is 0 to reach C

b) is correct.

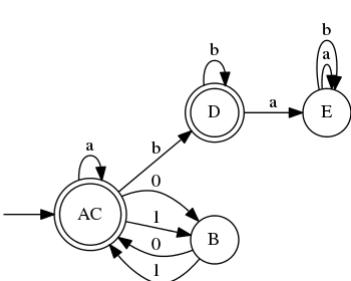
17 votes

-- Praveen Saini (53k points)

5.5.10 Finite Automata: GATE1996_12 [top](#)

<http://gateoverflow.in/2764>

We can combine the final state of M_1 with the start state of M_2 as follows recognizing $L_1 L_2$



3 votes

-- Arjun Suresh (287k points)

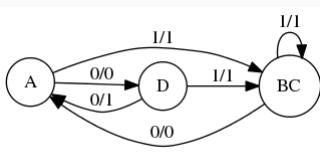
5.5.11 Finite Automata: GATE1996_2.23 [top](#)

<http://gateoverflow.in/2752>



Selected Answer

3 states are required in the minimized machine. States B and C can be combined as follows:



8 votes

-- Arjun Suresh (287k points)

5.5.12 Finite Automata: GATE1997_20 [top](#)

<http://gateoverflow.in/2280>

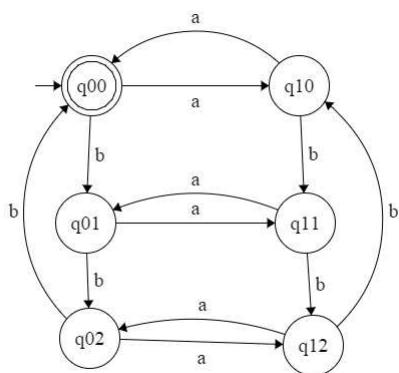


A state q_{xy} means $n_a \bmod 2 = x, n_b \bmod 3 = y$

q_{00} means $n_a \bmod 2 = 0, n_b \bmod 3 = 0$ [no of a's is divisible of 2 and no of b's are divisible of 3]

$q_{00} \xrightarrow{a} q_{10}$

$q_{00} \xrightarrow{b} q_{01}$ and so on



9 votes

-- Praveen Saini (53k points)

5.5.13 Finite Automata: GATE1997_21 [top](#)

<http://gateoverflow.in/2281>

Suppose we have a finite automation for L , then we can build a finite automation for L^P by marking all the states from which final state is reachable as the final states for new automaton, the reasoning is that suppose we can reach final state f from some state q , then that means there exists some string s' that takes automation from q to f , so if there is some string s that takes automation to state q from start state this string should belong to the new language L^P . (L^P is the set of all prefix strings for the string in L)

Also, we can obtain an automation for L^R by swapping the start and final states of original automation L and by reversing all the edges in the DFA.

11 votes

-- Omesh Pandita (2.7k points)

5.5.14 Finite Automata: GATE1998_1.10 [top](#)

<http://gateoverflow.in/1647>



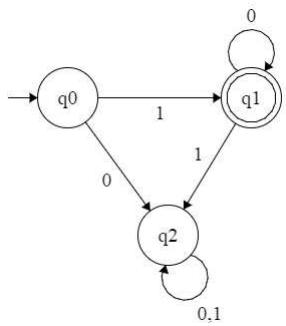
Option A is correct

A. is regular

$L = \{1, 10, 100, 1000, 10000, \dots\}$

regular expression 10^*

DFA :



B . $L = \{1, 11, 1111, 11111111, \dots\} = \{1^i \mid i = 2^n, n \geq 0\}$ is non regular language

C. Equal- Equal, is CFL, and non regular

D, $L = \{1^i 0 1^i \mid i > 0\} \cup \{1\}$ is also CFL, and non regular

19 votes

-- Praveen Saini (53k points)

5.5.15 Finite Automata: GATE1998_2.5 [top](#)

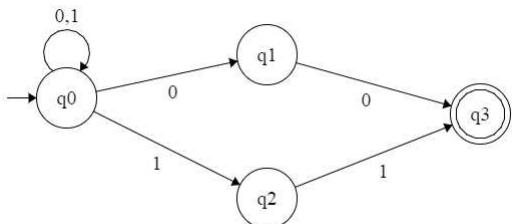
<http://gateoverflow.in/1677>



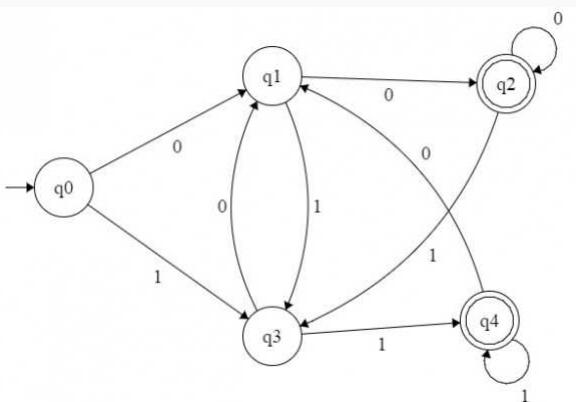
Selected Answer
binary strings whose last two symbols are same

regular expression = $(0+1)^*(00+11)$

Having NFA



Equivalent DFA



Total no of states = 5

12 votes

-- Praveen Saini (53k points)

5.5.16 Finite Automata: GATE1998_4 top

<http://gateoverflow.in/1695>

Selected Answer

suppose we have decimal no 3 after that we get a symbol 2 . it becomes 32 as $3 \times 10 + 2 = 32$

in binary if we have 10 (i.e 2 in decimal say old no) and after that we get symbol 1 it become 101(i.e 5 in decimal say new no)

$$2 (\text{old no.}) \times 2 (\text{base}) + 1 (\text{input symbol}) = 5 (\text{new no.})$$

Now in the given problem , binary no is divisible by 5 , i.e 0,101,1010,1111.....

We need 5 states in DFA , 0,1,2,3 and 4 .Each state represent remainder that comes when we divide no by 5.

$$\text{input symbol} = \{0,1\}$$

We get the transition

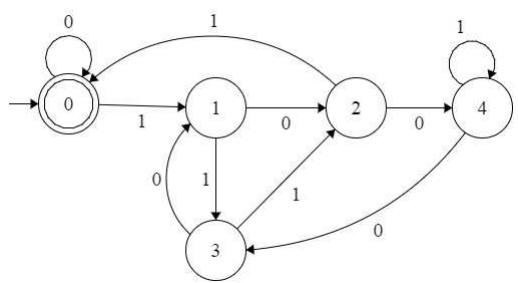
$$[\text{Old state } x \text{ base} + \text{input symbol}] \bmod 5 = \text{New state} , \text{ where base is 2}$$

$$[0 \times 2 + 0] \bmod 5 = 0 \quad [1 \times 2 + 0] \bmod 5 = 2 \quad [2 \times 2 + 0] \bmod 5 = 4$$

$$[0 \times 2 + 1] \bmod 5 = 1 \quad [1 \times 2 + 1] \bmod 5 = 3 \quad [2 \times 2 + 1] \bmod 5 = 0$$

$$[3 \times 2 + 0] \bmod 5 = 1 \quad [4 \times 2 + 0] \bmod 5 = 3$$

$$[3 \times 2 + 1] \bmod 5 = 2 \quad [4 \times 2 + 1] \bmod 5 = 4$$



10 votes

-- Praveen Saini (53k points)

5.5.17 Finite Automata: GATE1999_1.4 top

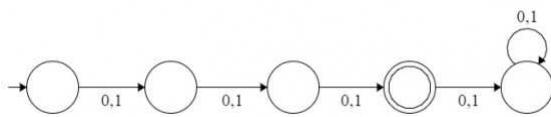
<http://gateoverflow.in/1445>

Selected Answer

As far as for above problem say regular expression for $(0+1)(0+1)\dots 3 \text{ times}$

$$= (0+1)(0+1)(0+1)$$

Having DFA



so for regular expression $(0+1)(0+1)\dots N$ times have **N+2** states in DFA

$N+1$ states in NFA (we can remove dead state)

When question is about minimum state finite automata (and nothing is mentioned NFA/DFA) then which ever having minimum no.

$N+1$ states.

18 votes

-- Praveen Saini (53k points)

5.5.18 Finite Automata: GATE2001-1.6 [top](#)



Selected Answer

ans is B. 2^N .

In DFA any subset of the N states (for N element set 2^N subsets possible) can become a new state and they can remain even when the DFA is minimized. So, maximum we can get 2^N states for the minimized DFA. (at least in question must be a typo for at most).

7 votes

-- jayendra (7.7k points)

5.5.19 Finite Automata: GATE2001-2.5 [top](#)



Selected Answer

Answer is D. It can be proved using Myhill Nerode theorem. We need a separate state for $\#a \bmod 6 = 0..5$ and $\#b \bmod 8 = 0..7$. Each combination of them must also be a new state giving $6*8 = 48$ minimum states required in the DFA.

Reading Myhill-Nerode theorem might be confusing though it is actually very simple.

<http://courses.cs.washington.edu/courses/cse322/05wi/handouts/MyhillNerode.pdf>

11 votes

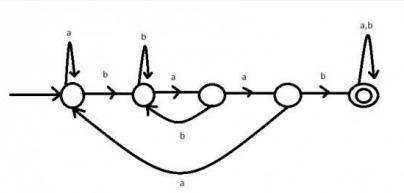
-- Arjun Suresh (287k points)

5.5.20 Finite Automata: GATE2001-5 [top](#)



Selected Answer

DFA for A:



9 votes

-- jayendra (7.7k points)

5.5.21 Finite Automata: GATE2002-2.13 [top](#)



Selected Answer

it is 3 states as we need a state each for length mod 3 = 0, 1 and 2.

9 votes

-- priya023 (217 points)

5.5.22 Finite Automata: GATE2002-2.5 [top](#)

<http://gateoverflow.in/835>



Selected Answer

answer should be option (A).

option (B) and (C) are clearly wrong . it says input 11 then o/p 01 and i/p 10 then o/p 00 but here at single bit o/p is 2 bit sequence

now for option (a) we can trace out .. suppose string is 0111

at A---0---> A---1---> B--1-->C---1-->C

O/P 00 01 10 10

we can see here at (A,0)---> (A,00) which sum of 0+0=00, (previous i/p bit + present i/p bit)

(A,1)--->(B,01) which is sum of 0+1= 1=01,

(B,1)--->(C,10) which is sum of 1+1= (previous i/p bit + present i/p bit)=10 ,

(C,1)---> (C,10) which is sum of 1+1=10

so answer should be (A).

14 votes

-- sonam vyas (12.8k points)

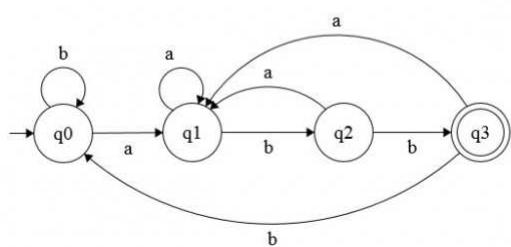
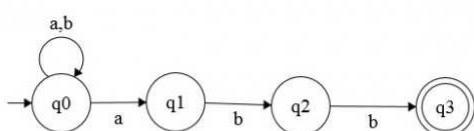
5.5.23 Finite Automata: GATE2002-21 [top](#)

<http://gateoverflow.in/874>



Selected Answer

NFA for regular expression $(a+b)^*abb$ and its equivalent DFA will be as follow:



9 votes

-- Praveen Saini (53k points)

5.5.24 Finite Automata: GATE2003-50 [top](#)

<http://gateoverflow.in/939>



Selected Answer

Language of above DFA is all strings over $\{0,1\}$ that contain substring 001.

Regular expression of above DFA is $(0+1)^*00\underline{1}(0+1)^*$

1 that is underlined can not be first bit of 7-bit binary no, but can be fourth bit or last bit.

Case 1: if it is 4th bit ,then possible set of strings can be

First 001 twobits Last = $1\ 001(00+01+10+11)1 = 4$ strings

Case 2 : if it is last bit, then possible set of strings can be

First twobits fourth 001 = $1(00+01+10+11)1\ 001 = 4$ strings

String common in both cases **1001001**

Total strings = $4 + 4 - 1 = 7$ strings

19 votes

-- Praveen Saini (53k points)

5.5.25 Finite Automata: GATE2003-55 [top](#)

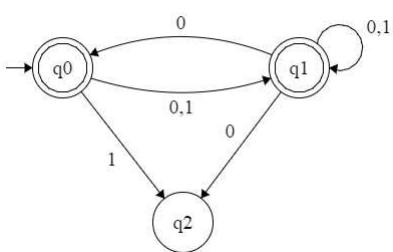
<http://gateoverflow.in/943>



Selected Answer

Answer is B .

As the problem said



as in above NFA language L_1 is $\{0,1\}^*$. [we don't know L , we need not to find out]

option A is wrong as L is accepting 1 and L_1 is also accepting 1

option C is wrong as L_1 accepting $^\wedge$,null, but L is not .

Option D is wrong for same reason as option C is wrong.

17 votes

-- Praveen Saini (53k points)

5.5.26 Finite Automata: GATE2004-86 [top](#)

<http://gateoverflow.in/1080>



Selected Answer

Option c and option d are cancelled out clearly because with 3 1s we can reach the final state ...there is an string where we can reach the final state by 6 1's now 6 is nt odd but it is divisible by 3 hence option a is correct

8 votes

-- Bhagirathi Nayak (13.1k points)

5.5.27 Finite Automata: GATE2004-IT-41 [top](#)

<http://gateoverflow.in/3684>



Selected Answer

$$\delta(A, a) = A, \quad A \rightarrow aA$$

$\delta(A, b) = B, \quad A \rightarrow bB$
 $\delta(B, a) = B, \quad B \rightarrow aB$
 $\delta(B, b) = A, \quad B \rightarrow bA$
 B is final state so $B \rightarrow \epsilon$

12 votes

-- Praveen Saini (53k points)

5.5.28 Finite Automata: GATE2005-53 [top](#)

<http://gateoverflow.in/1376>



Selected Answer

A is Wrong, since abbb is accepted. (1 a is followed by more than 2 bs)

C is Wrong, since abba contains abb as substring, but is still not accepted.

D is Wrong, since ab does not contain aa as substring, but is still not accepted.

Hence correct answer is B.

14 votes

-- saurabhrk (1.5k points)

5.5.29 Finite Automata: GATE2005-63 [top](#)

<http://gateoverflow.in/1386>



Selected Answer

answer = **option B**

for any binary no, FSM read input from LSB and remain unchanged till first 1, and it complement after that

100 -> 100 [1's complement of 100 + 1 = 011+ 1 = 100 = 2's complement of 100]

010 -> 110 [1's complement of 010 + 1 = 101 + 1 = 110 = 2's complement of 010]

1010100 -> 0101100 [1's complement of 1010100 + 1 = 0101011+ 1 = 0101100]

Note : Underline part is unchanged (till first 1 from lsb) then 0's changed to 1's and 1's changed to 0's

9 votes

-- Praveen Saini (53k points)

5.5.30 Finite Automata: GATE2005-IT-37 [top](#)

<http://gateoverflow.in/3784>



Selected Answer

In Qs. Z -> Y (0 edge)

L1 can have 00 string while L2 can't. L2 can have 01 while L1 can't

So we can conclude neither they are same set not proper subset of each other.

Hence Ans. D.

13 votes

-- shreya ghosh (3.4k points)

5.5.31 Finite Automata: GATE2005-IT-39 [top](#)

<http://gateoverflow.in/3786>



Selected Answer

$S \rightarrow Xa \mid Ya$

$X \rightarrow Za$
 $Z \rightarrow Sa \mid \epsilon$
 $Y \rightarrow Wa$
 $W \rightarrow Sa$

This is left linear grammar having language L . Convert it into right linear using following rule :

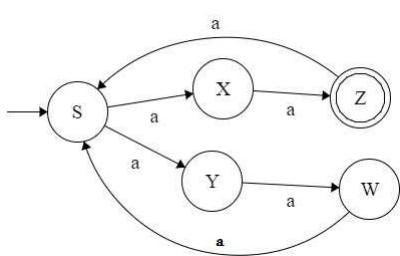
- The previous slide reversed the language!

$V_i \rightarrow V_j w$	Reverses to	$V_i \rightarrow w^R V_j$
$V_i \rightarrow w$	Reverses to	$V_i \rightarrow w^R$
- If the left linear grammar produced language L , then the resulting right linear grammar produces L^R .

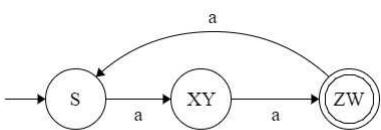
$S \rightarrow aX \mid aY$
 $X \rightarrow aZ$
 $Z \rightarrow aS \mid \epsilon$
 $Y \rightarrow aW$
 $W \rightarrow aS$

is right linear grammar having language L^R .

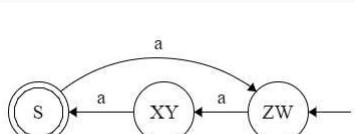
having NFA



Having DFA for language L^R



DFA for language L (reversal)



$L = \{ w : n_a(w) \bmod 3 = 2, w \text{ belong to } \{a,b\}^* \}$ same as Omesh Pandita answered

having 3 states

11 votes

-- Praveen Saini (53k points)

5.5.32 Finite Automata: GATE2006-34 top

<http://gateoverflow.in/1291>



Selected Answer

Given language $L = (111 + 11111)^*$

Strings , that belongs in the language

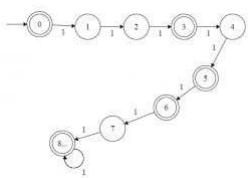
$L = \{\text{null}, 111, 11111, 111111, 11111111, 111111111, \dots \text{ form string length 8 , (number of 1's) , you can generate any length of string from length 3 and 5 (i.e. length 8 ,length 9, length 10 ,length 11 ,.....etc)}\}$

$L = \{\text{null}, 111, 11111, 111111, 11111111, 111111111\}$

Strings in length , that belongs in the language

$L = \{0, 3, 5, 6, 8, 9, 10, 11, \dots\}$

so , required DFA will be ,



So , there are 5 states are final states and 4 states are non-final states ,total number of states are 9 states . hence option D is true.

15 votes

-- Mithlesh Upadhyay (5.3k points)

there will be 9 states. with 1st,4th,6th,7th and 9th state being the final states. with a loop on the ninth state.

$\{0,3,5,6,8,9,10,11,12,\dots\}$ this is the set which will be accepted by the min dfa for this expression.

10 votes

-- Gate Keeda (18.8k points)

5.5.33 Finite Automata: GATE2006-IT-3 [top](#)

<http://gateoverflow.in/3542>



Selected Answer

for u	for v	for w
$\delta(s,abbaba)$	$\delta(s,bab)$	$\delta(s,aabb)$
$\vdash \delta(x,bbaba)$	$\vdash \delta(t,ab)$	$\vdash \delta(x,abb)$
$\vdash \delta(x,baba)$	$\vdash \delta(t,b)$	$\vdash \delta(s,bb)$
$\vdash \delta(x,aba)$	$\vdash s - \text{rejected}$	$\vdash \delta(t,b)$
$\vdash \delta(s,ba)$		$\vdash \delta(t,s)$
$\vdash \delta(t,a)$		$\vdash s - \text{rejected}$
$\vdash t$		-
		accepted

12 votes

-- Praveen Saini (5.3k points)

5.5.34 Finite Automata: GATE2006-IT-37 [top](#)

<http://gateoverflow.in/3576>



Selected Answer

Answer is (A)

For S is 1 only when: Either (S=1 and x=1) OR (s=0 and y=0)

Therefor $S(\text{next}) = S'y' + Sx$

8 votes

-- Sandeep_Uniyal (7.3k points)

Present State	Inputs		Next state
S	x	y	S^+
0	X	0	1
0	X	1	0
1	0	X	0
1	1	X	1

from above table

$$S^+ = S'y' + Sx$$

14 votes

-- Praveen Saini (53k points)

5.5.35 Finite Automata: GATE2007-29 [top](#)

<http://gateoverflow.in/1222>



Selected Answer

Answer will be (A) 15 states.

We need a separate state for #0 = 0, 1, 2 and #1 = 0, 1, 2, 3, 4 giving total minimum number of states = $3 * 5 = 15$.

This is a direct consequence of Myhill-Nerode theorem.

<http://courses.cs.washington.edu/courses/cse322/05wi/handouts/MyhillNerode.pdf>

6 votes

-- Arjun Suresh (287k points)

5.5.36 Finite Automata: GATE2007-74 [top](#)

<http://gateoverflow.in/1270>



Selected Answer

The answer for 74 is C.

You can see that both the states, Q1 and Q2 are final and are accepting $(a + b)^*$.

9 votes

-- Gate Keeda (18.8k points)

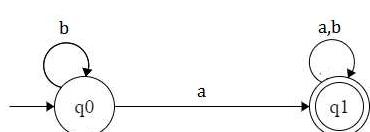
5.5.37 Finite Automata: GATE2007-75 [top](#)

<http://gateoverflow.in/43514>



Selected Answer

Ans will be 2 state



7 votes

-- Manoj Kumar (37k points)

5.5.38 Finite Automata: GATE2007-IT-47 [top](#)

<http://gateoverflow.in/3489>



Selected Answer

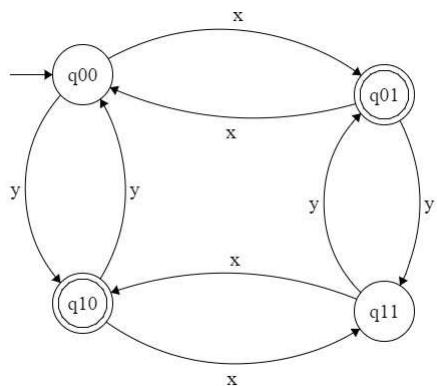
Above DFA can be redesigned as [S0 as q00, S1 as q10 , S2 as q11, S3 as q01]

where , each state as q_{ab} [$a = n_a \text{mod} 2$, $b = n_b \text{mod} 2$]

$q00$ as $n_a \text{mod} 2 = 0$ $n_b \text{mod} 2 = 0$ [no of x is even no of y is even]

and $\partial(q00, x) \rightarrow q10$ [$(0+1) \text{mod} 2 = 1$ as x increase from 0 to 1] $\partial(q00, y) \rightarrow q01$

and $\delta(q_{10}, x) \rightarrow q_{00}$ [$(1+1)\bmod 2 = 0$] $\delta(q_{00}, y) \rightarrow q_{01}$ and soon



q_{01} is final state mean where no of x is even and no of y is odd

q_{10} is final state mean where no of x is odd and no of y is even.

so D is correct answer

10 votes

-- Praveen Saini (53k points)

5.5.39 Finite Automata: GATE2007-IT-50 [top](#)

<http://gateoverflow.in/3492>



Design a DFA using P and Q having p_0q_0 as start state

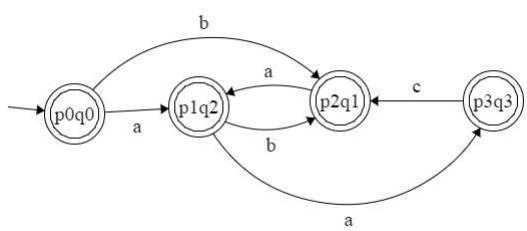
$$\delta(p_0q_0, a) \rightarrow \delta(p_0, a) \cup \delta(q_0, a)$$

$Q \setminus \Sigma$	a	b	c
$\rightarrow p_0q_0^*$	p_1q_2	p_2q_1	
$p_1q_2^*$	p_3q_3	p_2q_1	
$p_2q_1^*$	p_1q_2	p_3 (No Need)	
$p_3q_3^*$		q_2 (No Need)	p_2q_1

In case of intersection final state are those where final state of P and final state of Q comes together.

No Need mean when we reach to p_3 (or q_2) then we cannot reach to any final state bcoz we cannot have state of P and Q together (intersection) so no need to show it in diagram [may draw a dead state for it if required]

DFA results in



That is Option A

Note : i) DFA must have transition for each symbol $Q \times \Sigma \rightarrow Q$

15 votes

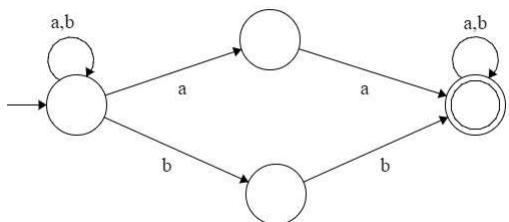
-- Praveen Saini (53k points)

5.5.40 Finite Automata: GATE2007-IT-71 [top](#)

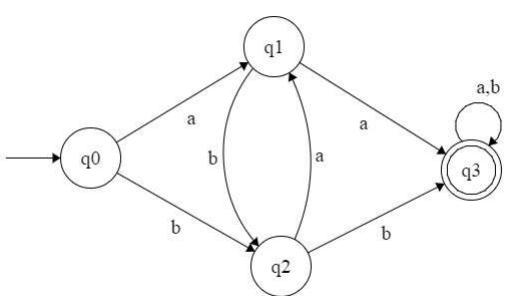
<http://gateoverflow.in/3523>



Design NFA for $R = (a+b)^*(aa+bb)(a+b)^*$



convert NFA to equivalent DFA



A is correct.

16 votes

-- Praveen Saini (53k points)

5.5.41 Finite Automata: GATE2007-IT-72 [top](#)

<http://gateoverflow.in/3524>



Selected Answer

DFA given in option A

having S₃ and S₄ are equivalent states .. that can minimized.

and result in DFA given in

http://gateoverflow.in/3523/gate2007-it_71

11 votes

-- Praveen Saini (53k points)

5.5.42 Finite Automata: GATE2008-49 [top](#)

<http://gateoverflow.in/462>



Selected Answer

	a	b
→11 (P)	12	22
12 (S)	11	21
21 (Q)	22	12
22 (F) (R)	21	11

So, 11 is P and 22 is R in choice. So, the answer should be (A) but in the row for S, it should be P and Q and not Q and P.

12 votes

-- Arjun Suresh (287k points)

5.5.43 Finite Automata: GATE2008-52 [top](#)

<http://gateoverflow.in/464>



Selected Answer

S – 4 is confirmed

R – 3 is true coz everything it accepts ends with 1; this is made mandatory only by 3
this rules out option B and option D

use string 01010 and compare P Vs Q; this makes Q – 2 as confirmed.

hence, **option C** is correct.

6 votes

-- Amar Vashishth (27.9k points)

5.5.44 Finite Automata: GATE2008-IT-32 [top](#)

<http://gateoverflow.in/3342>

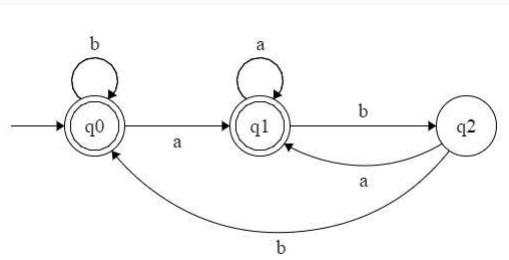


Selected Answer

Above DFA is for regular expression $(a+b)^*ab$. All strings end with ab.

Complement of DFA accepts all strings does not end with ab.

DFA(L') is



B. String begin with either a or b.

ab (string start with a) doesn't accept in it reach to nonfinal state q2.

bab (string start with b) doesn't accept in it reach to nonfinal state q2.

C. Set of strings that do not contain the substring ab

aba (have substring ab) does accept in it reach to final state q1.

D. The set described by the regular expression $b^*aa^*(ba)^*b^*$

b is string accepted by DFA(L') but above regular expression cannot derive it.

Option A is correct.

DFA (L') accepts all strings that doesn't end with ab.

14 votes

-- Praveen Saini (53k points)

5.5.45 Finite Automata: GATE2008-IT-36 [top](#)

<http://gateoverflow.in/3346>



Selected Answer

L1: $(0 + 10)^*11(0 + 1)^*$ L2: $(0 + 1)^*11(0 + 1)^*$ it is quite clear that $L1 = L2$. So, option (A)

18 votes

-- Vicky Bajoria (4.7k points)

5.5.46 Finite Automata: GATE2008-IT-6 [top](#)

<http://gateoverflow.in/3266>



Selected Answer

A state in a DFA will be a subset of the set of states of the equivalent NFA. So, the maximum number of states in the equivalent DFA of an NFA, will be 2^n , where n is the number of states in NFA, as a set with n items has maximum 2^n subsets.

So, answer here is (A).

11 votes

-- Arjun Suresh (287k points)

5.5.47 Finite Automata: GATE2009-27 [top](#)

<http://gateoverflow.in/1313>



Selected Answer

From above table , We look at Next state part

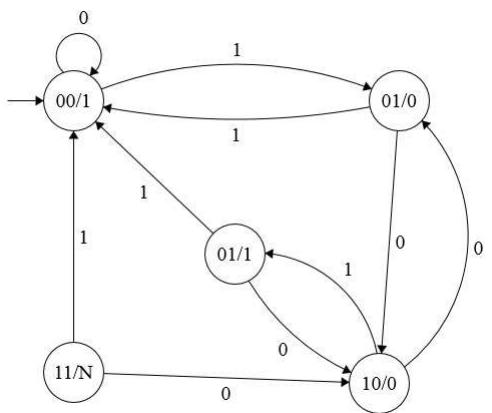
whenever we reach state 00 we get output 1 [at row 1, row 6, row 8], **so we have state 00 with output 1**

when we reach at state 01, we get output 0 [at row 3, row 5] and output 1 [row 7], **so we have two state 01 with output 0 , 01 with output 1**

when we reach at state 10, we get output we get output 0 [at row 2, row 4], **so we have state 10 with output 0**.

We dont reach at state 11 [11 is not there in next state part], but **we have state 11 with don't know (N) output**.

if we draw the Moore Machine for above FSM [from the table: present state x input symbol -> next state]



It is clear from FSM from state 00 to reach state 01 with output 1 i.e., $01/1$ with need **minimum length input 101**
minimum length of input = length of 101 that is 3

8 votes

-- Praveen Saini (53k points)

5.5.48 Finite Automata: GATE2009-41 [top](#)

<http://gateoverflow.in/1327>

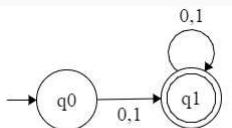


Selected Answer

A) begin either with 0 or 1

Regular expression $(0+1)(0+1)^*$

[begin with 0 or 1 Anything]



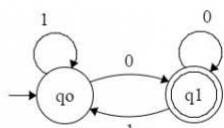
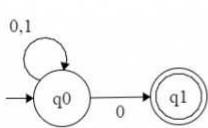
b) end with 0

regular expression = $(0+1)^*0$

[Anything end with 0]

NFA

Equivalent DFA



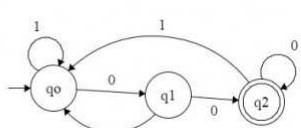
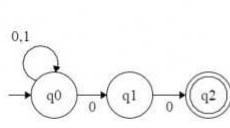
c) end with 00

regular expression $(0+1)^*00$

[Anything end with 00]

NFA

Equivalent DFA

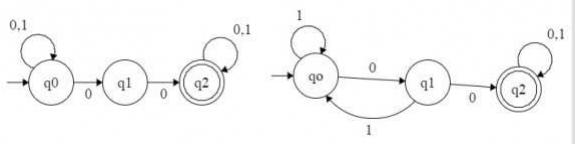


d) containing the substring

regular expression = $(0+1)^*00(0+1)^*$ [Anything substring 00 Anything]

NFA

Equivalent DFA



So C is the correct answer

15 votes

-- Praveen Saini (53k points)

5.5.49 Finite Automata: GATE2010-41 [top](#)

<http://gateoverflow.in/2342>



We need a state for counting the length. So, for length n we need n+1 states (one for length zero). We don't need a reject state for larger strings as we have NFA and not DFA. So, totally n+1 states are required. (For DFA it would be n+2).

26 votes

-- Arjun Suresh (287k points)

5.5.50 Finite Automata: GATE2011_42 [top](#)

<http://gateoverflow.in/2144>



(B) n+1

We need a state for strings of length 0, 1, 2, ... n (and their respective multiples with k). Each of these set of strings form an equivalence class as per Myhill-Nerode relation and hence needs a separate state in min-DFA.

Myhill-Nerode Class 1	Myhill-Nerode Class 2 ...	Myhill-Nerode Class n	Myhill-Nerode Class n+1
ϵ	$a, \#a = \#a = n-1, \#a = n, \#a = n+1, \#a = 2n-1, \#a = 2n, \#a = 2n+1, \dots$	$\#a = 3n-1, \dots$	$\#a = 3n, \dots$

One thing to notice here is $k > 0$. Because of this we are not able to combine Class 1 and Class n+1. Had it been $k \geq 0$, we would have had only n equivalent classes and equivalently n states in the minimal DFA.

12 votes

-- Arjun Suresh (287k points)

5.5.51 Finite Automata: GATE2011_45 [top](#)

<http://gateoverflow.in/2147>



(A) is the answer.

In (B) and (C) when the first letter of input is 'b' we reach final state, while in the given DFA first letter 'b' is not a final state. So, (B) and (C) are not accepting same language as the given DFA.

In (D) we can reach final state when the last letter is 'a', whatever be the previous transitions. But in the given DFA, when the first 2 letters are 'b' we can never reach the final state. So, (D) is also accepting a different language than the given DFA.

15 votes

-- Arjun Suresh (287k points)

5.5.52 Finite Automata: GATE2012_12 [top](#)

<http://gateoverflow.in/44>



Selected Answer

The language being accepted is a^+ . So, complement of the language is $\{\epsilon\}$.

20 votes

-- Arjun Suresh (287k points)

5.5.53 Finite Automata: GATE2012_46 [top](#)



Selected Answer

(D) is the answer. From 00 state, a '0' should take the DFA to the dead state-q. From 11, a '0' should go to 10 representing the 10 at the end of string so far. Similarly, from 00 a 1 should go to 01, from 01 a '1' should go to 11 and from 10 a '0' should go to '00'.

10 votes

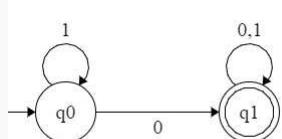
-- Arjun Suresh (287k points)

5.5.54 Finite Automata: GATE2013_33 [top](#)



Selected Answer

1. Complement of $L(A)$ (regular language) is regular , i.e also Context Free . True
2. Regular expression is $(11^*0 + 0)(0+1)^*$ True
- 3 . Minimized DFA is [both non-final states are equivalents can be minimized]



So 3 is False

4. From 3, shortest length string reached from q_0 to q_1 (final) is 0, so 4 is false

Note : a) $(0+1)^*0^*1^* = (0+1)^* + \text{something} = (0+1)^*$
 b) $(11^*0+0)(0+1)^* = (11^* + 0)(0+1)^* = 1^*0(0+1)^*$ look at Minimized DFA at 3.

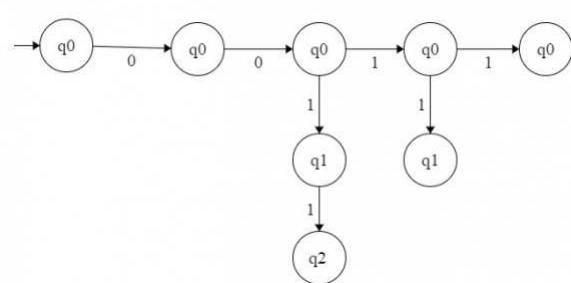
22 votes

-- Praveen Saini (53k points)

5.5.55 Finite Automata: GATE2014-1-16 [top](#)



Selected Answer



q_0, q_1 and q_2 are reachable from q_0 on input 0011

13 votes

-- Praveen Saini (53k points)

5.5.56 Finite Automata: GATE2015-1_52 [top](#)

<http://gateoverflow.in/8362>

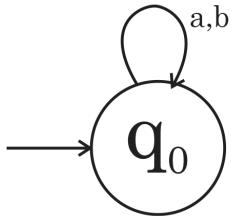


Selected Answer

$$L(M) = (a+b)^* a = \{a, aa, ba, aaa, aba, bba, \dots\}$$

$$L(N) = (a+b)^* b = \{b, ab, bb, aab, abb, bbb, \dots\}$$

So, $L(M) \cap L(N) = \{\}$. So, in the minimal DFA, we just have 1 start state with all transitions going to it self and no final state.



43 votes

-- Arjun Suresh (287k points)

5.5.57 Finite Automata: GATE2015-2_53 [top](#)

<http://gateoverflow.in/8258>



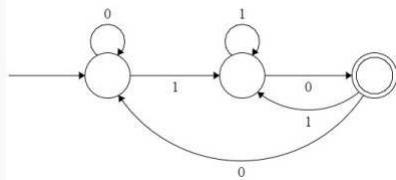
Selected Answer

All strings ending with 10. So, we need 3 states.

1. From first state on 1, we go to second state.
2. From second state on 0 we go to third state.
3. From third state on 0 we go to first state and on 1 we go to second state.

Only third state is final.

$L = (0+1)^* 10$ And following minimal DFA will be



16 votes

-- Arjun Suresh (287k points)

5.5.58 Finite Automata: GATE2015-3_18 [top](#)

<http://gateoverflow.in/8415>



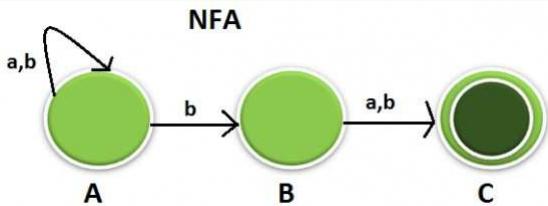
first we can draw dfa for L which has 5 states after that for L compliment we will convert all final to non final and all non final to final so total states is 5 .. option B

23 votes

-- Anoop Sonkar (4.8k points)

5.5.59 Finite Automata: GATE2017-1-22 [top](#)

<http://gateoverflow.in/118302>



NFA to DFA Conversion:

In the STT below you can see **4 states** are there but before confirming the ans **lets try to minimize**, It cant be minimized further because A goes to Nonfinal Nonfinal & AB goes to Final Final hence can not be merged. Similarly *AC goes NF NF & *ABC goes to F F hence can not be merged. **Hence 4 is the final ans.**

Delta	a	b
A	A	AB
AB	*AC	*ABC
*AC	A	AB
*ABC	*AC	*ABC

9 votes

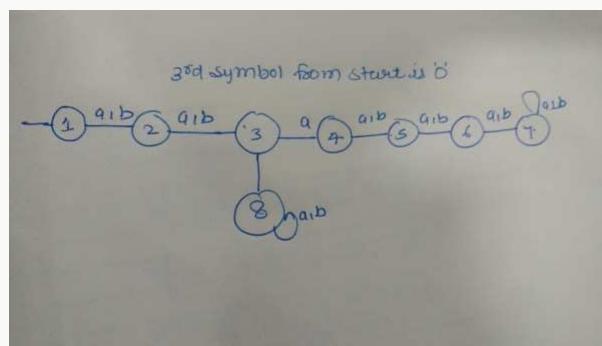
-- Ahwan Mishra (2.6k points)

5.5.60 Finite Automata: GATE2017-2-25 [top](#)

<http://gateoverflow.in/118160>



Answer is 8 states included one trap state(**8**) and final state(**7**).



10 votes

-- Prashant Singh (47.6k points)

5.5.61 Finite Automata: GATE2017-2-39 [top](#)

<http://gateoverflow.in/118384>



the transition is nothing but $((Q_2, a), b), a$

now applying Q_2, a we go to Q_0 but as lamda transition is there so it will also go to Q_2

so now $\{(Q_0, Q_2), b\} = Q_0, Q_2$

now $\{(Q_0, Q_2), a\} = Q_0, Q_1, Q_2$

so C should be correct answer here

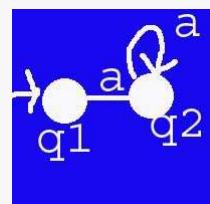
11 votes

-- **Aboveallplayer** (18.5k points)

5.5.62 Finite Automata: ISI2011-CS-4b [top](#)

<http://gateoverflow.in/48157>

consider the dfa whose language "strings which contains atleast one a" over $\Sigma = \{a\}$



here q_2 is final state

consider a string $(q_2, a^8) = (q_2, a^{12}) = q_2$

now $(q_2, a^{gcd(8,12)}) = (q_2, a^4) = q_2$

as gcd of 8,12 is 4...

and a^4 also goes to q_2 hence proved

2 votes

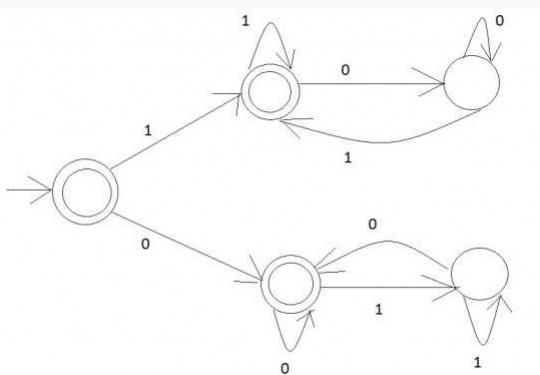
-- **Tauhin Gangwar** (9k points)

5.5.63 Finite Automata: ISI2014-CS-4a [top](#)

<http://gateoverflow.in/47443>



The DFA for that language w is



6 votes

-- **srestha** (55.2k points)

5.6

Grammar(1) [top](#)

5.6.1 Grammar: GATE2008-51 [top](#)

<http://gateoverflow.in/463>

Match the following:

E. Checking that identifiers are declared before their use	P. $L = \{a^n b^m c^n d^m \mid n \geq 1, m \geq 1\}$
F. Number of formal parameters in the declaration of a function agrees with the number of actual parameters in a use of that function	Q. $X \rightarrow XbX \mid XcX \mid dXf \mid g$
G. Arithmetic expressions with matched pairs of parentheses	R. $L = \{wcw \mid w \in (a \mid b)^*\}$
H. Palindromes	S. $X \rightarrow bXb \mid cXc \mid \epsilon$

- A. E-P, F-Q, G-S, H-R
- B. E-R, F-P, G-S, H-Q
- C. E-R, F-P, G-Q, H-S
- D. E-P, F-R, G-S, H-Q

[gate2008](#) [normal](#) [theory-of-computation](#) [grammar](#)

[Answer](#)

Answers: Grammar

5.6.1 Grammar: GATE2008-51 [top](#)

<http://gateoverflow.in/463>



Selected Answer

$H - S$ is true coz strings generated by this grammar satisfies the definition of an even length palindrome string. this rules out B and D options.

$G - Q$ is confirmed as both options A and C has it as true.

$E - R$ is true coz R is the only grammar that checks: what(w) has occurred earlier is present afterwards This equals the definition of E

Hence, **option C** is true.

14 votes

-- Amar Vashishth (27.9k points)

5.7

Identify Class Language(30) [top](#)

5.7.1 Identify Class Language: CMI2010-A-01 [top](#)

<http://gateoverflow.in/46131>

Over the alphabet $\{0,1\}$, consider the language

$$L = \{w \mid w \text{ does not contain the substring } 0011\}$$

Which of the following is true about L.

- A. L is not context free
- B. L is regular
- C. L is not regular but it is context free
- D. L is context free but not recursively enumerable

[cmi2010](#) [theory-of-computation](#) [identify-class-language](#)
[Answer](#)

5.7.2 Identify Class Language: GATE1988-2ix [top](#)

<http://gateoverflow.in/93949>

What is the type of the language L , where $L = \{a^n b^n \mid 0 < n < 327\text{-th prime number}\}$

[gate1988](#) [normal](#) [descriptive](#) [algorithms](#) [theory-of-computation](#) [identify-class-language](#)
[Answer](#)

5.7.3 Identify Class Language: GATE1991-17,a [top](#)

<http://gateoverflow.in/26653>

Show that the Turing machines, which have a read only input tape and constant size work tape, recognize precisely the class of regular languages.

[gate1991](#) [theory-of-computation](#) [descriptive](#) [identify-class-language](#)
[Answer](#)

5.7.4 Identify Class Language: GATE1994_19 [top](#)

<http://gateoverflow.in/2515>

(a) Given a set

$$S = \{x \mid \text{there is an } x\text{-block of 5's in the decimal expansion of } \pi\}$$

(Note: x -block is a maximal block of x successive 5's)

Which of the following statements is true with respect to S ? No reason to be given for the answer.

- i. S is regular
- ii. S is recursively enumerable
- iii. S is not recursively enumerable
- iv. S is recursive

(b) Given that a language L_1 is regular and that the language $L_1 \cup L_2$ is regular, is the language L_2 always regular? Prove your answer.

[gate1994](#) [theory-of-computation](#) [identify-class-language](#) [normal](#)
[Answer](#)

5.7.5 Identify Class Language: GATE1999_2.4 [top](#)

<http://gateoverflow.in/1482>

Multiple choices may be correct:

If L_1 is context free language and L_2 is a regular language which of the following is/are false?

- A. $L_1 - L_2$ is not context free
- B. $L_1 \cap L_2$ is context free
- C. $\sim L_1$ is context free
- D. $\sim L_2$ is regular

[gate1999](#) [theory-of-computation](#) [identify-class-language](#) [normal](#)
[Answer](#)

5.7.6 Identify Class Language: GATE2000-1.5 [top](#)

<http://gateoverflow.in/628>

Let L denote the languages generated by the grammar $S \rightarrow 0S0 \mid 00$.

Which of the following is TRUE?

- A. $L = 0^+$
- B. L is regular but not 0^+

- C. L is context free but not regular
 D. L is not context free

gate2000 theory-of-computation easy identify-class-language

[Answer](#)

5.7.7 Identify Class Language: GATE2002-1.7 [top](#)

<http://gateoverflow.in/811>

The language accepted by a Pushdown Automaton in which the stack is limited to 10 items is best described as

- A. Context free
 B. Regular
 C. Deterministic Context free
 D. Recursive

gate2002 theory-of-computation easy identify-class-language

[Answer](#)

5.7.8 Identify Class Language: GATE2004-87 [top](#)

<http://gateoverflow.in/1081>

The language $\{a^m b^n c^{m+n} \mid m, n \geq 1\}$ is

- A. regular
 B. context-free but not regular
 C. context-sensitive but not context free
 D. type-0 but not context sensitive

gate2004 theory-of-computation normal identify-class-language

[Answer](#)

5.7.9 Identify Class Language: GATE2005-55 [top](#)

<http://gateoverflow.in/1378>

Consider the languages:

$$L_1 = \{a^n b^n c^m \mid n, m > 0\} \text{ and } L_2 = \{a^n b^m c^m \mid n, m > 0\}$$

Which one of the following statements is FALSE?

- A. $L_1 \cap L_2$ is a context-free language
 B. $L_1 \cup L_2$ is a context-free language
 C. L_1 and L_2 are context-free languages
 D. $L_1 \cap L_2$ is a context sensitive language

gate2005 theory-of-computation identify-class-language normal

[Answer](#)

5.7.10 Identify Class Language: GATE2005-IT-4 [top](#)

<http://gateoverflow.in/3748>

Let L be a regular language and M be a context-free language, both over the alphabet Σ . Let L^c and M^c denote the complements of L and M respectively. Which of the following statements about the language $L^c \cup M^c$ is TRUE?

- A. It is necessarily regular but not necessarily context-free.
 B. It is necessarily context-free.
 C. It is necessarily non-regular.
 D. None of the above

gate2005-it theory-of-computation normal identify-class-language

[Answer](#)

5.7.11 Identify Class Language: GATE2005-IT-6 [top](#)

<http://gateoverflow.in/3751>

The language $\{0^n 1^n 2^n \mid 1 \leq n \leq 10^6\}$ is

- A. regular
- B. context-free but not regular
- C. context-free but its complement is not context-free
- D. not context-free

[gate2005-it](#) [theory-of-computation](#) [easy](#) [identify-class-language](#)

Answer

5.7.12 Identify Class Language: GATE2006-30 [top](#)

<http://gateoverflow.in/993>

For $s \in (0+1)^*$ let $d(s)$ denote the decimal value of s (e.g. $d(101) = 5$). Let

$$L = \{s \in (0+1)^* \mid d(s) \bmod 5 = 2 \text{ and } d(s) \bmod 7 \neq 4\}$$

Which one of the following statements is true?

- A. L is recursively enumerable, but not recursive
- B. L is recursive, but not context-free
- C. L is context-free, but not regular
- D. L is regular

[gate2006](#) [theory-of-computation](#) [normal](#) [identify-class-language](#)

Answer

5.7.13 Identify Class Language: GATE2006-33 [top](#)

<http://gateoverflow.in/996>

Let L_1 be a regular language, L_2 be a deterministic context-free language and L_3 a recursively enumerable, but not recursive, language. Which one of the following statements is false?

- A. $L_1 \cap L_2$ is a deterministic CFL
- B. $L_3 \cap L_1$ is recursive
- C. $L_1 \cup L_2$ is context free
- D. $L_1 \cap L_2 \cap L_3$ is recursively enumerable

[gate2006](#) [theory-of-computation](#) [normal](#) [identify-class-language](#)

Answer

5.7.14 Identify Class Language: GATE2007-30 [top](#)

<http://gateoverflow.in/1228>

The language $L = \{0^i 2 1^i \mid i \geq 0\}$ over the alphabet $\{0, 1, 2\}$ is:

- A. not recursive
- B. is recursive and is a deterministic CFL
- C. is a regular language
- D. is not a deterministic CFL but a CFL

[gate2007](#) [theory-of-computation](#) [normal](#) [identify-class-language](#)

Answer

5.7.15 Identify Class Language: GATE2008-9 [top](#)

<http://gateoverflow.in/407>

Which of the following is true for the language

$$\{a^p \mid p \text{ is a prime}\}?$$

- A. It is not accepted by a Turing Machine
 B. It is regular but not context-free
 C. It is context-free but not regular
 D. It is neither regular nor context-free, but accepted by a Turing machine

[gate2008](#) [theory-of-computation](#) [easy](#) [identify-class-language](#)

[Answer](#)

5.7.16 Identify Class Language: GATE2008-IT-33 [top](#)

<http://gateoverflow.in/3343>

Consider the following languages.

- $L_1 = \{a^i b^j c^k \mid i = j, k \geq 1\}$
- $L_2 = \{a^i b^j \mid j = 2i, i \geq 0\}$

Which of the following is true?

- A. L_1 is not a CFL but L_2 is
 B. $L_1 \cap L_2 = \emptyset$ and L_1 is non-regular
 C. $L_1 \cup L_2$ is not a CFL but L_2 is
 D. There is a 4-state PDA that accepts L_1 , but there is no DPDA that accepts L_2

[gate2008-it](#) [theory-of-computation](#) [normal](#) [identify-class-language](#)

[Answer](#)

5.7.17 Identify Class Language: GATE2009-40 [top](#)

<http://gateoverflow.in/1326>

Let $L = L_1 \cap L_2$, where L_1 and L_2 are languages as defined below:

$$L_1 = \{a^m b^m c a^n b^n \mid m, n \geq 0\}$$

$$L_2 = \{a^i b^j c^k \mid i, j, k \geq 0\}$$

Then L is

- A. Not recursive
 B. Regular
 C. Context free but not regular
 D. Recursively enumerable but not context free.

[gate2009](#) [theory-of-computation](#) [easy](#) [identify-class-language](#)

[Answer](#)

5.7.18 Identify Class Language: GATE2010-40 [top](#)

<http://gateoverflow.in/2341>

Consider the languages

$$L_1 = \{0^i 1^j \mid i \neq j\},$$

$$L_2 = \{0^i 1^j \mid i = j\},$$

$$L_3 = \{0^i 1^j \mid i = 2j + 1\},$$

$$L_4 = \{0^i 1^j \mid i \neq 2j\}$$

- A. Only L_2 is context free.
 B. Only L_2 and L_3 are context free.
 C. Only L_1 and L_2 are context free.
 D. All are context free

[gate2010](#) [theory-of-computation](#) [context-free-language](#) [identify-class-language](#) [normal](#)

[Answer](#)

5.7.19 Identify Class Language: GATE2011_26 [top](#)

<http://gateoverflow.in/2128>

Consider the languages L_1 , L_2 and L_3 as given below.

$$L_1 = \{0^p 1^q \mid p, q \in N\}, L_2 = \{0^p 1^q \mid p, q \in N \text{ and } p = q\} \text{ and } L_3 = \{0^p 1^q 0^r \mid p, q, r \in N \text{ and } p = q = r\}.$$

Which of the following statements is **NOT TRUE**?

- (A) Push Down Automata (PDA) can be used to recognize L_1 and L_2
- (B) L_1 is a regular language
- (C) All the three languages are context free
- (D) Turing machines can be used to recognize all the languages

[gate2011](#) [theory-of-computation](#) [identify-class-language](#) [normal](#)

[Answer](#)

5.7.20 Identify Class Language: GATE2013_32 [top](#)

<http://gateoverflow.in/1543>

Consider the following languages.

$$L_1 = \{0^p 1^q 0^r \mid p, q, r \geq 0\}$$

$$L_2 = \{0^p 1^q 0^r \mid p, q, r \geq 0, p \neq r\}$$

Which one of the following statements is **FALSE**?

- (A) L_2 is context-free.
- (B) $L_1 \cap L_2$ is context-free.
- (C) Complement of L_2 is recursive.
- (D) Complement of L_1 is context-free but not regular.

[gate2013](#) [theory-of-computation](#) [identify-class-language](#) [normal](#)

[Answer](#)

5.7.21 Identify Class Language: GATE2014-3-36 [top](#)

<http://gateoverflow.in/2070>

Consider the following languages over the alphabet $\Sigma = \{0, 1, c\}$

$$L_1 = \{0^n 1^n \mid n \geq 0\}$$

$$L_2 = \{wcw^r \mid w \in \{0, 1\}^*\}$$

$$L_3 = \{ww^r \mid w \in \{0, 1\}^*\}$$

Here, w^r is the reverse of the string w . Which of these languages are deterministic Context-free languages?

- A. None of the languages
- B. Only L_1
- C. Only L_1 and L_2
- D. All the three languages

[gate2014-3](#) [theory-of-computation](#) [identify-class-language](#) [context-free-language](#) [normal](#)

[Answer](#)

5.7.22 Identify Class Language: GATE2017-1-37 [top](#)

<http://gateoverflow.in/11832>

Consider the context-free grammars over the alphabet $\{a, b, c\}$ given below. S and T are non-terminals.

$$G_1 : S \rightarrow aSb \mid T, T \rightarrow cT \in$$

$$G_2 : S \rightarrow bSa \mid T, T \rightarrow cT \in$$

The language $L(G_1) \cap L(G_2)$ is

- (A) Finite.

- (B) Not finite but regular.
 (C) Context-Free but not regular.
 (D) Recursive but not context-free.

gate2017-1 theory-of-computation context-free-language identify-class-language normal

[Answer](#)

5.7.23 Identify Class Language: GATE2017-2-40 [top](#)

<http://gateoverflow.in/118615>

Consider the following languages.

- $L_1 = \{a^p \mid p \text{ is a prime number}\}$
- $L_2 = \{a^n b^m c^{2m} \mid n \geq 0, m \geq 0\}$
- $L_3 = \{a^n b^n c^{2n} \mid n \geq 0\}$
- $L_4 = \{a^n b^n \mid n \geq 1\}$

Which of the following are CORRECT?

- I. L_1 is context free but not regular
 - II. L_2 is not context free
 - III. L_3 is not context free but recursive
 - IV. L_4 is deterministic context free
- A. I, II and IV only
 B. II and III only
 C. I and IV only
 D. III and IV only

gate2017-2 theory-of-computation identify-class-language

[Answer](#)

5.7.24 Identify Class Language: TIFR2010-B-22 [top](#)

<http://gateoverflow.in/18622>

Let L consist of all binary strings beginning with a 1 such that its value when converted to decimal is divisible by 5. Which of the following is true?

- a. L can be recognized by a deterministic finite state automaton.
- b. L can be recognized by a non-deterministic finite state automaton but not by a deterministic finite state automaton.
- c. L can be recognized by a deterministic push-down automaton but not by a non-deterministic finite state automaton.
- d. L can be recognized by a non-deterministic push-down automaton but not by a deterministic push-down automaton.
- e. L cannot be recognized by any push-down automaton.

tifr2010 theory-of-computation identify-class-language

[Answer](#)

5.7.25 Identify Class Language: TIFR2010-B-35 [top](#)

<http://gateoverflow.in/19247>

Consider the following languages over the alphabet $\{0,1\}$.

$$\begin{aligned} L1 &= \{x.x^R \mid x \in \{0,1\}^*\} \\ L2 &= \{x.x \mid x \in \{0,1\}^*\} \end{aligned}$$

Where x^R is the reverse of string x ; e.g. $011^R = 110$. Which of the following is true?

- a. Both $L1$ and $L2$ are regular.
- b. $L1$ is context-free but not regular whereas $L2$ is regular.
- c. Both $L1$ and $L2$ are context free and neither is regular.
- d. $L1$ is context free but $L2$ is not context free.
- e. Both $L1$ and $L2$ are not context free.

tifr2010 theory-of-computation identify-class-language

Answer**5.7.26 Identify Class Language: TIFR2012-B-18** [top](#)<http://gateoverflow.in/25216>

Let a^i denote a sequence $a.a\dots a$ with i letters and let \mathbb{N} be the set of natural numbers $1, 2, \dots$. Let $L_1 = \{a^i b^{2i} \mid i \in \mathbb{N}\}$ and $L_2 = \{a^i b^{i^2} \mid i \in \mathbb{N}\}$ be two languages. Which of the following is correct?

- a. Both L_1 and L_2 are context-free languages.
- b. L_1 is context-free and L_2 is recursive but not context-free.
- c. Both L_1 and L_2 are recursive but not context-free.
- d. L_1 is regular and L_2 is context-free.
- e. Complement of L_2 is context-free.

[tifr2012](#) | [theory-of-computation](#) | [identify-class-language](#)**Answer****5.7.27 Identify Class Language: TIFR2014-B-13** [top](#)<http://gateoverflow.in/27320>

Let L be a given context-free language over the alphabet $\{a, b\}$. Construct L_1, L_2 as follows. Let $L_1 = L - \{xyx \mid x, y \in \{a, b\}^*\}$, and $L_2 = L \cdot L$. Then,

- a. Both L_1 and L_2 are regular.
- b. Both L_1 and L_2 are context free but not necessarily regular.
- c. L_1 is regular and L_2 is context free.
- d. L_1 and L_2 both may not be context free.
- e. L_1 is regular but L_2 may not be context free.

[tifr2014](#) | [theory-of-computation](#) | [identify-class-language](#)**Answer****5.7.28 Identify Class Language: TIFR2015-B-8** [top](#)<http://gateoverflow.in/29865>

Let $\sum_1 = \{a\}$ be a one letter alphabet and $\sum_2 = \{a, b\}$ be a two letter alphabet. A language over an alphabet is a set of finite length words comprising letters of the alphabet. Let L_1 and L_2 be the set of languages over \sum_1 and \sum_2 respectively. Which of the following is true about L_1 and L_2 :

- a. Both are finite.
- b. Both are countably infinite.
- c. L_1 is countable but L_2 is not.
- d. L_2 is countable but L_1 is not.
- e. Neither of them is countable.

[tifr2015](#) | [identify-class-language](#)**Answer****5.7.29 Identify Class Language: TIFR2017-B-14** [top](#)<http://gateoverflow.in/95825>

Consider the following grammar G with terminals $\{[,]\}$, start symbol S , and non-terminals $\{A, B, C\}$:

$$\begin{aligned} S &\rightarrow AC \mid SS \mid AB \\ C &\rightarrow SB \\ A &\rightarrow [\\ B &\rightarrow] \end{aligned}$$

A language L is called prefix-closed if for every $x \in L$, every prefix of x is also in L . Which of the following is FALSE?

- A. $L(G)$ is context free
- B. $L(G)$ is infinite
- C. $L(G)$ can be recognized by a deterministic push down automaton
- D. $L(G)$ is prefix-closed
- E. $L(G)$ is recursive

[tifr2017](#) | [theory-of-computation](#) | [identify-class-language](#)
Answer

5.7.30 Identify Class Language: TIFR2017-B-4 [top](#)

<http://gateoverflow.in/95680>

Let L be the language over the alphabet $\{1, 2, 3, (,)\}$ generated by the following grammar (with start symbol S , and non-terminals $\{A, B, C\}$):

$$S \rightarrow A B C A \rightarrow (B \rightarrow 1 B \mid 2 B \mid 3 B B \rightarrow 1 \mid 2 \mid 3 C \rightarrow)$$

Then, which of the following is TRUE?

- A. L is finite
- B. L is not recursively enumerable
- C. L is regular
- D. L contains only strings of even length
- E. L is context-free but not regular

[tifr2017](#) | [theory-of-computation](#) | [identify-class-language](#)
Answer

Answers: Identify Class Language

5.7.1 Identify Class Language: CMI2010-A-01 [top](#)

<http://gateoverflow.in/46131>

Selected Answer

L is regular.

1 votes

-- Praveen Saini (53k points)

5.7.2 Identify Class Language: GATE1988-2ix [top](#)

<http://gateoverflow.in/93949>

Selected Answer

Here n is finite.. and finite language must be regular..

if n is not restricted, then it would be DCFL...

3 votes

-- kirti singh (3.3k points)

5.7.3 Identify Class Language: GATE1991-17,a [top](#)

<http://gateoverflow.in/26653>

https://en.wikipedia.org/wiki/Read-only_Turing_machine

3 votes

-- srestha (55.2k points)

5.7.4 Identify Class Language: GATE1994_19 [top](#)

<http://gateoverflow.in/2515>

(b) No. need not be. Take $L_2 = \{a^n b^n \mid n > 0\}$ and $L_1 = \text{all strings over } \{a, b\}$. Now, $L_1 \cup L_2$ is L_1 only and is regular but L_2 is not regular.

7 votes

-- Arjun Suresh (287k points)

5.7.5 Identify Class Language: GATE1999_2.4 [top](#)

<http://gateoverflow.in/1482>

Selected Answer

L_2 is regular, so complement of L_2 , $(\sim L_2)$, is also regular

Regular languages under complement. So **D is True**

$L_1 \cap L_2$ is context free.

Intersection of Context free language with Regular language is Context free. So **b is True**

$L_1 - L_2 = L_1 \cap (\sim L_2)$ is context free

Intersection of Context free language with Regular language is Context free. So **a is False**

$\sim L_1$ is not context free

Context free languages are not closed under complement. So **c is False (May/not be)**.

13 votes

-- Praveen Saini (5.3k points)

5.7.6 Identify Class Language: GATE2000-1.5 [top](#)

<http://gateoverflow.in/628>



Selected Answer

B. is the answer for this question

Language generated by this grammar is $L = (00)^{\infty}$ that is regular but not 0^{∞}

14 votes

-- Manu Thakur (5.8k points)

5.7.7 Identify Class Language: GATE2002-1.7 [top](#)

<http://gateoverflow.in/811>



Selected Answer

B. Regular.

With only finite positions in stack, we can have only finite configurations and these can also be modeled as states in a finite automata.

10 votes

-- Arjun Suresh (287k points)

5.7.8 Identify Class Language: GATE2004-87 [top](#)

<http://gateoverflow.in/1081>



Selected Answer

Language is not regular bcoz we need to match count of c's is equal to count of b's + count of a's

and that can implement by PDA

$\delta(q_0, a, \epsilon) = (q_0, a)$ [push a in stack, as per language a comes first]

$\delta(q_0, a, a) = (q_0, aa)$ [push all a's into stack]

$\delta(q_0, b, a) = (q_1, ba)$ [push b in stack, state change to q1 that sure b comes after a]

$\delta(q_1, b, b) = (q_1, bb)$ [push all b's in stack]

$\delta(q_1, c, b) = (q_2, \epsilon)$ [pop one b for one c]

$\delta(q_2, c, b) = (q_2, c)$ [pop one b's for each c and continue same]

$\delta(q_2, c, a) = (q_3, \epsilon)$ [pop one a for one c , when there is no more b in stack]

$\delta(q_3, c, a) = (q_3, \epsilon)$ [pop one a for each c and continue same]

$\delta(q_3, \epsilon, \epsilon) = (q_f, \epsilon)$ [if sum of c's is sum of a's and b's then stack will be empty when there is no c in input]

answer = **option B** : language is context-free but not regular.

Note : 1. state changes make sure b's comes after a and c's comes after b's]

7 votes

-- Praveen Saini (53k points)

5.7.9 Identify Class Language: GATE2005-55 [top](#)

<http://gateoverflow.in/1378>



Selected Answer

L1 is CFL [put a's in stack , and pop a with each b]]

L2 is CFL [put b's in stack and pop b with each c]

c) is True.

b) is True CFL is closed under Union [$S \rightarrow S_1 | S_2$ where S_1 is grammar for L1 and S_2 for L2]

CFL is not closed under Intersection, so intersection of two CFLs may or may not be CFL. Lets examine:

$L_1 \cap L_2 = \{ a^i b^i c^i, i > 0 \}$ which is Context sensitive but not context free [can't match a's,b's and c's with one stack]

So a) is False

d) is True

13 votes

-- Praveen Saini (53k points)

5.7.10 Identify Class Language: GATE2005-IT-4 [top](#)

<http://gateoverflow.in/3748>



Selected Answer

Take $L = \Sigma^*$ then $L^c = \emptyset$ and $M^c \cup \emptyset = M^c$.

We know that complement of CFL need not be a CFL as CFL is not closed under complement.

So, (A) and (B) are false.

If we take $L = \emptyset$ then $L^c = \Sigma^*$ and $M^c \cup \Sigma^* = \Sigma^*$ which is regular - (C) is also false.

So, answer (D)

20 votes

-- Arjun Suresh (287k points)

5.7.11 Identify Class Language: GATE2005-IT-6 [top](#)

<http://gateoverflow.in/3751>



Selected Answer

Regular (in fact finite). Since n is finite, we have a finite set of strings satisfying the given conditions. So, we can easily make a finite automata for those strings.

11 votes

-- Arjun Suresh (287k points)

5.7.12 Identify Class Language: GATE2006-30 [top](#)

<http://gateoverflow.in/993>



Selected Answer

Refer this

http://gateoverflow.in/1695/gate1998_4

$L_1 = \{ s \in (0+1)^* \mid d(s) \bmod 5 = 2 \}$ is regular

having 2 as final state out of {0,1,2,3,4} states

as given in example in posted link [in same DFA , final state will be 2 instead of 0]

similarly $L_2 = \{ s \in (0+1)^* \mid d(s) \bmod 7 \neq 4 \}$ is also regular

having states {0,1,2,3,4,5,6} and all are final state except 4

$L_1 \cap L_2$ is L (given problem) is also regular

As regular languages are closed under intersection. D is correct option.

16 votes

-- Praveen Saini (53k points)

5.7.13 Identify Class Language: GATE2006-33 [top](#)



Selected Answer

A) True : DCFL are closed under Intersection with Regular Languages

C) True : L_1 is regular hence also CFL and every DCFL is also CFL and All CFL are closed under Union.

D) True : L_1 is regular hence also RE; L_2 is DCFL hence also RE; RE languages are closed under Intersection

B) False : L_1 is recursive hence also decidable, L_3 is RE but not Recursive hence it is undecidable. Intersection of Recursive language and Recursive Enumerable language is Recursive Enumerable language.

16 votes

-- Danish (3.5k points)

5.7.14 Identify Class Language: GATE2007-30 [top](#)



Selected Answer

$L = \{0^i 2 1^i \mid i \geq 0\}$ has only one comparison that can be done using a DPDA. Hence, its DCFL.

Context free languages are a proper subset of Recursive Languages. \therefore it is recursive too.

answer = **option B**

7 votes

-- Amar Vashishth (27.9k points)

5.7.15 Identify Class Language: GATE2008-9 [top](#)



Selected Answer

We have [algorithms to generate prime numbers](#) \implies we can generate sequence of p for the given language, hence strings as defined by the language definition.

So by Church Turing Thesis we can say that there exists a Turing Machine which can accept the given language.

answer = **option D**

10 votes

-- Amar Vashishth (27.9k points)

5.7.16 Identify Class Language: GATE2008-IT-33 [top](#)



Selected Answer

Both languages can be accepted by a DPDA :

L_1 = start pushing element X into the stack on input 'a' ... start popping X on input 'b' ... move to final state when stack

empty and input = 'c'

L_2 = start pushing elements XX into the stack on input 'a' ... start popping X on input 'b' ... move to final state when stack empty and input = 'epsilon'

so (A) and (D) are False

$L_1 \cup L_2$ is a CFL ... we can build it by having L_1 , L_2 and an extra state ... and an 'epsilon' transition to both L_1 and L_2 from that extra state.

so (C) is false

$L_1 \cap L_2 = \Phi$ because we have no string $a^i b^j$ where $i=j$ and $i=2j$ for $i,j \geq 1$

and clearly L_1 is not a regular language

so (B) is true.

16 votes

-- Danish (3.5k points)

5.7.17 Identify Class Language: GATE2009-40 [top](#)

<http://gateoverflow.in/1326>



Selected Answer

$L_1 \cap L_2 = \{a^m b^m c \mid m \geq 0\}$, which is context free but not regular.

Option C.

21 votes

-- Arjun Suresh (287k points)

5.7.18 Identify Class Language: GATE2010-40 [top](#)

<http://gateoverflow.in/2341>



Selected Answer

All are context free.

$L_1 \rightarrow$ Push 0 on stack and when 1 comes, start popping. If stack becomes empty and 1's are remaining start pushing 1. At end of string accept if stack is non- empty.

$L_2 \rightarrow$ Do the same as for L_1 , but accept if stack is empty at end of string.

$L_3 \rightarrow$ Do, the same as for L_2 , but for each 0, push two 0's on stack and start the stack with a 0.

$L_4 \rightarrow$ Do the same as for L_1 , but for each 0, push two 0's on stack

All are in fact DCFL. Pushing two 0's on stack might sound non-trivial but we can do this by pushing one symbol and going to a new state. Then on this new state on empty symbol, push one more symbol on stack and come back.

14 votes

-- Arjun Suresh (287k points)

5.7.19 Identify Class Language: GATE2011_26 [top](#)

<http://gateoverflow.in/2128>



Selected Answer

Answer is C.

L_1 is RL

L_2 is CFL

L_3 is CSL

Turning Machine is powerful Machine it can be used to accept all the languages (RL,CFL,CSL,RE)

14 votes

-- Sona Praneeth Akula (4k points)

5.7.20 Identify Class Language: GATE2013_32 [top](#)



Selected Answer

L1 is regular and hence context-free also. Regular expression for L1 is $0^*1^*0^*$. So, (D) is the false choice.

L2 is context-free but not regular. We need a stack to count if the number of 0's before and after the 1 (1 may be absent also) are not same. So, $L1 \cap L2$ is context-free as regular \cap context-free is context-free. \rightarrow (B) is true.

Complement of L2 is recursive as context-free complement is always recursive (actually even context-sensitive).

14 votes

-- Arjun Suresh (287k points)

5.7.21 Identify Class Language: GATE2014-3-36 [top](#)



Selected Answer

C.

L_3 is CFL and not DCFL as in no way we can deterministically determine the MIDDLE point of the input string.

13 votes

-- Gate Keeda (18.8k points)

5.7.22 Identify Class Language: GATE2017-1-37 [top](#)



Selected Answer

Since while intersection all strings produced by production aSb in G1 and bSa in G2 will be 0

so only common production will be

$S \rightarrow T$

$T \rightarrow cT \mid \epsilon$

which is nothing but c^* hence it is REGULAR and INFINITE

so OPTION (B)

4 votes

-- sriv_shubham (2.4k points)

5.7.23 Identify Class Language: GATE2017-2-40 [top](#)



Selected Answer

L_1 is Csl, L_2 is context free

L_3 is not Context free and L_4 is Dcfl

So Option D

10 votes

-- Swapnil (2k points)

5.7.24 Identify Class Language: TIFR2010-B-22 [top](#)



Selected Answer

I can be recognized by a dfa. we have a dfa to accept all such strings which when interpreted as decimal number are divisible by n. where n can be anything the dfa of such can be made by a trick.

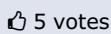
states are equal to possible remainders

	0	1
q0	q0	q1
q1	q2	q3
q2	q4	q0
q3	q1	q2
q4	q3	q4

if u can see the symmetry in it. write states and make fill like q0 q1 q2 q3 ...

now it is saying that it has to always start with 1 which the above dfa will not satisfy so make it a nfa by making a transition from q0 on zero to a new dead state. now you have a nfa reduce it which will result in a deterministic dfa .

so option a



-- No Need (13.7k points)

5.7.25 Identify Class Language: TIFR2010-B-35 [top](#)

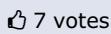
<http://gateoverflow.in/19247>



Selected Answer

L1={x.x^R|xε{0,1}*} Its a even palindrome so its CFL
L2={x.x|xε{0,1}*} Its a string matching so its a CSL

Option D) L1 is context free but L2 is not context free.



-- Umang Raman (14.5k points)

5.7.26 Identify Class Language: TIFR2012-B-18 [top](#)

<http://gateoverflow.in/25216>



Selected Answer

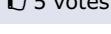
L1 - CFL, S-> aSbb|abb

L2 - Not CFL ,we can't count i^2 using CFL.

- a) False because L2 is not CFL.
- B) True. L2 is recursive
- C) False because L1 is CFL.
- D) False, L1 not regular.

E) False, as complement of L2 is also not context free. It still need to computer i^2 for checking for inequality.

Answer :- B



-- Akash (41.5k points)

5.7.27 Identify Class Language: TIFR2014-B-13 [top](#)

<http://gateoverflow.in/27320>



Selected Answer

L is a context free language over {a,b}
 $L_1 = L - \{xyx \mid x,y \in \{a,b\}^*\}$
 $= L - \{ \text{ all strings over } \{a,b\} \}$ [Note: all strings can be generated from y by putting $x = \epsilon$]
 $= L - (a+b)^* = \{\}, \text{ empty set.}$ [Note : $L_1 - L_2 = \{ \text{ string in } L_1 \text{ but not in } L_2 \}$]

So , L_1 is a Regular Language .

L is a context free language over {a,b}

$L_2 = L \cdot L$

Context free languages are closed under Concatenation.

So L_2 is Context Free Language.

Option C is correct.

16 votes

-- Praveen Saini (5.3k points)

5.7.28 Identify Class Language: TIFR2015-B-8 [top](#)



Selected Answer

languages over alphabet set are uncountable so ans should be e

3 votes

-- Pooja Palod (31.2k points)

5.7.29 Identify Class Language: TIFR2017-B-14 [top](#)



Selected Answer

The given grammar generates balanced parenthesis.

Lets take a smallest string : [[]] (say x)

Prefixes of x are : [, [[, [[]]

BUT they don't belong to the langauge generated by the given grammar.

SO, the answer will be **Option D**

Correct me if am wrong

5 votes

-- sarveswara rao vangala (1.4k points)

5.7.30 Identify Class Language: TIFR2017-B-4 [top](#)



Selected Answer

language accepted by that grammar is
 $((1+2+3)^+)$
hence regular!

3 votes

-- Motamarri Anusha (11.4k points)

5.8

Non Determinism(7) [top](#)

5.8.1 Non Determinism: GATE1992-02,xx [top](#)

<http://gateoverflow.in/577>

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

In which of the cases stated below is the following statement true?

"For every non-deterministic machine M_1 there exists an equivalent deterministic machine M_2 recognizing the same language".

- (a). M_1 is non-deterministic finite automaton.
- (b). M_1 is non-deterministic PDA.
- (c). M_1 is a non-deterministic Turing machine.
- (d). For no machines M_1 and M_2 , the above statement true.

[gate1992](#) [theory-of-computation](#) [easy](#) [non-determinism](#)

[Answer](#)

5.8.2 Non Determinism: GATE1994-1.16 [top](#)

<http://gateoverflow.in/2459>

Which of the following conversions is not possible (algorithmically)?

- A. Regular grammar to context free grammar
- B. Non-deterministic FSA to deterministic FSA
- C. Non-deterministic PDA to deterministic PDA
- D. Non-deterministic Turing machine to deterministic Turing machine

[gate1994](#) [theory-of-computation](#) [easy](#) [non-determinism](#)

[Answer](#)

5.8.3 Non Determinism: GATE1998-1.11 [top](#)

<http://gateoverflow.in/1648>

Regarding the power of recognition of languages, which of the following statements is false?

- A. The non-deterministic finite-state automata are equivalent to deterministic finite-state automata.
- B. Non-deterministic Push-down automata are equivalent to deterministic Push-down automata.
- C. Non-deterministic Turing machines are equivalent to deterministic Push-down automata.
- D. Non-deterministic Turing machines are equivalent to deterministic Turing machines.
- E. Multi-tape Turing machines are available are equivalent to Single-tape Turing machines.

[gate1998](#) [theory-of-computation](#) [easy](#) [non-determinism](#)

[Answer](#)

5.8.4 Non Determinism: GATE2004-IT-9 [top](#)

<http://gateoverflow.in/3650>

Which one of the following statements is FALSE?

- A. There exist context-free languages such that all the context-free grammars generating them are ambiguous
- B. An unambiguous context-free grammar always has a unique parse tree for each string of the language generated by it
- C. Both deterministic and non-deterministic pushdown automata always accept the same set of languages
- D. A finite set of strings from some alphabet is always a regular language

[gate2004-it](#) [theory-of-computation](#) [easy](#) [non-determinism](#)

[Answer](#)

5.8.5 Non Determinism: GATE2005-54 [top](#)

<http://gateoverflow.in/1377>

Let N_f and N_p denote the classes of languages accepted by non-deterministic finite automata and non-deterministic push-down automata, respectively. Let D_f and D_p denote the classes of languages accepted by deterministic finite automata and deterministic push-down automata respectively. Which one of the following is TRUE?

- A. $D_f \subset N_f$ and $D_p \subset N_p$
- B. $D_f \subset N_f$ and $D_p = N_p$
- C. $D_f = N_f$ and $D_p = N_p$
- D. $D_f = N_f$ and $D_p \subset N_p$

gate2005 theory-of-computation easy non-determinism

[Answer](#)

5.8.6 Non Determinism: GATE2009-16, ISRO2017-12 [top](#)

<http://gateoverflow.in/1308>

Which one of the following is FALSE?

- A. There is a unique minimal DFA for every regular language
- B. Every NFA can be converted to an equivalent PDA.
- C. Complement of every context-free language is recursive.
- D. Every nondeterministic PDA can be converted to an equivalent deterministic PDA.

gate2009 theory-of-computation easy isro2017 non-determinism

[Answer](#)

5.8.7 Non Determinism: GATE2011-8 [top](#)

<http://gateoverflow.in/2110>

Which of the following pairs have **DIFFERENT** expressive power?

- (A) Deterministic finite automata (DFA) and Non-deterministic finite automata (NFA)
- (B) Deterministic push down automata (DPDA) and Non-deterministic push down automata (NPDA)
- (C) Deterministic single tape Turing machine and Non-deterministic single tape Turing machine
- (D) Single tape Turing machine and multi-tape Turing machine

gate2011 theory-of-computation easy non-determinism

[Answer](#)

Answers: Non Determinism

5.8.1 Non Determinism: GATE1992-02,xx [top](#)

<http://gateoverflow.in/577>



Selected Answer

Answer: A and C.

- For every NFA there exists a DFA.
- For every NPDA there does not exist a deterministic PDA.
- Every nondeterministic Turing machine has an equivalent deterministic Turing Machine.

5 votes

-- Rajarshi Sarkar (34.4k points)

5.8.2 Non Determinism: GATE1994-1.16 [top](#)

<http://gateoverflow.in/2459>



Selected Answer

This will be C. Because if that would have been possible then NPDA and DPDA must have had same powers, which is not the case. You can take example of NFA and DFA. Both are convertible to each other and hence share the same power.

11 votes

-- Gate Keeda (18.8k points)

5.8.3 Non Determinism: GATE1998-1.11 [top](#)

<http://gateoverflow.in/1648>



Selected Answer

- A) True. Proof - Subset Construction Procedure
- B) False . No conversion from NPDA To DPDA>
- C) False :- Power(TM) > NPDA > DPDA.
- D) True E) True

Answer :- This question has multiple answers, B and C>

7 votes

-- Akash (41.5k points)

5.8.4 Non Determinism: GATE2004-IT-9 [top](#)

<http://gateoverflow.in/3650>



Selected Answer

- A) this is true for inherently ambiguous language
- B) always correct, that's why called unambiguous
- C) NPDA is a superset of DPDA, hence it's FALSE
- D) finite language is always regular

12 votes

-- Manu Thakur (5.8k points)

5.8.5 Non Determinism: GATE2005-54 [top](#)

<http://gateoverflow.in/1377>



Selected Answer

Option D.

NFA and DFA both have equivalent power.(every nfa can be converted into equivalent dfa) but NPDA can accept more languages than DPDA.

6 votes

-- shreya ghosh (3.4k points)

5.8.6 Non Determinism: GATE2009-16, ISRO2017-12 [top](#)

<http://gateoverflow.in/1308>



Selected Answer

Option d ndpa is more powerful than dpda so they are not equivalent..actually dpda is a proper subset of ndpa...

C is TRUE as CFL is a proper subset of recursive languages and recursive languages are closed under complement.

14 votes

-- Bhagirathi Nayak (13.1k points)

5.8.7 Non Determinism: GATE2011-8 [top](#)

<http://gateoverflow.in/2110>



Selected Answer

Expressing power of any machine can be defined as the maximum number of languages it can accept..if machine M1 can accept more languages than M2 then we can say that expressing power of M1 is greater than M2.

a)languages accepted by NFA,will also be accepted by DFA because we can make DFA corresponding to NFA.so their expressing power is same.

b)In this case languages accepted by NPDA is more than DPDA ,so expressing power of NPDA is more than DPDA

c) both deterministic and non deterministic turing can accept same language.so there expressing power is same.

d) In turing machine if we increase the number of tape then also language accepted by that machine is same as single tape turing machine.so there expressing power is same.

ans is b

13 votes

-- neha pawar (4.4k points)

(B) Deterministic push down automata (DPDA) and Non-deterministic push down automata (NPDA)

In rest of the options both machine are equivalent in power.

10 votes

-- Omesh Pandita (2.7k points)

5.9

Pumping Lemma(1) top

5.9.1 Pumping Lemma: GATE2005-IT-40 top

<http://gateoverflow.in/3787>

A language L satisfies the Pumping Lemma for regular languages, and also the Pumping Lemma for context-free languages. Which of the following statements about L is TRUE?

- A. L is necessarily a regular language.
- B. L is necessarily a context-free language, but not necessarily a regular language.
- C. L is necessarily a non-regular language.
- D. None of the above

[gate2005-it](#) [theory-of-computation](#) [pumping-lemma](#) [easy](#)

[Answer](#)

Answers: Pumping Lemma

5.9.1 Pumping Lemma: GATE2005-IT-40 top

<http://gateoverflow.in/3787>



Selected Answer

answer is (D). If a language is regular, it definitely satisfies pumping lemma. But converse need not be true. If a language satisfies pumping lemma then it may or may not be regular.

10 votes

-- Rajat Sharma (539 points)

By pumping lemma we can never say that a language is regular or cfg .it can only be used to prove that a certain lang is not reg or not cfg

16 votes

-- Bhagirathi Nayak (13.1k points)

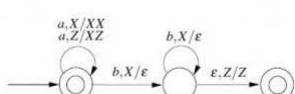
5.10

Pushdown Automata(13) top

5.10.1 Pushdown Automata: GATE 2016-1-43 top

<http://gateoverflow.in/39732>

Consider the transition diagram of a PDA given below with input alphabet $\Sigma = \{a, b\}$ and stack alphabet $\Gamma = \{X, Z\}$. Z is the initial stack symbol. Let L denote the language accepted by the PDA



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

- A. $L = \{a^n b^n \mid n \geq 0\}$ and is not accepted by any finite automata
 B. $L = \{a^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}$ and is not accepted by any deterministic PDA
 C. L is not accepted by any Turing machine that halts on every input
 D. $L = \{a^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}$ and is deterministic context-free

gate2016-1 theory-of-computation pushdown-automata normal

[Answer](#)

5.10.2 Pushdown Automata: GATE1996_13 [top](#)

<http://gateoverflow.in/2765>

Let $Q = (\{q_1, q_2\}, \{a, b\}, \{a, b, \perp\}, \delta, \perp, \phi)$ be a pushdown automaton accepting by empty stack for the language which is the set of all nonempty even palindromes over the set $\{a, b\}$. Below is an incomplete specification of the transitions δ . Complete the specification. The top of the stack is assumed to be at the right end of the string representing stack contents.

1. $\delta(q_1, a, \perp) = \{(q_1, \perp a)\}$
2. $\delta(q_1, b, \perp) = \{(q_1, \perp b)\}$
3. $\delta(q_1, a, a) = \{(q_1, aa)\}$
4. $\delta(q_1, b, a) = \{(q_1, ab)\}$
5. $\delta(q_1, a, b) = \{(q_1, ba)\}$
6. $\delta(q_1, b, b) = \{(q_1, bb)\}$
7. $\delta(q_1, a, a) = \{(\dots, \dots)\}$
8. $\delta(q_1, b, b) = \{(\dots, \dots)\}$
9. $\delta(q_2, a, a) = \{(q_2, \epsilon)\}$
10. $\delta(q_2, b, b) = \{(q_2, \epsilon)\}$
11. $\delta(q_2, \epsilon, \perp) = \{(q_2, \epsilon)\}$

gate1996 theory-of-computation pushdown-automata normal

[Answer](#)

5.10.3 Pushdown Automata: GATE1997_6.6 [top](#)

<http://gateoverflow.in/2262>

Which of the following languages over $\{a, b, c\}$ is accepted by a deterministic pushdown automata?

- (A) $\{wcw^R \mid w \in \{a, b\}^*\}$
 (B) $\{ww^R \mid w \in \{a, b, c\}^*\}$
 (C) $\{a^n b^n c^n \mid n \geq 0\}$
 (D) $\{w \mid w \text{ is a palindrome over } \{a, b, c\}\}$

Note: w^R is the string obtained by reversing ' w '.

gate1997 theory-of-computation pushdown-automata easy

[Answer](#)

5.10.4 Pushdown Automata: GATE1998_13 [top](#)

<http://gateoverflow.in/1727>

Let $M = (\{q_0, q_1\}, \{0, 1\}, \{z_0, X\}, \delta, q_0, z_0, \phi)$ be a Pushdown automation where δ is given by

- $\delta(q_0, 1, z_0) = \{(q_0, xz_0)\}$
 $\delta(q_0, \epsilon, z_0) = \{(q_0, \epsilon)\}$
 $\delta(q_0, 1, X) = \{(q_0, XX)\}$
 $\delta(q_1, 1, X) = \{(q_1, \epsilon)\}$
 $\delta(q_0, 0, X) = \{(q_1, X)\}$
 $\delta(q_0, 0, z_0) = \{(q_0, z_0)\}$ z

- What is the language accepted by this PDA by empty stack?
- Describe informally the working of the PDA

gate1998 theory-of-computation pushdown-automata descriptive

[Answer](#)

5.10.5 Pushdown Automata: GATE1999_1.6 [top](#)

<http://gateoverflow.in/377>

Let L_1 be the set of all languages accepted by a PDA by final state and L_2 the set of all languages accepted by empty stack. Which of the following is true?

- A) $L_1 = L_2$
- B) $L_1 \supset L_2$
- C) $L_1 \subset L_2$
- D) None

There are two modes of acceptance of DPDA - final state and empty stack. For languages accepted by empty stack there is a prefix property. Explain in simple terms about this property and its importance.

normal theory-of-computation gate1999 pushdown-automata

[Answer](#)

5.10.6 Pushdown Automata: GATE2000-8 [top](#)

<http://gateoverflow.in/679>

A push down automation (pda) is given in the following extended notation of finite state diagram:



The nodes denote the states while the edges denote the moves of the pda. The edge labels are of the form $d, s/s'$ where d is the input symbol read and s, s' are the stack contents before and after the move. For example the edge labeled $1, s/1s$ denotes the move from state q_0 to q_0 in which the input symbol 1 is read and pushed to the stack.

- Introduce two edges with appropriate labels in the above diagram so that the resulting pda accepts the language $\{x^2x^R \mid x \in \{0,1\}^*\}$, x^R denotes reverse of x , by empty stack.
- Describe a non-deterministic pda with three states in the above notation that accept the language $\{0^n 1^m \mid n \leq m \leq 2n\}$ by empty stack

gate2000 theory-of-computation descriptive pushdown-automata

[Answer](#)

5.10.7 Pushdown Automata: GATE2001-6 [top](#)

<http://gateoverflow.in/747>

Give a deterministic PDA for the language $L = \{a^n cb^{2n} \mid n \geq 1\}$ over the alphabet $\Sigma = \{a, b, c\}$. Specify the acceptance state.

gate2001 theory-of-computation normal pushdown-automata

[Answer](#)

5.10.8 Pushdown Automata: GATE2004-IT-40 [top](#)

<http://gateoverflow.in/3683>

Let $M = (K, \Sigma, \Gamma, \Delta, s, F)$ be a pushdown automaton, where

$K = (s, f)$, $F = \{f\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a\}$ and
 $\Delta = \{((s, a, \epsilon), (s, a)), ((s, b, \epsilon), (s, a)), ((s, a, \epsilon), (f, \epsilon)), ((f, a, a), (f, \epsilon)), ((f, b, a), (f, \epsilon))\}$.

Which one of the following strings is not a member of $L(M)$?

- A. aaa
- B. aabab
- C. baaba

D. bab

gate2004-it theory-of-computation pushdown-automata normal

Answer

5.10.9 Pushdown Automata: GATE2005-IT-38 [top](#)<http://gateoverflow.in/3785>

Let P be a non-deterministic push-down automaton (NPDA) with exactly one state, q, and exactly one symbol, Z, in its stack alphabet. State q is both the starting as well as the accepting state of the PDA. The stack is initialized with one Z before the start of the operation of the PDA. Let the input alphabet of the PDA be Σ . Let $L(P)$ be the language accepted by the PDA by reading a string and reaching its accepting state. Let $N(P)$ be the language accepted by the PDA by reading a string and emptying its stack.

Which of the following statements is TRUE?

- A. $L(P)$ is necessarily Σ^* but $N(P)$ is not necessarily Σ^* .
- B. $N(P)$ is necessarily Σ^* but $L(P)$ is not necessarily Σ^* .
- C. Both $L(P)$ and $N(P)$ are necessarily Σ^* .
- D. Neither $L(P)$ nor $N(P)$ are necessarily Σ^*

gate2005-it theory-of-computation pushdown-automata normal

Answer

5.10.10 Pushdown Automata: GATE2006-IT-31 [top](#)<http://gateoverflow.in/3570>

Which of the following languages is accepted by a non-deterministic pushdown automaton (PDA) but NOT by a deterministic PDA?

- A. $\{a^n b^n c^n \mid n \geq 0\}$
- B. $\{a^l b^m c^n \mid l \neq m \text{ or } m \neq n\}$
- C. $\{a^n b^n \mid n \geq 0\}$
- D. $\{a^m b^n \mid m, n \geq 0\}$

gate2006-it theory-of-computation pushdown-automata normal

Answer

5.10.11 Pushdown Automata: GATE2006-IT-33 [top](#)<http://gateoverflow.in/3572>

Consider the pushdown automaton (PDA) below which runs over the input alphabet $\{a, b, c\}$. It has the stack alphabet $\{Z_0, X\}$ where Z_0 is the bottom-of-stack marker. The set of states of the PDA is $\{s, t, u, f\}$ where s is the start state and f is the final state. The PDA accepts by final state. The transitions of the PDA given below are depicted in a standard manner. For example, the transition $(s, b, X) \rightarrow (t, XZ_0)$ means that if the PDA is in state s and the symbol on the top of the stack is X, then it can read b from the input and move to state t after popping the top of stack and pushing the symbols Z_0 and X (in that order) on the stack.

- $$\begin{aligned} (s, a, Z_0) &\rightarrow (s, XXZ_0) \\ (s, \epsilon, Z_0) &\rightarrow (f, \epsilon) \\ (s, a, X) &\rightarrow (s, XXX) \\ (s, b, X) &\rightarrow (t, \epsilon) \\ (t, b, X) &\rightarrow (t, \epsilon) \\ (t, c, X) &\rightarrow (u, \epsilon) \\ (u, c, X) &\rightarrow (u, \epsilon) \\ (u, \epsilon, Z_0) &\rightarrow (f, \epsilon) \end{aligned}$$

The language accepted by the PDA is

- A. $\{a^l b^m c^n \mid l = m = n\}$
- B. $\{a^l b^m c^n \mid l = m\}$
- C. $\{a^l b^m c^n \mid 2l = m + n\}$
- D. $\{a^l b^m c^n \mid m = n\}$

gate2006-it theory-of-computation pushdown-automata normal

Answer

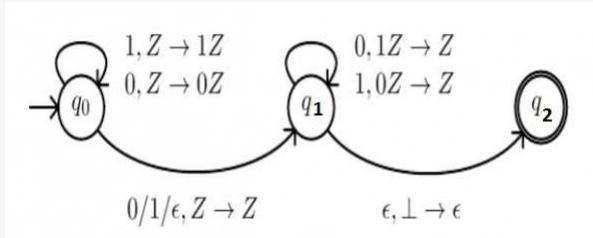
5.10.12 Pushdown Automata: GATE2015-1-51 [top](#)

<http://gateoverflow.in/8357>

Consider the NPDA

$$\langle Q = \{q_0, q_1, q_2\}, \Sigma = \{0, 1\}, \Gamma = \{0, 1, \perp\}, \delta, q_0, \perp, F = \{q_2\} \rangle$$

, where (as per usual convention) Q is the set of states, Σ is the input alphabet, Γ is the stack alphabet, δ is the state transition function q_0 is the initial state, \perp is the initial stack symbol, and F is the set of accepting states. The state transition is as follows:



Which one of the following sequences must follow the string 101100 so that the overall string is accepted by the automaton?

- A. 10110
- B. 10010
- C. 01010
- D. 01001

[gate2015-1](#) [theory-of-computation](#) [pushdown-automata](#) [normal](#)

[Answer](#)

5.10.13 Pushdown Automata: ISI 2015 PCB C4 A [top](#)

<http://gateoverflow.in/121031>

Design a context free grammar for the language consisting of all strings over $\{a, b\}$ that are not of the form ww for any string w

[pushdown-automata](#) [theory-of-computation](#) [isi2015](#)

[Answer](#)

Answers: Pushdown Automata

5.10.1 Pushdown Automata: GATE 2016-1-43 [top](#)

<http://gateoverflow.in/39732>



Selected Answer

Strings accepted at
 I^{st} final state are
 $a^n, n \geq 0$ and strings accepted at
 II^{nd} final state are
 $a^n b^n, n \geq 0$ (actually
 $n \geq 1$ at this state,
 $n = 0$ already included at first state).

$$L = \{a^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}$$

Language is deterministic context-free accepted by DPDA (dpda is already given) and so by TM too, and not regular (as we need stack to implement it), and so cannot be accepted by FA

30 votes

-- Praveen Saini (5.3k points)

Ya it should be D option. As 1st state is also the final state

13 votes

-- Sreyas S (1.8k points)

5.10.2 Pushdown Automata: GATE1996_13 [top](#)

<http://gateoverflow.in/2765>



Selected Answer

$\delta(q_1, a, b) = \{(q_2, ba)\}$ means from state q_1 on input a with stack top being b , the PDA moves to state q_2 and pushes a on top of stack.

So, here the missing transitions are at the middle of the input string:

$$\begin{aligned}\delta(q_1, a, a) &= \{(q_2, \epsilon)\} \\ \delta(q_1, b, b) &= \{(q_2, \epsilon)\}\end{aligned}$$

Once middle is reached, now we should start popping. And so, we must go to state q_2 as well as pop the previous character on the stack. (The character before and after the middle must be same as the string is even length palindrome)

(This is a non-deterministic PDA)

9 votes

-- Arjun Suresh (287k points)

5.10.3 Pushdown Automata: GATE1997_6.6 [top](#)

<http://gateoverflow.in/2262>



Selected Answer

(A) $\{wcw^R \mid w \in \{a, b\}^*\}$ //Can be realized using DPDA cz we know the center of the string that is c here //Hence Option A is ANS

(B) $\{ww^R \mid w \in \{a, b, c\}^*\}$ //((set of even palindrom)NPDA bcz we can't find deterministically the center of palindrom string

cfg will $\rightarrow S \rightarrow aSa \mid bSb \mid cSc \mid \epsilon$

(C) $\{a^n b^n c^n \mid n \geq 0\}$ //CSL

(D) $\{w \mid w \text{ is a palindrome over } \{a, b, c\}\}$ //it is a CFL similar as option B

cfg will $\rightarrow S \rightarrow aSa \mid bSb \mid cSc \mid a \mid b \mid c \mid \epsilon$

6 votes

-- Rajesh Pradhan (16.9k points)

5.10.4 Pushdown Automata: GATE1998_13 [top](#)

<http://gateoverflow.in/1727>



Selected Answer

q_0 is start state

$$\delta(q_0, 0, Z_0) = (q_0, Z_0)$$

[Do Nothing operation, just read any no of 0's but do not keep in stack (any no of 0's bcoz on reading 0's it remains on same state q_0)]

$$\delta(q_0, 1, Z_0) = (q_0, XZ_0) \quad [\text{Read first 1 and keep one } X \text{ in stack}]$$

$$\delta(q_0, 1, X) = (q_0, XX) \quad [\text{Read any no of 1's and keep one } X \text{ for each 1 in stack}]$$

$$\delta(q_0, 0, X) = (q_1, X)$$

[Read single 0 and do nothing in stack, state changed from q_0 to q_1]

$$\delta(q_1, 1, X) = (q_1, \epsilon)$$

[Pop out one X from stack on reading each 1 on state q_1 (matching each 1 with the 1 read before single 0)]

$$\delta(q_0, \epsilon, Z_0) = (q_0, \epsilon)$$

[stack is empty , inputs are accepted here ,that is , ϵ or any of 0's (we read earlier with Do Nothing operation)]

$L = \{ 0^m, m \geq 0 \}$

No input accept after reaching on q_1 because stack will remain with Z_0 , stack initial symbol

Note : if we add one more transition

$\delta(q_1, \epsilon, Z_0) = (q_1, \epsilon)$, then L will be $\{ 0^m \cup 0^i 1^j 0^j, m, i, j \geq 0 \}$

10 votes

-- Praveen Saini (53k points)

5.10.5 Pushdown Automata: GATE1999_1.6 [top](#)



Selected Answer

Answer to the question is (A) $L_1 = L_2$.

Reason is for any PDA which accepts by final state there is an equivalent PDA (equivalent means that accepts the same language) which accepts by empty stack and vice-versa.

Now, this is not the case for DPDA's.

The set of languages accepted by a DPDA by empty stack is a strict subset of the set of languages accepted by a DPDA by final state.

It can also be said that set of languages accepted by a DPDA by empty stack is the set of languages accepted by a DPDA by final state and which has the prefix property.

A language has prefix property means if $w \in L$, then no proper prefix of $w \in L$.

From the above definition of prefix property it must be clear why DPDA by empty stack has this property. If any prefix of a word w (w in L) is in L means the stack should have been empty even before completely processing w . But, being a deterministic PDA, once the stack becomes empty, the DPDA accepts and halts. So, in no way can a DPDA accept w and its prefix.

PS: A DPDA with acceptance by empty stack cannot even accept all regular languages- example a^* .

Good read: http://www.cs.ucr.edu/~jiang/cs150/slides4week7_PDA+EquivToCFG.pdf

20 votes

-- gatecse (13.1k points)

5.10.6 Pushdown Automata: GATE2000-8 [top](#)

<http://gateoverflow.in/679>

in a) $x2xR$

say for some word 0112110 we have to push every thing into the stack till 2 . then we get 1 then 1 will be at top of stack so pop it or if get 0 then 0 will at top of stack so pop it. For any word of language it is applicable. 2 is a mark that tell now we have to pop 0 for 0 and 1 for 1.

so on the edge q_0 to q_0 add $0,s/0,s$

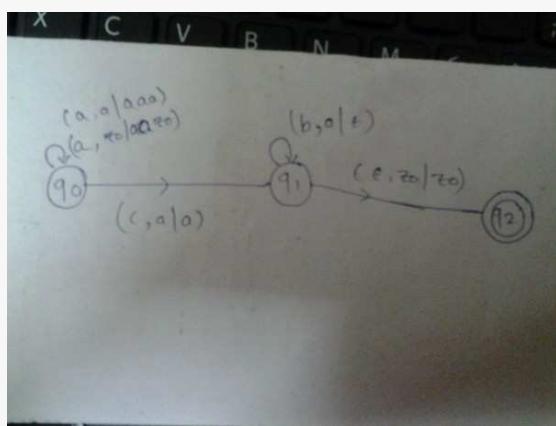
and on edge q_1 to q_1 add $0,0,s/s$

6 votes

-- Praveen Saini (53k points)

5.10.7 Pushdown Automata: GATE2001-6 [top](#)

<http://gateoverflow.in/747>



$L = \{a^n c b^{2n}\}$

$= \{acbb, aacbbbb, aaacbbbbbb, \dots\}$

Here 'c' act as center. Push 2 a's for 1 'a' and when you see 'c' pop the b's

3 votes

-- Saraswati Walijkar (321 points)

5.10.8 Pushdown Automata: GATE2004-IT-40 [top](#)

<http://gateoverflow.in/3683>



Selected Answer

ANS : B

The language is like.....

in start state a's or b's come....just push a's on stack...except for the last a...which is used to shift from "start state" to "final state" without consuming any stack symbol....now in "final state" , for equal no's of a's and b's just pop a's from stack.....that the interpretation of transitions given in language.

The transitions given are obviously WRONGbut what they must have meant is same as i explained.

Based on this.....

- 1) aaa : push a....skip a....pop a ACCEPTED
- 2)aabab: push a....skip a...pop a... REJECTED
- 3)baaba : push a...push a....skip a....pop a....pop a ACCEPTED
- 4)bab : push a....skip a....pop a ACCEPTED

6 votes

-- Rohan Mundhey (4.3k points)

5.10.9 Pushdown Automata: GATE2005-IT-38 [top](#)

<http://gateoverflow.in/3785>



Selected Answer

Answer is (D)

In NPDA we may have a dead configuration. This means we may not give any transition to any alphabet from this state.

we say that a string is accepted if PDA is in final state after reading the final symbol in the string or after it has read '\$' symbol denoting end of the string and **it is in final state**.

Now coming to options:

Question never says that we have transitions defined for all the alphabet symbols in the PDA. Although it is ALREADY in the FINAL state we may not have ANY transition for any input symbol. In this case string will be rejected as it will never finish reading the string.

To sum up: A string is rejected in following two ways:

1. If no transition is defined for any configuration(this includes the final state as well because to accept the string we need the transition $(f, \$, _) \rightarrow (f, _)$ in final state or accepting state where blank('_) denotes arbitrary stack symbol that does not matter because we are not accepting by EMPTY stack

2. If string enters a configuration for which no transition is defined, STRING is rejected.

So option (D) is correct. Because the same way it may not empty the stack when it finishes reading the string.

15 votes

-- Sandeep_Uniyal (7.3k points)

5.10.10 Pushdown Automata: GATE2006-IT-31 [top](#)

<http://gateoverflow.in/3570>



Selected Answer

option B is correct

$$L = \{a^l b^m c^n \mid l \neq m \text{ or } m \neq n\}$$

(q_0, a, Z_0)
 $\rightarrow (q_0, aZ_0)$

(q_0, a, a)
 $\rightarrow (q_0, aa)$

(q_0, b, a)
 $\rightarrow (q_1, \epsilon), (q_2, ba)$

[here it is NPDA where we have to check $l \neq m$ or $m \neq n$; for $l \neq m$ we need to pop a for b ; for $m \neq n$ we need to keep b in stack so that we can pop b for c]

(q_1, b, a)
 $\rightarrow (q_1, \epsilon)$

(q_1, c, a)
 $\rightarrow (q_f, \epsilon)$

(q_1, b, Z_0)
 $\rightarrow (q_f, \epsilon)$

(q_2, b, b)
 $\rightarrow q_2, bb)$

(q_2, c, b)
 $\rightarrow (q_3, \epsilon)$

(q_3, c, b)
 $\rightarrow (q_3, \epsilon)$

(q_3, c, a)
 $\rightarrow (q_f, \epsilon)$

(q_3, ϵ, b)
 $\rightarrow (q_f, \epsilon)$

A is wrong as it is not context free

D a^*b^* is regular, so must have DFA, and so an equivalent DPDA

C can be accepted using DPDA

12 votes

-- Praveen Saini (53k points)

5.10.11 Pushdown Automata: GATE2006-IT-33 [top](#)

<http://gateoverflow.in/3572>

for every a we put two X in stack [at state s]

after that for every b we pop out one X [reach to state t (getting b after a)]

after that for every c we pop out one X [reach to state u (getting c after b)]

if all X are popped out then reached to final state f , mean for every b there is a, for every c there is a .

a was followed by b and b was followed by c [state s to t , t to u , u to f, final]

means sum of no of b's and no of c's = twice of no of a's [one a for one b , one a for one c]

i.e. $\{a^l b^m c^n \mid 2l = m + n\}$

 14 votes

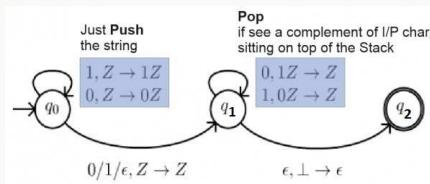
-- Praveen Saini (53k points)

5.10.12 Pushdown Automata: GATE2015-1-51 [top](#)

<http://gateoverflow.in/8357>

Here, Z is used to represent the entire stack content except the *top*. Z is the string in Stack read from top to bottom. $1, Z \rightarrow 1Z$ means, on input symbol 1, the stack content changes from Z to $1Z$.

In q_0 state, for '1', a '1' is pushed and for a '0' a '0' is pushed. In q_1 state, for a '0' a '1' is popped and for a '1' a '0' is popped. So, the given PDA is accepting all strings of the form $x0x'_r$ or $x1x'_r$ or xx'_r , where x'_r is the reverse of the 1's complement of x . i.e.:



The given string 101100 has 6 letters and we are given 5 letter strings. So, $x0$ is done, with $x = 10110$. So, $x'_r = (01001)_r = 10010$.

answer = **option B**

 27 votes

-- Arjun Suresh (287k points)

5.10.13 Pushdown Automata: ISI 2015 PCB C4 A [top](#)

<http://gateoverflow.in/121031>

Now, this is what we called a problem.

Here is the grammar,

$$S \rightarrow E \cup U \cup \epsilon$$

$E \rightarrow AB|BA$

$A \rightarrow ZAZ|a$

$B \rightarrow ZBZ|b$

$U \rightarrow ZUZ|Z$

$Z \rightarrow a|b$

For more Explore [this](#).

5 votes

-- Muktinath Vishwakarma (33.9k points)

5.11

Recursive And Recursively Enumerable Languages(13) [top](#)

5.11.1 Recursive And Recursively Enumerable Languages: GATE 2016-1-44 [top](#)

Let X be a recursive language and Y be a recursively enumerable but not recursive language. Let \overline{W} and \overline{Z} be two languages such that \overline{Y} reduces to \overline{W} , and \overline{Z} reduces to \overline{X} (reduction means the standard many-one reduction). Which one of the following statements is TRUE?

- A. \overline{W} can be recursively enumerable and \overline{Z} is recursive.
- B. \overline{W} can be recursive and \overline{Z} is recursively enumerable.
- C. \overline{W} is not recursively enumerable and \overline{Z} is recursive.
- D. \overline{W} is not recursively enumerable and \overline{Z} is not recursive.

gate2016-1 theory-of-computation easy recursive-and-recursively-enumerable-languages

Answer

5.11.2 Recursive And Recursively Enumerable Languages: GATE1990-3-vi [top](#)

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

Recursive languages are:

- A. A proper superset of context free languages.
- B. Always recognizable by pushdown automata.
- C. Also called type \emptyset languages.
- D. Recognizable by Turing machines.

gate1990 normal theory-of-computation turing-machine recursive-and-recursively-enumerable-languages

Answer

5.11.3 Recursive And Recursively Enumerable Languages: GATE2003-13 [http://gateoverflow.in/904](#)

top

Nobody knows yet if $P=NP$. Consider the language L defined as follows.

$$L = \begin{cases} (0+1)^* & \text{if } P=NP \\ \phi & \text{otherwise} \end{cases}$$

Which of the following statements is true?

- A. L is recursive
- B. L is recursively enumerable but not recursive
- C. L is not recursively enumerable
- D. Whether L is recursively enumerable or not will be known after we find out if $P=NP$

gate2003 theory-of-computation normal recursive-and-recursively-enumerable-languages

Answer

5.11.4 Recursive And Recursively Enumerable Languages: GATE2003-15 <http://gateoverflow.in/120>

top

If the strings of a language L can be effectively enumerated in lexicographic (i.e., alphabetic) order, which of the following statements is true?

- A. L is necessarily finite
- B. L is regular but not necessarily finite
- C. L is context free but not necessarily regular
- D. L is recursive but not necessarily context-free

theory-of-computation gate2003 normal recursive-and-recursively-enumerable-languages

Answer

5.11.5 Recursive And Recursively Enumerable Languages: GATE2005-56 <http://gateoverflow.in/1379>

top

Let L_1 be a recursive language, and let L_2 be a recursively enumerable but not a recursive language. Which one of the following is TRUE?

- A. L_1' is recursive and L_2' is recursively enumerable
- B. L_1' is recursive and L_2' is not recursively enumerable
- C. L_1' and L_2' are recursively enumerable
- D. L_1' is recursively enumerable and L_2' is recursive

gate2005 theory-of-computation recursive-and-recursively-enumerable-languages easy

Answer

5.11.6 Recursive And Recursively Enumerable Languages: GATE2008-13, ISRO2016-36 <http://gateoverflow.in/411>

top

If L and \bar{L} are recursively enumerable then L is

- A. regular
- B. context-free
- C. context-sensitive
- D. recursive

gate2008 theory-of-computation easy isro2016 recursive-and-recursively-enumerable-languages

Answer

5.11.7 Recursive And Recursively Enumerable Languages: GATE2008-48 <http://gateoverflow.in/461>

top

Which of the following statements is false?

- A. Every NFA can be converted to an equivalent DFA
- B. Every non-deterministic Turing machine can be converted to an equivalent deterministic Turing machine
- C. Every regular language is also a context-free language
- D. Every subset of a recursively enumerable set is recursive

gate2008 theory-of-computation easy recursive-and-recursively-enumerable-languages

Answer

5.11.8 Recursive And Recursively Enumerable Languages: GATE2010-17 <http://gateoverflow.in/2190>

top

Let L_1 be the recursive language. Let L_2 and L_3 be languages that are recursively enumerable but not recursive. Which of the following statements is not necessarily true?

- A. $L_2 - L_1$ is recursively enumerable.
- B. $L_1 - L_3$ is recursively enumerable.
- C. $L_2 \cap L_3$ is recursively enumerable.
- D. $L_2 \cup L_3$ is recursively enumerable.

gate2010 theory-of-computation recursive-and-recursively-enumerable-languages decidability normal

[Answer](#)

5.11.9 Recursive And Recursively Enumerable Languages: GATE2014-1-35 [top](#)

Let L be a language and \bar{L} be its complement. Which one of the following is NOT a viable possibility?

- A. Neither L nor \bar{L} is recursively enumerable (r.e.).
- B. One of L and \bar{L} is r.e. but not recursive; the other is not r.e.
- C. Both L and \bar{L} are r.e. but not recursive.
- D. Both L and \bar{L} are recursive.

gate2014-1 theory-of-computation easy recursive-and-recursively-enumerable-languages

[Answer](#)

5.11.10 Recursive And Recursively Enumerable Languages: GATE2014-2-16 [top](#)

Let $A \leq_m B$ denotes that language A is mapping reducible (also known as many-to-one reducible) to language B. Which one of the following is FALSE?

- A. If $A \leq_m B$ and B is recursive then A is recursive.
- B. If $A \leq_m B$ and A is undecidable then B is undecidable.
- C. If $A \leq_m B$ and B is recursively enumerable then A is recursively enumerable.
- D. If $A \leq_m B$ and B is not recursively enumerable then A is not recursively enumerable.

gate2014-2 theory-of-computation recursive-and-recursively-enumerable-languages normal

[Answer](#)

5.11.11 Recursive And Recursively Enumerable Languages: GATE2015-1_3 [top](#)

For any two languages L_1 and L_2 such that L_1 is context-free and L_2 is recursively enumerable but not recursive, which of the following is/are necessarily true?

- I. \bar{L}_1 (Compliment of L_1) is recursive
- II. \bar{L}_2 (Compliment of L_2) is recursive
- III. \bar{L}_1 is context-free
- IV. $\bar{L}_1 \cup \bar{L}_2$ is recursively enumerable

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

gate2015-1 theory-of-computation recursive-and-recursively-enumerable-languages normal

[Answer](#)

5.11.12 Recursive And Recursively Enumerable Languages: TIFR2010-B-40 [top](#)

Which of the following statement is FALSE?

- a. All recursive sets are recursively enumerable.
- b. The complement of every recursively enumerable sets is recursively enumerable.
- c. Every Non-empty recursively enumerable set is the range of some totally recursive function.
- d. All finite sets are recursive.
- e. The complement of every recursive set is recursive.

tifr2010 | theory-of-computation | recursive-and-recursively-enumerable-languages

Answer

5.11.13 Recursive And Recursively Enumerable Languages: TIFR2012-B-19 [top](#)

Which of the following statements is TRUE?

- Every turning machine recognizable language is recursive.
- The complement of every recursively enumerable language is recursively enumerable.
- The complement of a recursive language is recursively enumerable.
- The complement of a context-free language is context-free.
- The set of turning machines which do not halt on empty input forms a recursively enumerable set.

tifr2010 | theory-of-computation | recursive-and-recursively-enumerable-languages

Answer

Answers: Recursive And Recursively Enumerable Languages

5.11.1 Recursive And Recursively Enumerable Languages: GATE 2016-1-44 [top](#)



Selected Answer

\overline{X} is recursive language, so
 $\overline{\overline{X}}$ is also recursive.

\overline{Y} is recursively enumerable, but not recursive so
 $\overline{\overline{Y}}$ is not recursively enumerable language.

$A \leq B$, (A is reducible to B), i.e., solving A cannot be "harder" than solving B.

- if A is reducible to B, and B is decidable, then A is decidable.
 - if A is reducible to B, and B is recursive, then A is recursive.
- if A is undecidable, and reducible to B, then B is undecidable.
 - if B is recursively enumerable, and A is reducible to B, then A is recursively enumerable.
 - if A is not recursively enumerable, and reducible to B, then B is not recursively enumerable.

Now Back to question.

\overline{Y} is not recursively enumerable, and reducible to
 W , then
 W is not recursively enumerable (using 2(ii)).

Z is reducible to
 \overline{X} and
 \overline{X} is recursive, then
 Z is recursive (using 1(i)).

Option C is correct

28 votes

-- Praveen Saini (53k points)

5.11.2 Recursive And Recursively Enumerable Languages: GATE1990-3-vi [top](#)



Selected Answer

A. A proper superset of context free languages. **TRUE**, Since there are languages which are not CFL still Recursive

B. Always recognizable by pushdown automata. **FALSE**

C. Also called type 0 languages. **FALSE**, R.E languages are actually type-0 languages

[http://gateoverflow.in/94833](#)

D.Recognizable by Turing machines,**TRUE**

4 votes

-- Prajwal Bhat (11.6k points)

5.11.3 Recursive And Recursively Enumerable Languages: GATE2003-13 <http://gateoverflow.in/904>

[top](#)



(A) L is recursive. If $P = NP$, L is Σ^* which is recursive (in fact regular). If not, $L = \emptyset$ which is again recursive. So, in both cases L is recursive.

16 votes

-- Arjun Suresh (287k points)

5.11.4 Recursive And Recursively Enumerable Languages: GATE2003-15 <http://gateoverflow.in/120>

[top](#)



Answer is (c)- L is recursive but not necessarily Regular or not even context-free.

Since, the strings of L can be enumerated it means L is recursively enumerable. That is we have a TM which accepts all strings in L . Now, to be recursive the TM should reject all strings not in L . Since, the strings of the language can be enumerated in lexicographic order, it's easy to do this. For any word w , if we see a word in the enumeration which is lexicographically higher than w but no w , it means w is not in the language. This makes L "recursive".

Now, why L need not be context free or regular? Consider

$$L = \{a^n b^n c^n \mid \dots\}$$

The strings of this language can be enumerated in lexicographic order. But we know L is not context free as no PDA can accept L .

30 votes

-- Arjun Suresh (287k points)

5.11.5 Recursive And Recursively Enumerable Languages: GATE2005-56 <http://gateoverflow.in/1379>

[top](#)



L_1 being recursive, we have a TM M for L_1 which accepts all words in L_1 and rejects all words in L_1' . So, this TM also works for L_1' by changing the accept and reject states. Thus L_1' is recursive.

L_2 being recursively enumerable but not recursive means TM for L_2 can accept all words in L_2 but cannot reject all words not in L_2 => TM for L_2' cannot exist (as otherwise TM for L_2 could simulate the moves of that TM to reject words in L_2')=> L_2' is not recursively enumerable. So, (B).

12 votes

-- Arjun Suresh (287k points)

5.11.6 Recursive And Recursively Enumerable Languages: GATE2008-13, ISRO2016-36 <http://gateoverflow.in/411>



(D) recursive

L is recursively enumerable means a TM accepts all strings in L . \bar{L} is recursively enumerable means a TM accepts all strings in \bar{L} . So, we can always decide if a string is in L or not, making L recursive.

If a language L and its complement \bar{L} are both recursively enumerable, then both languages are recursive. If L is recursive, then \bar{L} is also recursive, and consequently both are recursively enumerable.

Proof: If L and \bar{L} are both recursively enumerable, then there exist Turing machines M and \bar{M} that serve as enumeration procedures for L and \bar{L} , respectively. The first will produce w_1, w_2, \dots in L , the second $\hat{w}_1, \hat{w}_2, \dots$ in \bar{L} . Suppose now we are given any $w \in \Sigma^+$. We first let M generate w_1 and compare it with w . If they are not the same, we let M generate w_1 and compare again. If we need to continue, we next let M generate w_2 , then \bar{M} generate \hat{w}_2 , and so on. Any $w \in \Sigma^+$ will be generated by either M or \bar{M} , so eventually we will get a match. If the matching string is produced by M , w belongs to L , otherwise it is in \bar{L} . The process is a membership algorithm for both L and \bar{L} , so they are both recursive.

For the converse, assume that L is recursive. Then there exists a membership algorithm for it. But this becomes a membership algorithm for \bar{L} by simply complementing its conclusion. Therefore, \bar{L} is recursive. Since any recursive language is recursively enumerable, the proof is completed. ■

<http://goo.gl/RtV8MO>

14 votes

-- Keith Kr (6.2k points)

5.11.7 Recursive And Recursively Enumerable Languages: GATE2008-48 http://gateoverflow.in/461

top



Selected Answer

there exists a set of languages which is RE but not REC(i.e. Recursively Enumerable but not Recursive), this set is a subset of RE but is Not Recursive

option D tells us that every subset of RE is REC this is false,
Hence, **option D** is chosen

11 votes

-- Amar Vashishth (27.9k points)

5.11.8 Recursive And Recursively Enumerable Languages: GATE2010-17 http://gateoverflow.in/2190

top



Selected Answer

Recursively enumerable languages are closed under union and intersection. So, lets consider each option

(A) $L_2 - L_1 = L_2 \cap \bar{L}_1$

Recursive languages are closed under complement, and so \bar{L}_1 is recursive and hence recursively enumerable also. So, $L_2 \cap \bar{L}_1$ is recursively enumerable is always TRUE.

(B) $L_1 - L_3 = L_1 \cap \bar{L}_3$

Recursively enumerable languages are not closed under complement. So, \bar{L}_3 may or may not be recursively enumerable and hence we can't say anything if $L_1 \cap \bar{L}_3$ is recursively enumerable or not.

(C) Intersection of two recursively enumerable languages is always recursively enumerable(RE closed under intersection).

(D) Union of two recursively enumerable languages is always recursively enumerable(RE closed under union).

For verifying closure properties:

http://gatcse.in/wiki/Closure_Property_of_Language_Families

31 votes

-- Arjun Suresh (287k points)

5.11.9 Recursive And Recursively Enumerable Languages: GATE2014-1-35 top



Selected Answer

(C) is not possible. If L is r.e. we have a TM that accepts string in L . If L' is r.e., we have a TM that accepts strings in L' . So, using both these TMs we can make a new TM M which accepts strings in L and rejects strings in $L' -$ that is M decides L , making L recursive.

15 votes

-- Arjun Suresh (287k points)

5.11.10 Recursive And Recursively Enumerable Languages: GATE2014-2-16 [top](#)



Selected Answer

<http://gateoverflow.in/1972>

$A \leq_m B$ means A cannot be harder than B . (Since A can be reduced to B , instead of deciding A , we can now decide B)

So, the first 3 options are correct. Option (D) is false, as B is not recursively enumerable doesn't guarantee A is not recursively enumerable.

14 votes

-- Arjun Suresh (287k points)

5.11.11 Recursive And Recursively Enumerable Languages: GATE2015-1_3 [top](#)



Selected Answer

<http://gateoverflow.in/8019>

D.

L_1 is context-free and hence recursive also. Recursive set being closed under complement, L_1' will be recursive.

L_1' being recursive it is also recursively enumerable and Recursively Enumerable set is closed under Union. So, $L_1' \cup L_2$ is recursively enumerable.

Context free languages are not closed under complement- so III is false

Recursive set is closed under complement. So, if L_2' is recursive, $(L_2')' = L_2$ is also recursive which is not the case here. So, II is also false.

17 votes

-- Arjun Suresh (287k points)

5.11.12 Recursive And Recursively Enumerable Languages: TIFR2010-B-40 [top](#)



Selected Answer

<http://gateoverflow.in/19048>

B) The complement of every recursively enumerable sets is recursively enumerable.
because RE language are not closed under complement.

8 votes

-- Umang Raman (14.5k points)

5.11.13 Recursive And Recursively Enumerable Languages: TIFR2012-B-19 [top](#)



Selected Answer

<http://gateoverflow.in/25218>

- a. False. Turing recognizable language are recursive enumerable and recursive set is a **proper subset** of it.
- b. False, Complement of r.e. **need not** be r.e.
- c. True. Complement of recursive language is recursive and every recursive language is recursive enumerable.
- d. False. Complement of CFL **need not** be CFL (but is guaranteed to be a CSL).
- e. False. **NOT halt** on empty, yes ans not possible. Only no ans is there. So it will be non r.e. while its complement is r.e. but not recursive.

8 votes

-- srestha (55.2k points)

5.12**Regular Expressions(28) [top](#)****5.12.1 Regular Expressions: GATE 1992-2 (xii) Which of the following regular expression identities are true? [top](#)**<http://gateoverflow.in/83956>

- (A) $r(*) = r^*$
- (B) $(r^*s^*)^* = (r + s)^*$
- (C) $(r + s)^* = r^* + s^*$
- (D) $r^*s^* = r^* + s^*$

[gate1992](#) [theory-of-computation](#) [regular-expressions](#)**Answer****5.12.2 Regular Expressions: GATE 2016-1-18 [top](#)**<http://gateoverflow.in/39647>

Which one of the following regular expressions represents the language: *the set of all binary strings having two consecutive 0's and two consecutive 1's?*

- A. $(0 + 1)^*0011(0 + 1)^* + (0 + 1)^*1100(0 + 1)^*$
- B. $(0 + 1)^*(00(0 + 1)^*11 + 11(0 + 1)^*00)(0 + 1)^*$
- C. $(0 + 1)^*00(0 + 1)^* + (0 + 1)^*11(0 + 1)^*$
- D. $00(0 + 1)^*11 + 11(0 + 1)^*00$

[gate2016-1](#) [theory-of-computation](#) [regular-expressions](#) [normal](#)**Answer****5.12.3 Regular Expressions: GATE1987-10d [top](#)**<http://gateoverflow.in/82455>

Give a regular expression over the alphabet $\{0,1\}$ to denote the set of proper non-null substrings of the string 0110.

[gate1987](#) [theory-of-computation](#) [regular-expressions](#)**Answer****5.12.4 Regular Expressions: GATE1991_03,xiii [top](#)**<http://gateoverflow.in/527>

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

Let $r = 1(1 + 0)^*$, $s = 11^*0$ and $t = 1^*0$ be three regular expressions. Which one of the following is true?

- A. $L(s) \subseteq L(r)$ and $L(s) \subseteq L(t)$
- B. $L(r) \subseteq L(s)$ and $L(s) \subseteq L(t)$
- C. $L(s) \subseteq L(t)$ and $L(s) \subseteq L(r)$
- D. $L(t) \subseteq L(s)$ and $L(s) \subseteq L(r)$
- E. None of the above

[gate1991](#) [theory-of-computation](#) [regular-expressions](#) [normal](#)**Answer****5.12.5 Regular Expressions: GATE1992_02,xvii [top](#)**<http://gateoverflow.in/575>

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

Which of the following regular expression identities is/are TRUE?

- (a) $r^{(*)} = r^*$
 (b) $(r^* s^*) = (r + s)^*$
 (c) $(r + s)^* = r^* + s^*$
 (d) $r^* s^* = r^* + s^*$

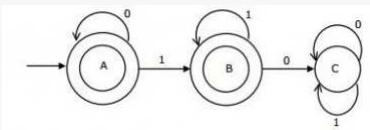
gate1992 theory-of-computation regular-expressions easy

Answer

5.12.6 Regular Expressions: GATE1994_2.10 [top](#)

<http://gateoverflow.in/2477>

The regular expression for the language recognized by the finite state automaton of figure is _____



gate1994 theory-of-computation finite-automata regular-expressions easy

Answer

5.12.7 Regular Expressions: GATE1995-1.9 , ISRO2017-13 [top](#)

<http://gateoverflow.in/2596>

In some programming language, an identifier is permitted to be a letter followed by any number of letters or digits. If L and D denote the sets of letters and digits respectively, which of the following expressions defines an identifier?

- A. $(L + D)^+$
 B. $(L \cdot D)^*$
 C. $L(L + D)^*$
 D. $L(L \cdot D)^*$

gate1995 theory-of-computation regular-expressions easy isro2017

Answer

5.12.8 Regular Expressions: GATE1996_1.8 [top](#)

<http://gateoverflow.in/2712>

Which two of the following four regular expressions are equivalent? (ϵ is the empty string).

- i. $(00)^*(\epsilon + 0)$
 ii. $(00)^*$
 iii. 0^*
 iv. $0(00)^*$

- A. (i) and (ii)
 B. (ii) and (iii)
 C. (i) and (iii)
 D. (iii) and (iv)

gate1996 theory-of-computation regular-expressions easy

Answer

5.12.9 Regular Expressions: GATE1997_6.4 [top](#)

<http://gateoverflow.in/2260>

Which one of the following regular expressions over $\{0,1\}$ denotes the set of all strings not containing 100 as substring?

- (a) $0^*(1+0)^*$
 (b) 0^*1010^*
 (c) $0^*1^*01^*$
 (d) $0^*(10+1)^*$

gate1997 theory-of-computation regular-expressions normal

Answer

5.12.10 Regular Expressions: GATE1998_1.12 [top](#)

<http://gateoverflow.in/1649>

The string 1101 does not belong to the set represented by

- (a) $110^*(0+1)$
 (b) $1(0+1)^*101$
 (c) $(10)^*(01)^*(00+11)^*$
 (d) $(00+(11)^*0)^*$

gate1998 theory-of-computation regular-expressions easy

Answer

5.12.11 Regular Expressions: GATE1998_1.9 [top](#)

<http://gateoverflow.in/1646>

If the regular set A is represented by $A = (01 + 1)^*$ and the regular set B is represented by $B = ((01)^*1^*)^*$, which of the following is true?

- (a) $A \subset B$
 (b) $B \subset A$
 (c) A and B are incomparable
 (d) $A = B$

gate1998 theory-of-computation regular-expressions normal

Answer

5.12.12 Regular Expressions: GATE1998_3b [top](#)

<http://gateoverflow.in/2941>

Give a regular expression for the set of binary strings where every 0 is immediately followed by exactly k 1's and preceded by at least k 1's (k is a fixed integer)

gate1998 theory-of-computation regular-expressions easy

Answer

5.12.13 Regular Expressions: GATE2000-1.4 [top](#)

<http://gateoverflow.in/627>

Let S and T be languages over $\Sigma = \{a, b\}$ represented by the regular expressions $(a+b^*)^*$ and $(a+b)^*$, respectively. Which of the following is true?

- A. $S \subset T$
 B. $T \subset S$
 C. $S = T$
 D. $S \cap T = \emptyset$

gate2000 theory-of-computation regular-expressions easy

Answer

5.12.14 Regular Expressions: GATE2003-14 [top](#)

<http://gateoverflow.in/965>

The regular expression $0^*(10^*)^*$ denotes the same set as

- A. $(1^*0)^*1^*$
 B. $0 + (0+10)^*$
 C. $(0+1)^*10(0+1)^*$

- D. None of the above

gate2003 theory-of-computation regular-expressions easy

[Answer](#)

5.12.15 Regular Expressions: GATE2004-IT-7 [top](#)

<http://gateoverflow.in/3648>

Which one of the following regular expressions is NOT equivalent to the regular expression $(a + b + c)^*$?

- A. $(a^* + b^* + c^*)^*$
- B. $(a^*b^*c^*)^*$
- C. $((ab)^* + c^*)^*$
- D. $(a^*b^* + c^*)^*$

gate2004-it theory-of-computation regular-expressions normal

[Answer](#)

5.12.16 Regular Expressions: GATE2005-IT-5 [top](#)

<http://gateoverflow.in/3749>

Which of the following statements is TRUE about the regular expression 01^*0 ?

- A. It represents a finite set of finite strings.
- B. It represents an infinite set of finite strings.
- C. It represents a finite set of infinite strings.
- D. It represents an infinite set of infinite strings.

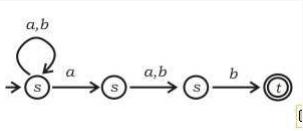
gate2005-it theory-of-computation regular-expressions easy

[Answer](#)

5.12.17 Regular Expressions: GATE2006-IT-5 [top](#)

<http://gateoverflow.in/3544>

Which regular expression best describes the language accepted by the non-deterministic automaton below?



- A. $(a + b)^* a(a + b)b$
- B. $(abb)^*$
- C. $(a + b)^* a(a + b)^* b(a + b)^*$
- D. $(a + b)^*$

gate2006-it theory-of-computation regular-expressions normal

[Answer](#)

5.12.18 Regular Expressions: GATE2007-IT-73 [top](#)

<http://gateoverflow.in/3525>

Consider the regular expression $R = (a + b)^* (aa + bb) (a + b)^*$

Which one of the regular expressions given below defines the same language as defined by the regular expression R?

- A. $(a(ba)^* + b(ab)^*)(a + b)^+$
- B. $(a(ba)^* + b(ab)^*)^*(a + b)^*$
- C. $(a(ba)^* (a + bb) + b(ab)^*(b + aa))(a + b)^*$
- D. $(a(ba)^* (a + bb) + b(ab)^*(b + aa))(a + b)^+$

gate2007-it theory-of-computation regular-expressions normal

[Answer](#)

5.12.19 Regular Expressions: GATE2008-IT-5 [top](#)

<http://gateoverflow.in/3265>

Which of the following regular expressions describes the language over {0, 1} consisting of strings that contain exactly two 1's?

- A. $(0 + 1)^* 11(0 + 1)^*$
- B. $0^* 110^*$
- C. $0^* 10^* 10^*$
- D. $(0 + 1)^* 1(0 + 1)^* 1(0 + 1)^*$

gate2008-it theory-of-computation regular-expressions easy

[Answer](#)

5.12.20 Regular Expressions: GATE2009-15 [top](#)

<http://gateoverflow.in/1307>

Which one of the following languages over the alphabet {0,1} is described by the regular expression: $(0 + 1)^* 0(0 + 1)^* 0(0 + 1)^*$?

- A. The set of all strings containing the substring 00
- B. The set of all strings containing at most two 0's
- C. The set of all strings containing at least two 0's
- D. The set of all strings that begin and end with either 0 or 1

gate2009 theory-of-computation regular-expressions easy

[Answer](#)

5.12.21 Regular Expressions: GATE2010-39 [top](#)

<http://gateoverflow.in/2340>

Let $L = \{w \in (0 + 1)^* \mid w \text{ has even number of } 1\text{s}\}$. i.e., L is the set of all the bit strings with even numbers of 1s. Which one of the regular expressions below represents L ?

- A. $(0^* 10^* 1)^*$
- B. $0^* (10^* 10^*)^*$
- C. $0^* (10^* 1)^* 0^*$
- D. $0^* 1 (10^* 1)^* 10^*$

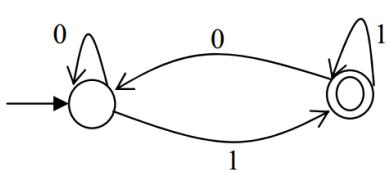
gate2010 theory-of-computation regular-expressions normal

[Answer](#)

5.12.22 Regular Expressions: GATE2014-1-36 [top](#)

<http://gateoverflow.in/1914>

Which of the regular expressions given below represent the following DFA?



- I. $0^* 1 (1 + 00^* 1)^*$
- II. $0^* 1^* 1 + 11^* 0^* 1$
- III. $(0 + 1)^* 1$

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

gate2014-1 theory-of-computation regular-expressions finite-automata easy

[Answer](#)

5.12.23 Regular Expressions: GATE2014-3-15 [top](#)

<http://gateoverflow.in/2049>

The length of the shortest string NOT in the language (over $\Sigma = \{a, b\}$) of the following regular expression is _____.

$$a^*b^*(ba)^*a^*$$

gate2014-3 theory-of-computation regular-expressions numerical-answers easy

[Answer](#)

5.12.24 Regular Expressions: Gateforum test series [top](#)

<http://gateoverflow.in/70908>

Identify the regular expression which represents the language containing all strings of a's and b's where each string contains at least two b's

Identify the regular expression which represents the language containing all strings of a's and b's where each string contains at least two b's.

- (A) $(a+b)^*ba^*b$
- (B) $(a+b)^*ba^*ba$
- (C) $(a+b)^*ba^*b(a+b)^*$
- (D) None of these

theory-of-computation regular-expressions gateforum-test-series

[Answer](#)

5.12.25 Regular Expressions: ISI2013-CS-4b [top](#)

<http://gateoverflow.in/47638>

Write a regular expression for all strings of 0's and 1's in which the total number of 0's to the right of each 1 is even. Justify your answer.

descriptive isi2013 theory-of-computation regular-expressions

[Answer](#)

5.12.26 Regular Expressions: TIFR2010-B-34 [top](#)

<http://gateoverflow.in/19047>

Let r, s, t be regular expressions. Which of the following identities is correct?

- a. $(r + s)^* = r^*s^*$
- b. $r(s + t) = rs + t$
- c. $(r + s)^* = r^* + s^*$
- d. $(rs + r)^*r = r(sr + r)^*$
- e. $(r^*s)^* = (rs)^*$

tifr2010 theory-of-computation regular-expressions

[Answer](#)

5.12.27 Regular Expressions: TIFR2015-B-7 [top](#)

<http://gateoverflow.in/29861>

Let a, b, c be regular expressions. Which of the following identities is correct?

- a. $(a + b)^* = a^*b^*$
- b. $a(b + c) = ab + c$
- c. $(a + b)^* = a^* + b^*$
- d. $(ab + a)^*a = a(ba + a)^*$
- e. None of the above.

tifr2015 theory-of-computation regular-expressions

[Answer](#)

5.12.28 Regular Expressions: TIFR2017-B-9 [top](#)

<http://gateoverflow.in/95705>

Which of the following regular expressions correctly accepts the set of all 0/1-strings with an even (possibly zero) number of 1s?

- A. $(10^*10^*)^*$
- B. $(0^*10^*1)^*$
- C. $0^*1(10^*1)^*10^*$
- D. $0^*1(0^*10^*10^*)^*10^*$

E. $(0^*10^*1)^*0^*$

tifr2017 theory-of-computation regular-expressions

Answer

Answers: Regular Expressions

5.12.1 Regular Expressions: GATE 1992-2 (xii) Which of the following regular expression identities are true? [top](#)

<http://gateoverflow.in/83856>

B is only correct

why not A?

A's RHS can accept null strings, LHS can't.
if $r=ab$ then $r^*=abababa\dots$
 $r^*)=ab(*)=ab$

2 votes

-- Amitabh Tiwari (2.3k points)

5.12.2 Regular Expressions: GATE 2016-1-18 [top](#)

<http://gateoverflow.in/39647>



Selected Answer

Set of all binary strings having two consecutive 0s and two consecutive 1s

Anything 00 Anything 11 Anything + Anything 11 Anything 00 Anything

$(0+1)^*00(0+1)^*11(0+1)^* + (0+1)^*11(0+1)^*00(0+1)^*$

And it is same after taking common.

$(0+1)^*(00(0+1)^*11 + 11(0+1)^*00)(0+1)^*$

So **Option B** is Ans

Neither they said Both are immediate nor they give a predefined order, so it should be as above

30 votes

-- Praveen Saini (53k points)

5.12.3 Regular Expressions: GATE1987-10d [top](#)

<http://gateoverflow.in/82455>

$\epsilon^*0110\epsilon^*$

2 votes

-- billy (527 points)

5.12.4 Regular Expressions: GATE1991_03,xiii [top](#)

<http://gateoverflow.in/527>



Selected Answer

ans is A and C

to know the ans let us check all the options.

a) $L(s) \subseteq L(r)$: strings generated by s are any numbers of 1's followed by one 0, i.e., 10, 110, 1110, 1110, ... Strings generated by r are 1 followed by any combination of 0 or 1, i.e., 1, 10, 11, 1110, 101, 110, ... This shows that all the strings that can be generated by s , can also be generated by r it means $L(s) \subseteq L(r)$ is true.

$L(s) \subseteq L(t)$: here strings generated by t are any numbers of 1 (here 1^* means we have strings as $\epsilon, 1, 11, 111, \dots$) followed by only one 0, i.e., 0, 10, 110, 1110, ... So we can see that all the strings that are present in s can also be generated by t , hence $L(s) \subseteq L(t)$ which shows that option A is true.

b) $L(r) \subseteq L(s)$: this is false because string 1 which can be generated by r , cannot be generated by s .

c) Same as option A.

d) $L(t) \subseteq L(s)$: this is false because string 0 which can be generated by t , cannot be generated by s .

13 votes

-- neha pawar (4.4k points)

5.12.5 Regular Expressions: GATE1992_02,xvii [top](#)



Selected Answer

(a) is the answer

(b) RHS generates Σ^* while LHS can't generate strings where r comes after s like sr, srr etc. LHS \subset RHS.

(c) LHS generates Σ^* while RHS can't generate strings where r comes after an s . RHS \subset LHS.

(d) LHS contains all strings where after an s , no r comes. RHS contains all strings of either r or s but no combination of them. So, RHS \subset LHS.

15 votes

-- Arjun Suresh (287k points)

5.12.6 Regular Expressions: GATE1994_2.10 [top](#)



Selected Answer

$$L = 0^* 1^*$$

L contains all binary strings where a 1 is not followed by a 0.

13 votes

-- Manu Thakur (5.8k points)

5.12.7 Regular Expressions: GATE1995-1.9 , ISRO2017-13 [top](#)



Selected Answer

It is C

It has to be started by a letter followed by any number of letters and digits.

16 votes

-- Gate Keeda (18.8k points)

5.12.8 Regular Expressions: GATE1996_1.8 [top](#)



Selected Answer

C.

you can have any no. of 0's as well as null.

A is false because you cannot have single 0 in ii). same for option B. In D you are forced to have single 0 in iv) whereas not in iii).

14 votes

-- Gate Keeda (18.8k points)

5.12.9 Regular Expressions: GATE1997_6.4 [top](#)



Selected Answer

"A regular expression denoting a language (set of strings) means it should generate all string in L and not generate any string not in L "

- (a) - generates 100
 (b) doesn't generate 0 (start trying strings in lexicographic order- 0, 1, 00, 01, 10,...)
 (c) doesn't generate 1
 (d) is the answer

11 votes

-- Arjun Suresh (287k points)

5.12.10 Regular Expressions: GATE1998_1.12 [top](#)

<http://gateoverflow.in/1649>

Selected Answer

Only (a) and (b) can generate 1101.
 In (c) after 11, we can not have 01 and so 1101 cannot be generated.
 In (d) Every 11 followed by some 0. So it cannot generate 1101 or 11011.

18 votes

-- Arjun Suresh (287k points)

5.12.11 Regular Expressions: GATE1998_1.9 [top](#)

<http://gateoverflow.in/1648>

Selected Answer

(d) $A = B$. Both generates all strings over $\{0,1\}$ where a 0 is immediately followed by a 1.

9 votes

-- Arjun Suresh (287k points)

5.12.12 Regular Expressions: GATE1998_3b [top](#)

<http://gateoverflow.in/2841>

Selected Answer

The expression will be of the form $1^* 1^k (01^k)^*$.

7 votes

-- neha pawar (4.4k points)

5.12.13 Regular Expressions: GATE2000-1.4 [top](#)

<http://gateoverflow.in/627>

Selected Answer

(c) $S = T$. Both generates all strings over Σ .

13 votes

-- Arjun Suresh (287k points)

5.12.14 Regular Expressions: GATE2003-14 [top](#)

<http://gateoverflow.in/905>

Selected Answer

(A) is the answer. Both (A) and the given expression generates all strings over Σ .
 (B) doesn't generate 11
 (C) doesn't generate 11

15 votes

-- Arjun Suresh (287k points)

5.12.15 Regular Expressions: GATE2004-IT-7 [top](#)

<http://gateoverflow.in/3648>

Selected Answer

- A) $(a^* + b^* + c^*)^* = (\wedge + a+aa+\dots+b+bb+\dots+c+cc+\dots)^*$
 $= (a+b+c+ aa+\dots+bb+\dots+cc+\dots)^* = (a+b+c)^*$
[any combination of rest of aa ,bb,cc, etc already come in
 $(a+b+c)^*$]
- B) $(a^*b^*c^*)^* = (a^*+b^*+c^* + a*b^*+b*c^*+a*c^*+\dots)^* =$
 $(a+b+c+\dots)^* = (a+b+c)^*$
- C) $((ab)^* + c^*)^* = (ab+c+\wedge+abab+\dots)^* = (ab+c)^*$
- D) $(a^*b^* + c^*)^* = (a^*+b^*+c^*+\dots)^* = (a+b+c+\dots)^* =$
 $(a+b+c)^*$

13 votes

-- Praveen Saini (53k points)

5.12.16 Regular Expressions: GATE2005-IT-5 [top](#)



Selected Answer

(B). Infinite set (because of *) of finite strings. A string is defined as a **FINITE sequence** of characters and hence can never be infinite.

14 votes

-- Arjun Suresh (287k points)

5.12.17 Regular Expressions: GATE2006-IT-5 [top](#)



Selected Answer

Well A is answer

Say s1, s2 , s3 and t are states (in sequence)

s1 is start state

$s1 = \wedge + s1a + s1b = \wedge + s1(a+b) = (a+b)^*$ [using arden's theorem $R = Q+RP$, then $R = QP^*$][\wedge bcoz of start state]

$s2 = s1a = (a+b)^*a$

$s3 = s2a+s2b = s2(a+b) = (a+b)^*a(a+b)$

$t = s3b = (a+b)^*a(a+b)b$

t is final state so regular expression is $(a+b)^*a(a+b)b$

10 votes

-- Praveen Saini (53k points)

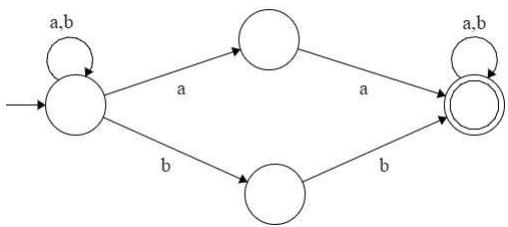
5.12.18 Regular Expressions: GATE2007-IT-73 [top](#)



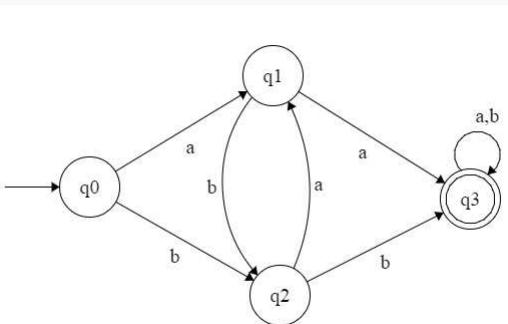
Selected Answer

$R = (a+b)^*(aa+bb)(a+b)^*$

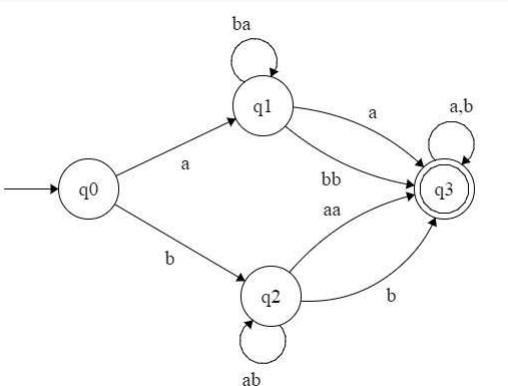
having NFA



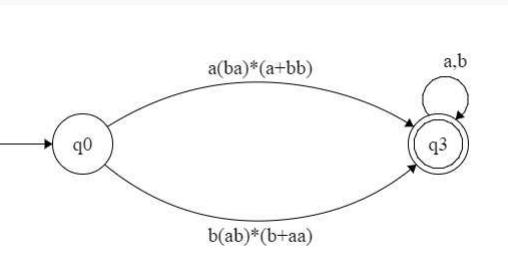
Equivalent DFA :



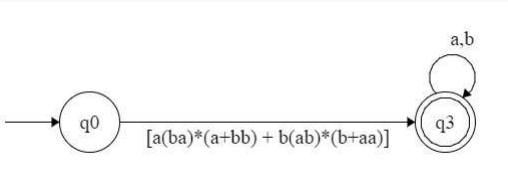
which is equivalent Transition graph [by removing transition from q1 to q2 and q2 to q1 but does not effect on language ..be careful]



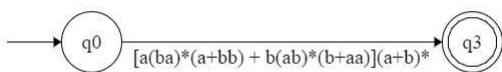
That is equivalent to



which is equivalent to



which is equivalent to



so equivalent regular expression is $[a(ba)^*(a+bb) + b(ab)^*(b+aa)](a+b)^*$

so option C is answer.

15 votes

-- Praveen Saini (53k points)

5.12.19 Regular Expressions: GATE2008-IT-5 [top](#)



Selected Answer

- A) with at least 2 consecutive 1's, any no of 0's and any no of 1's
- B) exactly two consecutive 1's
- C) exactly two 1's but need not be consecutive
- D) any no of 1's and 0's with at least two 1's

Hence C) is the correct option.

16 votes

-- Manu Thakur (5.8k points)

5.12.20 Regular Expressions: GATE2009-15 [top](#)



Selected Answer

(C) is the answer.

Counter example for other choices:

- (A) 1010 is accepted which doesn't contain 00
- (B) 000 is accepted
- (D) 01 is not accepted

12 votes

-- Arjun Suresh (287k points)

5.12.21 Regular Expressions: GATE2010-39 [top](#)



Selected Answer

- (A) - If the string contains a 1, it must end in a 1 hence cannot generate all bit strings with even number of 1's (eg, 1010)
- (B) - is the answer
- (C) - between the second and third 1's a 0 is not allowed (eg, 011011)
- (D) - 00 is not allowed, zero is an even number.

24 votes

-- Arjun Suresh (287k points)

5.12.22 Regular Expressions: GATE2014-1-36 [top](#)



Selected Answer

(B) is the answer. (II) doesn't generate 11011 which is accepted by the given DFA

24 votes

-- Arjun Suresh (287k points)

5.12.23 Regular Expressions: GATE2014-3-15 [top](#)

<http://gateoverflow.in/2049>



Selected Answer

$$R = a^*b^*(ba)^*a^*$$

for finding shortest string that is not in language it is better to look strings of length 0, then of length 1 and so on

length0 $\{\epsilon\}$ is in L

length1 $\{a, b\}$ all belong to L

length2 $\{aa, ab, ba, bb\}$ all belong to L

length 3 $\{aaa, aab, aba, abb, baa, bab, bba, bbb\}$ ***bab does not belong to L***

26 votes

-- Praveen Saini (5.3k points)

5.12.24 Regular Expressions: Gateforum test series [top](#)



Selected Answer

ans is C

Regular expression which represents the language containing all strings of a's and b's where each string contains at least two b's

$$\text{is } (a+b)^* b \ a^* b \ (a+b)^*$$

option A does not accept bba

option B does not accept bbb

option C accept all

3 votes

-- Shubham Pandey (5.9k points)

5.12.25 Regular Expressions: ISI2013-CS-4b [top](#)



Selected Answer

Here Total no. of o's to the right of each 1's should be even. So, to the left side of every 1's no. of o's can be anything.

$$L = \{\epsilon, 0, 00, 000, 01, 011, 0111, 1, 11, 111, 1111, 100, 10000, 100100, 10010000, \dots\}$$

Hence,

$$R.E. = o^*(1+oo)^* = o^*(1(oo)^*)^* = o^*(1+(oo)^*)^* = o^*(1^*+(oo)^*)^* = o^*(1^*+(oo)^*)^*$$

5 votes

-- Leen Sharma (31.4k points)

5.12.26 Regular Expressions: TIFR2010-B-34 [top](#)



Selected Answer

- a. $(r + s)^* = r^*s^*$ LHS can generate 'sr' but RHS not
- b. $r(s + t) = rs + t$ LHS can generate 'rt' but RHS not
- c. $(r + s)^* = r^* + s^*$ LHS can generate 's' but RHS not
- d. $(rs + r)^* = r(sr + r)^*$ They are equivalent
- e. $(r^*s)^* = (rs)^*$ LHS can generate 'rrs' but RHS not

So option D is correct answer.

11 votes

-- Umang Raman (14.5k points)

5.12.27 Regular Expressions: TIFR2015-B-7 [top](#)

<http://gateoverflow.in/29861>



Selected Answer

a)
 $(a+b)^*$ = { any strings of over {a,b} }

a^*b^* = { any number of a's followed by any number of b's }

False, as strings, ba , aba, bab, etc are not present in
 a^*b^*

b)
 $a(b+c) = \{ab, ac\}$

$ab + c = \{ab, c\}$

False

c)
 $(a+b)^*$ = { any strings of over {a,b} }
 $a^* + b^*$ = { any numbers of a's or any numbers of b's }

False , as strings , ab, ba, aba, bab etc are not present in
 $a^* + b^*$

d)
 $(ab+a)^*a = a(ba+a)^*$, **True**

$p(pq)^* = p\{\epsilon, qp, qpqp, qpqpqp, \dots\} = \{p, pqp, pqqp, pqqpqp, \dots\} = \{\epsilon, pq, pqpa, pqpqpq, \dots\}p = (pq)^*p$

$(ab+a)^*a = (a(b+\epsilon))^*a = a((b+\epsilon)a)^* = a(ba+a)^*$

4 14 votes

-- Praveen Saini (53k points)

5.12.28 Regular Expressions: TIFR2017-B-9 [top](#)

<http://gateoverflow.in/95705>



Selected Answer

As, mentioned in the question, the regular expression must accept all strings of 0 and 1 but with even no of 1s (including no 1s). Hence, 00 must be in the language. Option a, b, c, d do not accept 00. Hence, option e is correct.

4 4 votes

-- tarun_svbk (1k points)

5.13

Regular Grammar(3) [top](#)

5.13.1 Regular Grammar: GATE1990-15a [top](#)

<http://gateoverflow.in/86865>

Is the language generated by the grammar G regular? If so, give a regular expression for it, else prove otherwise

G: $S \rightarrow aB$

$B \rightarrow bC$

$C \rightarrow xB$

$C \rightarrow c$

[gate1990](#) [descriptive](#) [theory-of-computation](#) [regular-languages](#) [regular-grammar](#) [grammar](#)

[Answer](#)

5.13.2 Regular Grammar: GATE2006-IT-29 [top](#)

<http://gateoverflow.in/3568>

Consider the regular grammar below

$S \rightarrow bS \mid aA \mid \epsilon$

$A \rightarrow aS \mid bA$

The Myhill-Nerode equivalence classes for the language generated by the grammar are

- A. $\{w \in (a+b)^* \mid \#a(w) \text{ is even}\}$ and $\{w \in (a+b)^* \mid \#a(w) \text{ is odd}\}$
- B. $\{w \in (a+b)^* \mid \#a(w) \text{ is even}\}$ and $\{w \in (a+b)^* \mid \#b(w) \text{ is odd}\}$
- C. $\{w \in (a+b)^* \mid \#a(w) = \#b(w)\}$ and $\{w \in (a+b)^* \mid \#a(w) \neq \#b(w)\}$
- D. $\{\epsilon\}$, $\{wa \mid w \in (a+b)^*\}$ and $\{wb \mid w \in (a+b)^*\}$

gate2006-it theory-of-computation normal regular-grammar

Answer

5.13.3 Regular Grammar: GATE2015-2_35 [top](#)

<http://gateoverflow.in/8159>

Consider the alphabet $\Sigma = \{0,1\}$, the null/empty string λ and the set of strings X_0, X_1 , and X_2 generated by the corresponding non-terminals of a regular grammar. X_0, X_1 , and X_2 are related as follows.

$$X_0 = 1X_1$$

$$X_1 = 0X_1 + 1X_2$$

$$X_2 = 0X_1 + \{\lambda\}$$

Which one of the following choices precisely represents the strings in X_0 ?

- A. $10(0^*+(10)^*)1$
- B. $10(0^*+(10)^*)^*1$
- C. $1(0+10)^*1$
- D. $10(0+10)^*1 + 110(0+10)^*1$

gate2015-2 theory-of-computation regular-grammar normal

Answer

Answers: Regular Grammar

5.13.1 Regular Grammar: GATE1990-15a [top](#)

<http://gateoverflow.in/86864>



First of all this grammar is right linear..And we know :

Two special types of linear grammars are the following:

- the **left-linear** or **left regular grammars**, in which **all nonterminals** in right hand sides are **at the left ends**;
- the **right-linear** or right regular grammars, in which **all nonterminals** in right hand sides are **at the right ends**.

Hence the given grammar is regular and hence the language generated by regular grammar will also be regular..Alternatively we can also write a regular expression for it ..Let us see how to do it :

Given grammar

$$G: S \rightarrow aB$$

$$B \rightarrow bC$$

$$C \rightarrow xB \mid c$$

So we can reverse substitution from C onwards to S to see what S generates..

Substituting the yield of C in B , we have :

$$B \rightarrow bxB \mid bc$$

which gives $B = (bx)^* bc$

Now substituting B in S we have :

$$S \rightarrow aB$$

$$S = a(bx)^*bc$$

Hence the corresponding regular expression is : $a(bx)^*bc$

8 votes

-- HABIB MOHAMMAD KHAN (66.6k points)

5.13.2 Regular Grammar: GATE2006-IT-29 [top](#)

<http://gateoverflow.in/3568>

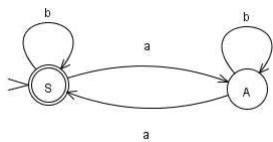


Selected Answer

Option A is correct.

The given grammar generates all string over the alphabet $\{a, b\}$ which have an even number of a 's.

The given right-linear grammar can be converted to the following DFA.



10 votes

-- Pragy Agarwal (19.3k points)

5.13.3 Regular Grammar: GATE2015-2_35 [top](#)

<http://gateoverflow.in/8159>



Selected Answer

Here we have little different version of Arden's Theorem

if we have $R = PR + Q$ then it has a solution $R = P^*Q$

Proof :

$$R = PR + Q$$

$$= P(PR+Q) + Q \quad (\text{by putting } R = PR+Q)$$

$$= PPR + PQ + Q$$

$$= PPP(R+Q) + PQ + Q \quad (\text{by putting } R = PR+Q)$$

$$= PPPR + PPQ + PQ + Q$$

and so on , we get $R = \{ \dots \dots \dots + PPPPQ + PPPQ + PPQ + PQ + Q \}$

$$= \{ \dots \dots \dots + PPPP + PPP + PP + P + \} Q = P^*Q$$

or Another way

$$R = PR + Q$$

$$= P(P^*Q) + Q \quad (\text{by putting } R = P^*Q)$$

$$= (PP^* + \) Q = P^*Q$$

So Equation is Proved .

Now for the Above Question

$$X_1 = 0X_1 + 1X_2 \quad (\text{Equation 2})$$

$$= 0X_1 + 1(0X_1 + \) \quad (\text{Put the value of } X_2 \text{ from Equation 3})$$

$=0X_1 + 10 X_1 + 1 = (0+10)X_1 + 1$
 $X_1 = (0+10)*1 \text{ (Apply if } R = PR + Q \text{ then } R = P*Q)$
 $X_0 = 1 X_1 \text{ (Equation 1)}$
 $X_0 = 1(0+10)*1 \text{ (Put the value of } X_1 \text{ we calculated).}$
So $1(0+10)*1$ option C is correct.

17 votes

-- Praveen Saini (53k points)

option c is correct because look at the smallest string generated by the grammar is 11 which is not accepted by any option.

17 votes

-- Anoop Sonkar (4.8k points)

5.14

Regular Languages(36) [top](#)

5.14.1 Regular Languages: CMI2012-A-01 [top](#)

<http://gateoverflow.in/46529>

Let $L \subseteq \{0,1\}^*$. Which of the following is true?

- A. If L is regular, all subsets of L are regular.
- B. If all proper subsets of L are regular, then L is regular.
- C. If all finite subsets of L are regular, then L is regular.
- D. If a proper subset of L is not regular, then L is not regular.

[cmi2012](#) [theory-of-computation](#) [regular-languages](#)

Answer

5.14.2 Regular Languages: CMI2015-A-09 [top](#)

<http://gateoverflow.in/47046>

Let L_1 and L_2 be languages over an alphabet Σ such that $L_1 \subseteq L_2$. Which of the following is true:

- A. If L_2 is regular, then L_1 must also be regular.
- B. If L_1 is regular, then L_2 must also be regular.
- C. Either both L_1 and L_2 are regular, or both are not regular.
- D. None of the above.

[cmi2015](#) [theory-of-computation](#) [regular-languages](#)

Answer

5.14.3 Regular Languages: GATE 2016-2-17 [top](#)

<http://gateoverflow.in/39542>

Language L_1 is defined by the grammar: $S_1 \rightarrow aS_1b \mid \epsilon$

Language L_2 is defined by the grammar: $S_2 \rightarrow abS_2 \mid \epsilon$

Consider the following statements:

- P: L_1 is regular
- Q: L_2 is regular

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

- A. Both P and Q are true.
- B. P is true and Q is false.
- C. P is false and Q is true.
- D. Both P and Q are false.

[gate2016-2](#) [theory-of-computation](#) [normal](#) [regular-languages](#)

Answer**5.14.4 Regular Languages: GATE1987-2h** [top](#)<http://gateoverflow.in/80590>

State whether the following statements are TRUE or FALSE:

Regularity is preserved under the operation of string reversal.

[gate1987](#) [regular-languages](#) [theory-of-computation](#)

Answer**5.14.5 Regular Languages: GATE1987-2i** [top](#)<http://gateoverflow.in/80590>

State whether the following statements are TRUE or FALSE:

All subsets of regular sets are regular.

[gate1987](#) [theory-of-computation](#) [regular-languages](#)

Answer**5.14.6 Regular Languages: GATE1990-3-viii** [top](#)<http://gateoverflow.in/84837>

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

Let R_1 and R_2 be regular sets defined over the alphabet Σ . Then:

- A. $R_1 \cap R_2$ is not regular.
- B. $R_1 \cup R_2$ is regular.
- C. $\Sigma^* - R_1$ is regular.
- D. R_1^* is not regular.

[gate1990](#) [normal](#) [theory-of-computation](#) [regular-languages](#)

Answer**5.14.7 Regular Languages: GATE1991-03,xiv** [top](#)<http://gateoverflow.in/528>

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

Which of the following is the strongest correct statement about a finite language over some finite alphabet Σ ?

- (a). It could be undecidable
- (b). It is Turing-machine recognizable
- (c). It is a context sensitive language.
- (d). It is a regular language.
- (e). None of the above,

[gate1991](#) [theory-of-computation](#) [easy](#) [regular-languages](#)

Answer**5.14.8 Regular Languages: GATE1995_2.24** [top](#)<http://gateoverflow.in/2637>

Let $\Sigma = \{0,1\}$, $L = \Sigma^*$ and $R = \{0^n 1^n \mid n > 0\}$ then the languages $L \cup R$ and R are respectively

- (A) regular, regular
- (B) not regular, regular
- (C) regular, not regular
- (D) not regular, not regular

[gate1995](#) [theory-of-computation](#) [easy](#) [regular-languages](#)

Answer

5.14.9 Regular Languages: GATE1996_1.10 [top](#)

<http://gateoverflow.in/2714>

Let $L \subseteq \Sigma^*$ where $\Sigma = \{a, b\}$. Which of the following is true?

- (a) $L = \{x \mid x \text{ has an equal number of } a's \text{ and } b's\}$ is regular
- (b) $L = \{a^n b^n \mid n \geq 1\}$ is regular
- (c) $L = \{x \mid x \text{ has more number of } a's \text{ than } b's\}$ is regular
- (d) $L = \{a^m b^n \mid m \geq 1, n \geq 1\}$ is regular

[gate1996](#) [theory-of-computation](#) [normal](#) [regular-languages](#)

[Answer](#)

5.14.10 Regular Languages: GATE1998_2.6 [top](#)

<http://gateoverflow.in/1678>

Which of the following statements is false?

- (a) Every finite subset of a non-regular set is regular
- (b) Every subset of a regular set is regular
- (c) Every finite subset of a regular set is regular
- (d) The intersection of two regular sets is regular

[gate1998](#) [theory-of-computation](#) [easy](#) [regular-languages](#)

[Answer](#)

5.14.11 Regular Languages: GATE1999_6 [top](#)

<http://gateoverflow.in/1505>

- Given that A is regular and $(A \cup B)$ is regular, does it follow that B is necessarily regular? Justify your answer.
- Given two finite automata $M1, M2$, outline an algorithm to decide if $L(M1) \subset L(M2)$. (note: strict subset)

[gate1999](#) [theory-of-computation](#) [normal](#) [regular-languages](#)

[Answer](#)

5.14.12 Regular Languages: GATE2000-2.8 [top](#)

<http://gateoverflow.in/655>

What can be said about a regular language L over $\{a\}$ whose minimal finite state automaton has two states?

- A. L must be $\{a^n \mid n \text{ is odd}\}$
- B. L must be $\{a^n \mid n \text{ is even}\}$
- C. L must be $\{a^n \mid n \geq 0\}$
- D. Either L must be $\{a^n \mid n \text{ is odd}\}$, or L must be $\{a^n \mid n \text{ is even}\}$

[gate2000](#) [theory-of-computation](#) [easy](#) [regular-languages](#)

[Answer](#)

5.14.13 Regular Languages: GATE2001-1.4 [top](#)

<http://gateoverflow.in/697>

Consider the following two statements:

- S1 : $\{0^{2n} \mid n \geq 1\}$ is a regular language
- S2 : $\{0^m 1^n 0^{m+n} \mid m \geq 1 \text{ and } n \geq 1\}$ is a regular language

Which of the following statement is correct?

- A. Only S1 is correct
- B. Only S2 is correct
- C. Both S1 and S2 are correct
- D. None of S1 and S2 is correct

[gate2001](#) [theory-of-computation](#) [easy](#) [regular-languages](#)

[Answer](#)

5.14.14 Regular Languages: GATE2001-2.6 [top](#)

<http://gateoverflow.in/724>

Consider the following languages:

- $L_1 = \{ww \mid w \in \{a,b\}^*\}$
- $L_2 = \{ww^R \mid w \in \{a,b\}^*, w^R \text{ is the reverse of } w\}$
- $L_3 = \{0^{2i} \mid i \text{ is an integer}\}$
- $L_4 = \{0^{i^2} \mid i \text{ is an integer}\}$

Which of the languages are regular?

- A. Only L1 and L2
- B. Only L2, L3 and L4
- C. Only L3 and L4
- D. Only L3

[gate2001](#) [theory-of-computation](#) [normal](#) [regular-languages](#)

[Answer](#)

5.14.15 Regular Languages: GATE2006-29 [top](#)

<http://gateoverflow.in/992>

If s is a string over $(0+1)^*$ then let $n_0(s)$ denote the number of 0's in s and $n_1(s)$ the number of 1's in s. Which one of the following languages is not regular?

- A. $L = \{s \in (0+1)^* \mid n_0(s) \text{ is a 3-digit prime}\}$
- B. $L = \{s \in (0+1)^* \mid \text{for every prefix } s' \text{ of } s, |n_0(s') - n_1(s')| \leq 2\}$
- C. $L = \{s \in (0+1)^* \mid n_0(s) - n_1(s) \leq 4\}$
- D. $L = \{s \in (0+1)^* \mid n_0(s) \bmod 7 = n_1(s) \bmod 5 = 0\}$

[gate2006](#) [theory-of-computation](#) [normal](#) [regular-languages](#)

[Answer](#)

5.14.16 Regular Languages: GATE2006-IT-30 [top](#)

<http://gateoverflow.in/3569>

Which of the following statements about regular languages is NOT true ?

- A. Every language has a regular superset
- B. Every language has a regular subset
- C. Every subset of a regular language is regular
- D. Every subset of a finite language is regular

[gate2006-it](#) [theory-of-computation](#) [easy](#) [regular-languages](#)

[Answer](#)

5.14.17 Regular Languages: GATE2006-IT-80 [top](#)

<http://gateoverflow.in/3624>

Let L be a regular language. Consider the constructions on L below:

- I. repeat (L) = {ww | w ∈ L}
- II. prefix (L) = {u | ∃v : uv ∈ L}
- III. suffix (L) = {v | ∃u : uv ∈ L}
- IV. half (L) = {u | ∃v : |v| = |u| and uv ∈ L}

Which of the constructions could lead to a non-regular language?

- A. Both I and IV
- B. Only I
- C. Only IV
- D. Both II and III

[gate2006-it](#) [theory-of-computation](#) [normal](#) [regular-languages](#)

[Answer](#)

5.14.18 Regular Languages: GATE2006-IT-81 [top](#)

<http://gateoverflow.in/3637>

Let L be a regular language. Consider the constructions on L below:

- I. repeat (L) = $\{ww \mid w \in L\}$
- II. prefix (L) = $\{u \mid \exists v : uv \in L\}$
- III. suffix (L) = $\{v \mid \exists u : uv \in L\}$
- IV. half (L) = $\{u \mid \exists v : |v| = |u| \text{ and } uv \in L\}$

Which of the constructions could lead to a non-regular language?

- a. Both I and IV
- b. Only I
- c. Only IV
- d. Both II and III

Which choice of L is best suited to support your answer above?

- A. $(a + b)^*$
- B. $\{\epsilon, a, ab, bab\}$
- C. $(ab)^*$
- D. $\{a^n b^n \mid n \geq 0\}$

[gate2006-it](#) [theory-of-computation](#) [normal](#) [regular-languages](#)

[Answer](#)

5.14.19 Regular Languages: GATE2007-31 [top](#)

<http://gateoverflow.in/1229>

Which of the following languages is regular?

- A. $\{ww^R \mid w \in \{0,1\}^+\}$
- B. $\{ww^Rx \mid x, w \in \{0,1\}^+\}$
- C. $\{wxw^R \mid x, w \in \{0,1\}^+\}$
- D. $\{xuwu^R \mid x, w \in \{0,1\}^+\}$

[gate2007](#) [theory-of-computation](#) [normal](#) [regular-languages](#)

[Answer](#)

5.14.20 Regular Languages: GATE2007-7 [top](#)

<http://gateoverflow.in/1205>

Which of the following is TRUE?

- A. Every subset of a regular set is regular
- B. Every finite subset of a non-regular set is regular
- C. The union of two non-regular sets is not regular
- D. Infinite union of finite sets is regular

[gate2007](#) [theory-of-computation](#) [easy](#) [regular-languages](#)

[Answer](#)

5.14.21 Regular Languages: GATE2008-IT-35 [top](#)

<http://gateoverflow.in/3345>

Which of the following languages is (are) non-regular?

- $L_1 = \{0^m 1^n \mid 0 \leq m \leq n \leq 10000\}$
 - $L_2 = \{w \mid w \text{ reads the same forward and backward}\}$
 - $L_3 = \{w \in \{0, 1\}^* \mid w \text{ contains an even number of 0's and an even number of 1's}\}$
- A. L_2 and L_3 only
 B. L_1 and L_2 only
 C. L_3 only
 D. L_2 only

[gate2008-it](#) [theory-of-computation](#) [normal](#) [regular-languages](#)

[Answer](#)

5.14.22 Regular Languages: GATE2008_53 [top](#)

<http://gateoverflow.in/478>

Which of the following are regular sets?

- I. $\{a^n b^{2m} \mid n \geq 0, m \geq 0\}$
 - II. $\{a^n b^m \mid n = 2m\}$
 - III. $\{a^n b^m \mid n \neq m\}$
 - IV. $\{xcy \mid x, y \in \{a, b\}^*\}$
- A. I and IV only
 B. I and III only
 C. I only
 D. IV only

[gate2008](#) [theory-of-computation](#) [normal](#) [regular-languages](#)

[Answer](#)

5.14.23 Regular Languages: GATE2011-24 [top](#)

<http://gateoverflow.in/3429>

Let P be a regular language and Q be a context-free language such that $Q \subseteq P$. (For example, let P be the language represented by the regular expression $p^* q^*$ and Q be $\{p^n q^n \mid n \in N\}$). Then which of the following is **ALWAYS** regular?

- (A) $P \cap Q$
 (B) $P - Q$
 (C) $\Sigma^* - P$
 (D) $\Sigma^* - Q$

[gate2011](#) [theory-of-computation](#) [easy](#) [regular-languages](#)

[Answer](#)

5.14.24 Regular Languages: GATE2012-25 [top](#)

<http://gateoverflow.in/1609>

Given the language $L = \{ab, aa, baa\}$, which of the following strings are in L^* ?

1. abaabaaaabaa
2. aaaaabaaaa
3. baaaaabaaaaab
4. baaaaabaa

- (A) 1, 2 and 3
 (B) 2, 3 and 4
 (C) 1, 2 and 4
 (D) 1, 3 and 4

gate2012 | theory-of-computation | easy | regular-languages

[Answer](#)

5.14.25 Regular Languages: GATE2013-8 [top](#)

<http://gateoverflow.in/1411>

Consider the languages $L_1 = \phi$ and $L_2 = \{a\}$. Which one of the following represents $L_1 L_2^* \cup L_1^*$?

- (A) $\{\epsilon\}$
 (B) ϕ
 (C) a^*
 (D) $\{\epsilon, a\}$

gate2013 | theory-of-computation | normal | regular-languages

[Answer](#)

5.14.26 Regular Languages: GATE2014-1-15 [top](#)

<http://gateoverflow.in/1781>

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

- A. The language $L = \{a^n b^n \mid n \geq 0\}$ is regular.
 B. The language $L = \{a^n \mid n \text{ is prime}\}$ is regular.
 C. The language $L = \{w \mid w \text{ has } 3k+1 \text{ } b's \text{ for some } k \in N \text{ with } \Sigma = \{a, b\}\}$ is regular.
 D. The language $L = \{ww \mid w \in \Sigma^* \text{ with } \Sigma = \{0, 1\}\}$ is regular.

gate2014-1 | theory-of-computation | regular-languages | normal

[Answer](#)

5.14.27 Regular Languages: GATE2014-2-15 [top](#)

<http://gateoverflow.in/1781>

If $L_1 = \{a^n \mid n \geq 0\}$ and $L_2 = \{b^n \mid n \geq 0\}$, consider

- I. $L_1 \cdot L_2$ is a regular language
 II. $L_1 \cdot L_2 = \{a^n b^n \mid n \geq 0\}$

Which one of the following is CORRECT?

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

gate2014-2 | theory-of-computation | normal | regular-languages

[Answer](#)

5.14.28 Regular Languages: GATE2014-2-36 [top](#)

<http://gateoverflow.in/1995>

$L_1 = \{w \in \{0, 1\}^* \mid w \text{ has at least as many occurrences of } (110)'s \text{ as } (011)'s\}$.
 $L_2 = \{w \in \{0, 1\}^* \mid w \text{ has at least as many occurrences of } (000)'s \text{ as } (111)'s\}$. Which one of the following is TRUE?

- A. L_1 is regular but not L_2
 B. L_2 is regular but not L_1
 C. Both L_1 and L_2 are regular
 D. Neither L_1 nor L_2 are regular

gate2014-2 | theory-of-computation | normal | regular-languages

[Answer](#)

5.14.29 Regular Languages: GATE2015-2_51 [top](#)

<http://gateoverflow.in/8254>

Which of the following is/are regular languages?

$L_1 : \{wxw^R \mid w, x \in \{a, b\}^* \text{ and } |w|, |x| > 0\}$, w^R is the reverse of string w

$L_2 : \{a^n b^m \mid m \neq n \text{ and } m, n \geq 0\}$

$L_3 : \{a^p b^q c^r \mid p, q, r \geq 0\}$

- A. L_1 and L_3 only
- B. L_2 only
- C. L_2 and L_3 only
- D. L_3 only

[gate2015-2](#) [theory-of-computation](#) [normal](#) [regular-languages](#)

[Answer](#)

5.14.30 Regular Languages: ISI2014-CS-4b [top](#)

<http://gateoverflow.in/47445>

Consider the following statement:

For all languages $L \subseteq \{0, 1\}^*$, if L^* is regular then L is regular.

Is the above statement true? Justify your answer.

[descriptive](#) [isi2014](#) [theory-of-computation](#) [regular-languages](#)

[Answer](#)

5.14.31 Regular Languages: ISI2015-CS-5a [top](#)

<http://gateoverflow.in/47327>

Construct two nonregular languages L_1 and L_2 such that $L_1 \cup L_2$ is regular.

Prove that the languages L_1 and L_2 constructed above are nonregular and $L_1 \cup L_2$ is regular.

[descriptive](#) [isi2015](#) [theory-of-computation](#) [regular-languages](#)

[Answer](#)

5.14.32 Regular Languages: TIFR2013-B-6 [top](#)

<http://gateoverflow.in/25667>

Let L and L' be languages over the alphabet Σ . The left quotient of L by L' is

$$L/L' = \{w \in \Sigma^* : wx \in L \text{ for some } x \in L'\}$$

Which of the following is true?

- a. If L/L' is regular then L' is regular.
- b. If L is regular then L/L' is regular.
- c. If L/L' is regular then L is regular.
- d. L/L' is a subset of L .
- e. If L/L' and L' are regular, then L is regular.

[tifr2013](#) [theory-of-computation](#) [regular-languages](#)

[Answer](#)

5.14.33 Regular Languages: TIFR2013-B-8 [top](#)

<http://gateoverflow.in/25670>

Which one of the following languages over the alphabet $0, 1$ is regular?

- a. The language of balanced parentheses where $0, 1$ are thought of as $(,)$ respectively.
- b. The language of palindromes, i.e. bit strings x that read the same from left to right as well as right to left.
- c. $L = \{0^{m^2} : 3 \leq m\}$

- d. The Kleene closure L^* , where L is the language in (c) above.
e. $\{0^m 1^n \mid 1 \leq m \leq n\}$

tifr2013 theory-of-computation regular-languages

[Answer](#)

5.14.34 Regular Languages: TIFR2014-B-12 [top](#)

<http://gateoverflow.in/27314>

Consider the following three statements:

- (i) Intersection of infinitely many regular languages must be regular.
- (ii) Every subset of a regular language is regular.
- (iii) If L is regular and M is not regular then $L \bullet M$ is necessarily not regular.

Which of the following gives the correct true/false evaluation of the above?

- a. true, false, true.
- b. false, false, true.
- c. true, false, true.
- d. false, false, false.
- e. true, true, true.

tifr2014 theory-of-computation regular-languages

[Answer](#)

5.14.35 Regular Languages: TIFR2015-B-10 [top](#)

<http://gateoverflow.in/30039>

Consider the languages

$$L_1 = \{a^m b^n c^p \mid (m = n \vee n = p) \wedge m + n + p \geq 10\}$$

$$L_2 = \{a^m b^n c^p \mid (m = n \vee n = p) \wedge m + n + p \leq 10\}$$

State which of the following is true?

- A. L_1 and L_2 are both regular.
- B. Neither L_1 nor L_2 is regular.
- C. L_1 is regular and L_2 is not regular.
- D. L_1 is not regular and L_2 is regular.
- E. Both L_1 and L_2 are infinite.

tifr2015 regular-languages

[Answer](#)

5.14.36 Regular Languages: TIFR2015-B-6 [top](#)

<http://gateoverflow.in/29860>

Let B consist of all binary strings beginning with a 1 whose value when converted to decimal is divisible by 7.

- A. B can be recognized by a deterministic finite state automaton.
- B. B can be recognized by a non-deterministic finite state automaton but not by a deterministic finite state automaton.
- C. B can be recognized by a deterministic push-down automaton but not by a non-deterministic finite state automaton.
- D. B can be recognized by a non-deterministic push-down automaton but not by a deterministic push-down automaton.
- E. B cannot be recognized by any push down automaton, deterministic or non-deterministic.

tifr2015 theory-of-computation regular-languages

[Answer](#)

Answers: Regular Languages

5.14.1 Regular Languages: CMI2012-A-01 [top](#)

<http://gateoverflow.in/46529>

Ans C) Every finite subset should always be regular

A set may be regular

It's all subset is not regular all the time

Say $(0+1)^*$ is regular but 0^n1^n is not regular

2 votes

-- srestha (55.2k points)

5.14.2 Regular Languages: CMI2015-A-09 [top](#)



Selected Answer

Ans should be option D.

Contradiction for A. Let $L_2 = \{a,b\}^*$... which is regular.

And $L_1 = a^n b^n$ which is CFL but not regular.

And here L_1 is subset of L_2 .

Contradiction for B. Let $L_1 = ab$,

which is regular. And $L_2 = a^n b^n$ which is CFL but not regular.

And here L_1 is subset of L_2 .

C- > False , (reason A and B).

4 votes

-- Dhananjay Kumar Sharma (24.9k points)

5.14.3 Regular Languages: GATE 2016-2-17 [top](#)



Selected Answer

Answer is C

$$S_1 \rightarrow aS_1b|\epsilon$$

$L_1 = \{a^n b^n \mid n \geq 0\}$ is CFL

$$S_2 \rightarrow abS_2|\epsilon$$

$L_2 = \{(ab)^n \mid n \geq 0\}$ is Regular having regular expression $(ab)^*$

16 votes

-- Praveen Saini (53k points)

5.14.4 Regular Languages: GATE1987-2h [top](#)



Selected Answer

true!

reverse all the edges and interchange final and initial states in DFA

1 votes

-- Motamarri Anusha (11.4k points)

5.14.5 Regular Languages: GATE1987-2i [top](#)



Selected Answer

false!

a^*b^* is regular but its subset a^nb^n is not regular

4 votes

-- Motamarri Anusha (11.4k points)

5.14.6 Regular Languages: GATE1990-3-viii [top](#)



Selected Answer

Regular Languages are closed under

1. Intersection
2. Union
3. Complement
4. Kleen-Closure

$\Sigma^* - R_1$ is the complement of R_1

B,C are true

4 votes

-- pC (20.7k points)

5.14.7 Regular Languages: GATE1991-03,xiv [top](#)



Selected Answer

(b), (c) and (d) are true. But the strongest answer would be (d) a regular language. It is trivial to say that a finite set of strings (finite language) can be accepted using a finite set of states. And regular language \subset context-free \subset context-sensitive \subset Turing recognizable, would imply that regular language is the strongest answer.

11 votes

-- gatecse (13.1k points)

5.14.8 Regular Languages: GATE1995_2.24 [top](#)



Selected Answer

Answer is (C). $L \cup R$ is nothing but L as R is a subset of L and hence regular. R is deterministic context-free but not regular as we require a stack to keep the count of 0's to match that of 1's.

9 votes

-- Arjun Suresh (287k points)

5.14.9 Regular Languages: GATE1996_1.10 [top](#)



Selected Answer

Only D. because n and m are independent and thus no memory element required.

a and b are same and are DCFL's.

c is $L = \{ a^n b^m \mid n > m \}$. which is not regular.

Correction:I think c should be that x has more a's than b's.

5 votes

-- Gate Keeda (18.8k points)

5.14.10 Regular Languages: GATE1998_2.6 [top](#)



Selected Answer

(b) is False. Any language is a subset of Σ^* which is a regular set. So, if we take any non-regular language, it is a subset of a regular language.

(a) and (c) are regular as any finite language is regular.

(d) is regular as regular set is closed under intersection.

11 votes

-- Arjun Suresh (287k points)

5.14.11 Regular Languages: GATE1999_6 [top](#)



Selected Answer

a) A is regular , A U B is regular , then B is not necessary regular

example :- A = (a+b)* B = $a^n b^n$ $n \geq 0$ A U B is $(a+b)^*$ while B is not regular.

b) we have two machine M 1 and M 2

draw a DFA using M1 and M2 where start state is, say, $p_0 q_0$ (where p_0 is start state in M1 and q_0 is start state in M2)

$\delta(p_0 q_0, 0) = \delta(p_0, 0) \cup \delta(q_0, 0)$

if $L(M1) \subseteq L(M2)$

Then final state of M1 will come together with final state of M2, while Final state of M2 can come alone.

i.e All inputs of M1 is also in machine M2 , and there may be different inputs in M2.

12 votes

-- Praveen Saini (53k points)

5.14.12 Regular Languages: GATE2000-2.8 [top](#)



Selected Answer

Ans 4) Either L must be $\{a^n \mid n \text{ is odd}\}$, or L must be $\{a^n \mid n \text{ is even}\}$

Because if we draw the minimal dfa for each of them, we will get two states each.
Whereas, $\{a^n \mid n \geq 0\}$ requires only one state.

12 votes

-- Keith Kr (6.2k points)

5.14.13 Regular Languages: GATE2001-1.4 [top](#)



Selected Answer

Only s1 is correct a dfa with 2 states where one of the states is both the initial and final state..

8 votes

-- Bhagirathi Nayak (13.1k points)

5.14.14 Regular Languages: GATE2001-2.6 [top](#)



Selected Answer

$L1 = \{ww \mid w \in \{a,b\}^*\}$ CSL

$L2 = \{ww^R \mid w \in \{a,b\}^*, w^R \text{ is the reverse of } w\}$ Palindrome so CFL

$L3 = \{0^{2i} \mid i \text{ is an integer}\}$ Linear Power and regular expression can be stated as $(00)^*$

$L4 = \{0^{i^2} \mid i \text{ is an integer}\}$ non linear power So CSL

Therefore answer is option D

7 votes

-- Umang Raman (14.5k points)

5.14.15 Regular Languages: GATE2006-29 [top](#)

<http://gateoverflow.in/992>

A. There are only finite 3 digit primes. And any finite set is regular

B. Here we need just 6 states to recognise L.

1. #0 - #1 = 0
2. #0 - #1 = 1
3. #0 - #1 = 2
4. #0 - #1 = -1
5. #0 - #1 = -2

If the difference goes above 2 or below -2, we go to a dead state. All other states are accepting. This transition to dead state is possible because of the words "for every prefix s' of s" in L and that is what makes this language regular.

C. L is not regular

```
#0 - #1 = 1  
#0 - #1 = 2  
#0 - #1 = 3  
#0 - #1 = 4  
#0 - #1 = 5  
.....  
#0 - #1 = 1000  
.....
```

All these form distinct equivalent classes under Myhill-Nerode theorem meaning from the strings in each of these sets, we can append a string which will take the new string to L, while the same string appended to string in any other set won't reach L.

For example, for 000000, we append 11, for 0000000, we append 111 etc. So, in short we need to maintain the count of 1's and 0's and the count here is not finite.

D. This is regular. We need a finite automata with $5 * 7 = 35$ states for maintaining the counts of 0's mod 7 and 1's mod 5 and there cannot be more than 35 possibilities for this. With each input symbol, the transition must be going to one among these.

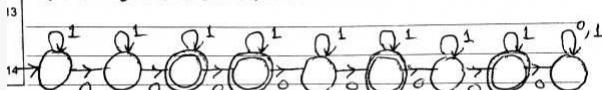
11 votes

-- Arjun Suresh (287k points)

Sorry for bad handwriting... :P

A. $n_0(S)$ is a 3 digit prime.
 It means no. of 0's are in range (set) {101, 103, ..., 997} which is finite.
 So, L will be regular.
 For simplicity lets consider $n_0(S)$ is a 1 digit prime.
 So, set will be {2, 3, 5, 7}.

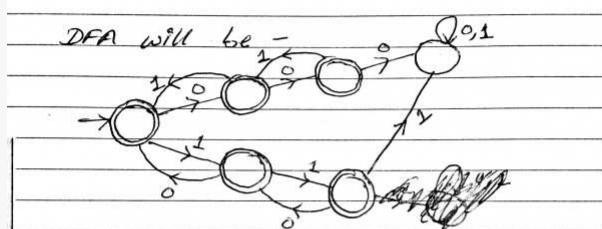
DFA will look like -



B. Prefix - For a string S, its prefix S' is a substring from the beginning of S.
 e.g. $S = 10110$
 $S' = 1, 10, 101, 1011, 10110$

$L = \{S \in (0+1)^*\mid \text{for every prefix } S'\text{ of } S, |n_0(S') - n_1(S')| \leq 2\}$.
 So, here we have to ensure that nowhere in our string the difference between 1's and 0's is greater than 2.

If anywhere the above restriction fails we will enter the dead state as we got a substring (prefix) that violates the restriction.



So, here once the difference between 0's and 1's crosses two we will be in dead state.

e.g. $S = 10110$ $S' = \{1, 10, 101, 1011, 10110\}$
 This string will be accepted as nowhere in S' the difference exceeds 2.

$S = 011110$ $S' = \{0, 01, 011, 0111, 01111\}$
 difference = 3.

This string is not accepted.

C. The difference between number of 0's and 1's should not exceed 4.

Here, once we exceed the limit there is possibility that we might cover up the difference in further string.

e.g. 011111000
 Here just by seeing upto last '1' we can't move to dead state.
 So, it needs counting. Hence, not regular.

D. This language can be stated as the language containing strings where no. of 0's are divisible by 7 and no. of 1's are divisible by 5.
 So, make DFA by cross product method.

It is regular.

14 votes

-- Rajendra Dangwal (635 points)

5.14.16 Regular Languages: GATE2006-IT-30 [top](#)

<http://gateoverflow.in/3569>



Selected Answer

Option C is not True.

A) Every language has a regular superset: True. Σ^* is such a superset.

B) Every language has a regular subset: True. \emptyset is such a subset.

C) Every subset of a regular language is regular: False. $a^n b^n \subset \Sigma^*$, but $a^n b^n$ is not Regular.

D) Every subset of a finite language is regular: True. Every subset of a finite set must be finite by definition. Every finite set is regular. Hence, every subset of a finite language is regular.

18 votes

-- Pragy Agarwal (19.3k points)

5.14.17 Regular Languages: GATE2006-IT-80 [top](#)



Selected Answer

correct answer is B. only I .

repeat (L) = $\{ww \mid w \in L\}$ is non regular language

half(L), suffix(L), and prefix(L) are regular languages

refer

http://gateoverflow.in/3637/gate2006-it_81

3 votes

-- Praveen Saini (53k points)

5.14.18 Regular Languages: GATE2006-IT-81 [top](#)



Selected Answer

i) repeat(L) is non regular

<http://www.cs.odu.edu/~toida/nerzic/390teched/regular/reg-lang/non-regularity.html> [example 2]

ii) prefix(L) is regular

<http://www.public.asu.edu/~ccolbou/src/355hw2sols09.pdf> [2(a)]

iii) suffix(L) is regular

<http://www.public.asu.edu/~ccolbou/src/355hw2sols09.pdf> [2(b)]

(iv) Half(L) is regular

<https://www.complang.tuwien.ac.at/lkovacs/ATCSHW/hw4-sol.pdf> [4.2(a)]

so in first part of question . option B is correct only i.

[Need someone to explain these language with more clear explanation @Arjun . that can be understandable by an average student]

for second part of question

A is answer.

Explanation is in comment given below by Arjun (for option A,C and D)

note : for option B $L = \{\epsilon, a, ab, bab\}$, Repeat(L) = $\{\epsilon, aa, abab, babbab\}$ is regular

7 votes

-- Praveen Saini (53k points)

5.14.19 Regular Languages: GATE2007-31 [top](#)

<http://gateoverflow.in/1229>



Selected Answer

- A. CFL
- B. CFL
- C. Regular, language is string starting and ending with the same symbol and having length at least 3. e.g. 0x0 or 1x1
- D. CFL

http://gatcse.in/wiki/Identify_the_class_of_the_language

12 votes

-- Vikrant Singh (13.4k points)

5.14.20 Regular Languages: GATE2007-7 [top](#)

<http://gateoverflow.in/1205>



Selected Answer

(B) Every finite subset of a non-regular set is regular.

Any finite set is always regular.

Σ^* being regular any non regular language is a subset of this, and hence (A) is false.

If we take a CFL which is not regular, and takes union with its complement (complement of a CFL which is not regular won't be regular as regular is closed under complement), we get Σ^* which is regular. So, (C) is false.

Regular set is not closed under infinite union. Example:

Let $L_i = \{0^i 1^i\}$, $i \in N$

Now, if we take infinite union over all i, we get

$L = \{0^i 1^i \mid i \in N\}$, which is not regular.

So, D is false.

12 votes

-- Omesh Pandita (2.7k points)

5.14.21 Regular Languages: GATE2008-IT-35 [top](#)

<http://gateoverflow.in/3345>



Selected Answer

L_1 is regular.. since 10000 is finite.. so finite states are required..

L_3 is also regular.. we can make DFA for L_3 .. states will represent mod 2 for 0 and mod 2 for 1, which is finite

L_2 is non. regular.. it is CFG $S \rightarrow aSa \mid \dots \mid zSz \mid \epsilon \mid [a-z]$

so option (d)

12 votes

-- Vicky Bajoria (4.7k points)

5.14.22 Regular Languages: GATE2008_53 [top](#)

<http://gateoverflow.in/476>



Selected Answer

Answer is A.

Since in option 2 and 3, n is dependent on m, therefore a comparison has to be done to evaluate those and hence are not regular.

I and IV are clearly regular sets.

11 votes

-- Gate Keeda (18.8k points)

5.14.23 Regular Languages: GATE2011-24 [top](#)

<http://gateoverflow.in/3429>



Selected Answer

c) complement of regular Language is regular

18 votes

-- VOOTLA SRINIVAS (293 points)

5.14.24 Regular Languages: GATE2012-25 [top](#)

<http://gateoverflow.in/1609>



Selected Answer

$$L = \{ab, aa, baa\}$$

- | | | |
|-----------------|----------------------------------|---|
| 1. abaabaaaabaa | = ab aa baa ab aa | belong to L^* (combinations of strings in L) |
| 2. aaaabaaaa | = aa aa baa aa | belong to L^* |
| 3. baaaaabaaaab | = baa aa ab aa b | does not belong to L^* |
| 4. baaaaabaa | = baa aa ab aa | belong to L^* |

9 votes

-- Praveen Saini (53k points)

5.14.25 Regular Languages: GATE2013-8 [top](#)

<http://gateoverflow.in/1411>



Selected Answer

Concatenation of empty language with any language will give the empty language and $L_1^* = \phi^* = \epsilon$.

Therefore,

$$\begin{aligned} L_1 L_2^* \cup L_1^* \\ &= \phi \cdot (L_2)^* \cup \phi^* \\ &= \phi \cup \{\epsilon\} (\because \phi \text{ concatenated with anything is } \phi \text{ and } \phi^* = \{\epsilon\}) \\ &= \{\epsilon\}. \end{aligned}$$

Hence option (a) is True.

PS: $\phi^* = \epsilon$, where ϵ is the regular expression and the language it generates is $\{\epsilon\}$.

19 votes

-- Praveen Saini (53k points)

5.14.26 Regular Languages: GATE2014-1-15 [top](#)

<http://gateoverflow.in/1781>



Selected Answer

(A) is CFL and (B) and (D) are CSL. (C) is regular and regular expression for (C) would be

$$a^* b (a^* b a^* b)^+ a^*$$

9 votes

-- Arjun Suresh (287k points)

5.14.27 Regular Languages: GATE2014-2-15 [top](#)

<http://gateoverflow.in/1971>



Selected Answer

Option A.

$$L_1 = \{\epsilon, a, aa, aaa, aaaa, \dots\}$$

$$L_2 = \{\epsilon, b, bb, bbb, bbbb, \dots\}$$

$$L_1 \cdot L_2 = \left\{ \begin{array}{ll} \epsilon, & \\ a, & b, \\ aa, & ab, bb \\ aaa, & aab, abb, bbb, \\ aaaa, & aaab, aabb, abbb, bbbb, \dots \end{array} \right\}$$

$$L_1 \cdot L_2 = a^* b^*$$

Thus, $L_1 \cdot L_2$ is Regular.

(Also, since both L_1 and L_2 are Regular, their concatenation has to be Regular since Regular languages are closed under concatenation)

However, $L_1 \cdot L_2 \neq a^n b^n$. This is because in $a^* b^*$, the number of a 's and b 's can be different whereas in $a^n b^n$ they have to be the same.

19 votes

-- Viral Kapoor (2k points)

5.14.28 Regular Languages: GATE2014-2-36 [top](#)

<http://gateoverflow.in/1995>



Selected Answer

(A) is True. Though at first look both L_1 and L_2 looks non-regular, L_1 is in fact regular. The reason is the relation between 110 and 011.

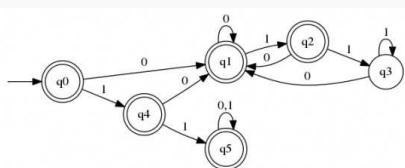
We cannot have two 110's in a string without a 011 or vice versa. And this would mean that we only need a finite number of states to check for acceptance of any word in this language.

That was just an intuitive explanation. Now I say that L contains all binary strings starting with 11. Yes, if a binary string starts with 11, it can never have more no. of 011 than 110.

Lets take an example:

11 011 011 -There are two 011's. But there are also two 110's. Similarly for any binary string starting with 11.

Using this property, DFA for L_1 can be constructed as follows:



36 votes

-- gatecse (13.1k points)

5.14.29 Regular Languages: GATE2015-2_51 [top](#)

<http://gateoverflow.in/8254>



Selected Answer

Ans A.

L_1 : all strings of length 3 or more with same start and end letter- because everything in middle can be consumed by x as per the definition of L .

L_2 : We need to compare number of a 's and b 's and these are not bounded. So, we need at least a DPDA.

L3: Any number of a's followed by any number of b's followed by any number of c's. Hence regular.

1 20 votes

-- Vikrant Singh (13.4k points)

5.14.30 Regular Languages: ISI2014-CS-4b [top](#)



Selected Answer

If L^* is regular, then L is not necessarily regular.

Let's take one possible example.

Let $\Sigma = \{a\}$ and consider the language $L = \{a^{2^n} \mid n \in \mathbb{N}\}$.

This language is not regular

and the language L^* is the language a^* , which is regular. To see this, notice that since L contains the string a , the language L^* contains all strings of the form a^n for any natural number n .

Hence, Option(2) Need not Regular is the correct choice.

1 6 votes

-- Leen Sharma (31.4k points)

5.14.31 Regular Languages: ISI2015-CS-5a [top](#)



Selected Answer

Consider a language $L_1 = \{a^n b^n \mid n \geq 0\}$ is a non regular language .

And take another language $L_2 = L_1^c$ which is also non regular .(Since regular set is closed under complementation.)

As we know **the union of any language and its complement is Σ^*** .

So $L_1 \cup L_2 = \Sigma^*$.

Which is regular.

1 4 votes

-- Manoj Kumar (37k points)

5.14.32 Regular Languages: TIFR2013-B-6 [top](#)



Selected Answer

A) False because - $L = a^* b^*$, $L' = a^n b^n$ Here $L/L' = a^*$. L/L' is regular, but L' is not.

B) True. If L is regular, L/L' is prefix of language. Regular languages are closed under Quotient/Prefix. So this is correct.

C) False $L' = \emptyset$. Then L/L' is Empty set whatever L is. Here L can be say $a^n b^n$. See definition of L/L' to see why L/L' should be empty set.

D) False because L/L' can accept prefixes of string of Language L , which may or may not be accepted by L itself. So L/L' is not subset. (It is not Superset either , because L' can be empty set)

E) False. Same explanation as C.

Answer :- B.

1 3 votes

-- Akash (41.5k points)

5.14.33 Regular Languages: TIFR2013-B-8 [top](#)

<http://gateoverflow.in/25670>



2

Selected Answer

Here, **OPTION D** is **regular**, reason is as follows :

$$L = \{ 0^m : m \geq 3 \}$$

Now, in L^* if we can generate **9 continuous powers of zero**, then further every power can be generated by **concatenating 0^9** .

$$\text{Here, } L = \{0^9, 0^{16}, 0^{25}, \dots\}$$

So, here are **9 continuous powers**:

$$0^{120} : 0^{16} 0^{16} 0^{16} 0^9 0^9 0^9 0^9 0^9 0^9$$

$$0^{121} : 0^{16} 0^{16} 0^{16} 0^{16} 0^{16} 0^{25}$$

$$0^{122} : 0^{16} 0^{16} 0^9 0^9 0^9 0^9 0^9 0^9 0^9 0^9$$

$$0^{123} : 0^{16} 0^{16} 0^{16} 0^{25} 0^{25} 0^{25}$$

$$0^{124} : 0^{16} 0^{18} 0^{18} 0^{18} 0^{18} 0^{18} 0^{18}$$

$$0^{125} : 0^{25} 0^{25} 0^{25} 0^{25} 0^{25}$$

$$0^{126} : 0^{18} 0^{18} 0^{18} 0^{18} 0^{18} 0^{18} \quad \{0^{18} \text{ can be generated as } 0^9 0^9\}$$

$$0^{127} : 0^{16} 0^{16} 0^{16} 0^{16} 0^9 0^9 0^9 0^9 0^9 0^9$$

$$0^{128} : 0^{16} 0^{16} 0^{16} 0^{16} 0^{16} 0^{16} 0^{16}$$

Now, 0^{129} can be given as $0^{120} 0^9$ and so on..

Every Further powers can be generated by concatenating 0^9 .

5 votes

-- Himanshu Agarwal (15.9k points)

5.14.34 Regular Languages: TIFR2014-B-12 [top](#)

<http://gateoverflow.in/27314>



Selected Answer

i) **False**

Regular Languages are not closed under Infinite Union and Intersection

$$L_1 \cup L_2 \cup L_3 \cup L_4 \cup \dots$$

For example :

$$ab \cup aabb \cup aaabbb \cup aaaabbbb \cup \dots$$

$$= \{a^n b^n, n \geq 1\}$$

So Infinite Union is not closed

$$L_1 \cap L_2 \cap L_3 \cap L_4 \cap \dots$$

$$= (L_1' \cup L_2' \cup L_3' \cup L_4' \cup \dots)'$$

As Infinite Union is not closed, So Infinite Intersection is also not closed

ii) **False.**

$a^* b^*$ is regular

its subset $a^n b^n, n \geq 1$ is not regular

a^* is regular

a^p, p is prime, is not regular

iii) **False**

$L = \{\}$ is regular

M be non-regular like $\{0^n 1^n \mid n > 0\}$.

$L \cdot M = \{\}$, is regular

11 votes

-- Praveen Saini (53k points)

5.14.35 Regular Languages: TIFR2015-B-10 [top](#)

<http://gateoverflow.in/30039>



Selected Answer

L_2 is finite, so regular.

L_1 is non-regular.

(It seems CFL to me as I think it can be implemented with the help of PDA, as stack can ensure $(m = n \vee n = p)$ and we can also ensure $(m + n + p \geq 10)$ with minimum states changes along with transitions.)

L_2 is actually {
 $a^p \mid$
 $p \leq 10\}$
 $\cup \{$
 $abc^p \mid$
 $p \leq 8\}$
 $\cup \{$
 $a^2b^2c^p \mid$
 $p \leq 6\}$
 $\cup \{$
 $a^3b^3c^p \mid$
 $p \leq 4\}$
 $\cup \{$
 $a^4b^4c^p \mid$
 $p \leq 2\}$
 $\cup \{$
 $a^5b^5\}$
 $\cup \{$
 $a^p \mid$
 $p \leq 10\}$
 $\cup \{$
 $a^pbc \mid$
 $p \leq 8\}$
 $\cup \{$
 $a^pb^2c^2 \mid$
 $p \leq 6\}$
 $\cup \{$
 $a^pb^3c^3 \mid$
 $p \leq 4\}$
 $\cup \{$
 $a^pb^4c^4 \mid$
 $p \leq 2\}$
 $\cup \{$
 $b^5c^5 \mid$
 $p \leq 10\}$

12 votes

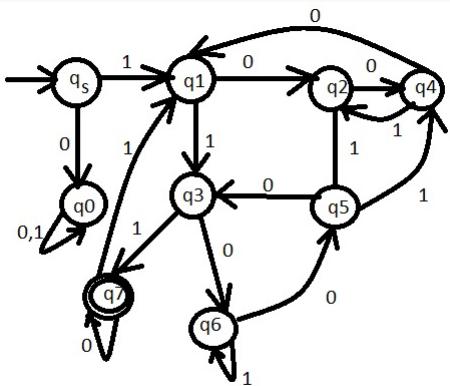
-- Praveen Saini (53k points)

5.14.36 Regular Languages: TIFR2015-B-6 [top](#)

<http://gateoverflow.in/29860>



Selected Answer



Answer will be (A)

if it starts with 1 it goes to final state

if it starts with 0 it goes to the reject state

8 votes

-- srestha (55.2k points)

5.15**Turing Machine(9)** top**5.15.1 Turing Machine: GATE 2016-2-44** top<http://gateoverflow.in/39596>

Consider the following languages.

$$L_1 = \{\langle M \rangle \mid M \text{ takes at least } 2016 \text{ steps on some input}\} ,$$

$$L_2 = \{\langle M \rangle \mid M \text{ takes at least } 2016 \text{ steps on all inputs}\} \quad \text{and}$$

$$L_3 = \{\langle M \rangle \mid M \text{ accepts } \epsilon\} ,$$

where for each Turing machine M , $\langle M \rangle$ denotes a specific encoding of M . Which one of the following is TRUE?

- A. L_1 is recursive and L_2, L_3 are not recursive
- B. L_2 is recursive and L_1, L_3 are not recursive
- C. L_1, L_2 are recursive and L_3 is not recursive
- D. L_1, L_2, L_3 are recursive

[gate2016-2](#) [theory-of-computation](#) [turing-machine](#)

Answer

5.15.2 Turing Machine: GATE1987-2m top<http://gateoverflow.in/80606>

State whether the following statements are TRUE or FALSE:

The problem as to whether a Turing machine M accepts input w is undecidable.[gate1987](#) [theory-of-computation](#) [turing-machine](#) [decidability](#)

Answer

5.15.3 Turing Machine: GATE2001-7 top<http://gateoverflow.in/748>Let a decision problem X be defined as follows: X : Given a Turing machine M over Σ and any word $w \in \Sigma$, does M loop forever on w ?

You may assume that the halting problem of Turing machine is undecidable but partially decidable.

- a. Show that X is undecidable

- b. Show that X is not even partially decidable

gate2001 theory-of-computation decidability turing-machine easy descriptive

[Answer](#)

5.15.4 Turing Machine: GATE2002-14 [top](#)

<http://gateoverflow.in/867>

The aim of the following question is to prove that the language $\{M \mid M \text{ is the code of the Turing Machine which, irrespective of the input, halts and outputs a } 1\}$, is undecidable. This is to be done by reducing from the language $\{M' \mid M' \text{ halts on } x\}$, which is known to be undecidable. In parts (a) and (b) describe the 2 main steps in the construction of M . In part (c) describe the key property which relates the behaviour of M on its input w to the behaviour of M' on x .

- a. On input w , what is the first step that M must make?
- b. On input w , based on the outcome of the first step, what is the second step M must make?
- c. What key property relates the behaviour of M on w to the behaviour of M' on x ?

gate2002 theory-of-computation decidability normal turing-machine descriptive difficult

[Answer](#)

5.15.5 Turing Machine: GATE2003-53 [top](#)

<http://gateoverflow.in/941>

A single tape Turing Machine M has two states q_0 and q_1 , of which q_0 is the starting state. The tape alphabet of M is $\{0, 1, B\}$ and its input alphabet is $\{0, 1\}$. The symbol B is the blank symbol used to indicate end of an input string. The transition function of M is described in the following table.

	0	1	B
q_0	$q_1, 1, R$	$q_1, 1, R$	Halt
q_1	$q_1, 1, R$	$q_0, 1, L$	q_0, B, L

The table is interpreted as illustrated below.

The entry $(q_1, 1, R)$ in row q_0 and column 1 signifies that if M is in state q_0 and reads 1 on the current page square, then it writes 1 on the same tape square, moves its tape head one position to the right and transitions to state q_1 .

Which of the following statements is true about M ?

- A. M does not halt on any string in $(0 + 1)^+$
- B. M does not halt on any string in $(00 + 1)^*$
- C. M halts on all strings ending in a 0
- D. M halts on all strings ending in a 1

gate2003 theory-of-computation turing-machine normal

[Answer](#)

5.15.6 Turing Machine: GATE2003-54 [top](#)

<http://gateoverflow.in/355>

Define languages L_0 and L_1 as follows :

$$L_0 = \{\langle M, w, 0 \rangle \mid M \text{ halts on } w\}$$

$$L_1 = \{\langle M, w, 1 \rangle \mid M \text{ does not halts on } w\}$$

Here $\langle M, w, i \rangle$ is a triplet, whose first component M is an encoding of a Turing Machine, second component w is a string, and third component i is a bit.

Let $L = L_0 \cup L_1$. Which of the following is true?

- A. L is recursively enumerable, but L' is not
- B. L' is recursively enumerable, but L is not
- C. Both L and L' are recursive
- D. Neither L nor L' is recursively enumerable

theory-of-computation turing-machine gate2003 difficult

[Answer](#)

5.15.7 Turing Machine: GATE2004-89 [top](#)

<http://gateoverflow.in/1083>

L_1 is a recursively enumerable language over Σ . An algorithm A effectively enumerates its words as w_1, w_2, w_3, \dots . Define another language L_2 over $\Sigma \cup \{\#\}$ as $\{w_i \# w_j \mid w_i, w_j \in L_1, i < j\}$. Here $\#$ is new symbol. Consider the following assertions.

- $S_1 : L_1$ is recursive implies L_2 is recursive
- $S_2 : L_2$ is recursive implies L_1 is recursive

Which of the following statements is true?

- A. Both S_1 and S_2 are true
- B. S_1 is true but S_2 is not necessarily true
- C. S_2 is true but S_1 is not necessarily true
- D. Neither is necessarily true

[gate2004](#) [theory-of-computation](#) [turing-machine](#) [difficult](#)

[Answer](#)

5.15.8 Turing Machine: GATE2014-2-35 [top](#)

<http://gateoverflow.in/1994>

Let $\langle M \rangle$ be the encoding of a Turing machine as a string over $\Sigma = \{0, 1\}$. Let

$$L = \{\langle M \rangle \mid M \text{ is a Turing machine that accepts a string of length 2014}\}.$$

Then L is

- A. decidable and recursively enumerable
- B. undecidable but recursively enumerable
- C. undecidable and not recursively enumerable
- D. decidable but not recursively enumerable

[gate2014-2](#) [theory-of-computation](#) [turing-machine](#) [normal](#)

[Answer](#)

5.15.9 Turing Machine: GateBook Mock Test_2(TOC) [top](#)

<http://gateoverflow.in/116951>

Given TMs $M_1, M_2, M_3, \dots, M_n$ and $L = \{x \mid \text{Every } M_i \text{ halts on input } x\}$ which of the following is true about L ?

- (A) L is recursively enumerable but not recursive
- (B) L is Recursive but not Context free
- (C) L is Not Recursively Enumerable
- (D) L is regular

[gatebook_mt2](#) [theory-of-computation](#) [decidability](#) [turing-machine](#)

[Answer](#)

Answers: Turing Machine

5.15.1 Turing Machine: GATE 2016-2-44 [top](#)

<http://gateoverflow.in/39596>



Selected Answer

L_3 is not recursive as it asks if $L(M)$ contains ϵ which is a non-trivial property of r.e. languages and hence undecidable as per Rice's theorem.

L_1 and L_2 are slightly trickier as these are not describing properties of recursively enumerable languages, but rather of Turing machines. So, we can see if there is some procedure for deciding these.

For L_1 we can give the TM an input of length 2016. Now, it should at least make 2016 steps or reach the halt state before completing the input processing. The second case is possible only if the TM reaches a halt state before reaching the end of string (blank) of input, for all possible inputs of length at least 2016 and can be decided. So, we can be sure that otherwise TM will have at least 2016 steps making L_1 recursive.

L_2 is recursive and it is more easier to prove. For the complement of L_2 we need M to make less than 2016 steps for some input and we can just give it all possible inputs of length less than 2016 and see if it reaches a halt state within 2016 steps. Thus complement of L_2 is recursive $\implies L_2$ is recursive.

So, answer here is C.

30 votes

-- Arjun Suresh (287k points)

One more possible approach:

L_3 is not recursive (Can be proved using Rice theorem).

Lets talk about L_1 and L_2 , and let me take L_2 first

$L_2 = \{(M) | M \text{ takes at least 2016 steps on all inputs}\}$, I want to check if for all strings in Σ^* M takes more than or equal to 2016 steps or not.

First of all i will restrict number of steps in M to 2016, and i will never run M more than 2016 steps. Because for any string, if M halts (accepts then halts or rejects then halts, does not matter) in less than 2016 steps then that string is not in L_2 . And if M does not halts within 2016 steps (after 2016 steps, I am not interested whether M is in infinite loop or will halt eventually) then string is in L_2

\implies Number of steps to be run in M is not more than 2016.

Since we bound the number of steps that M runs on an input, then there is no point on looking at any strings that are longer than that number, since if a TM is allowed to run for at most c steps, it is not possible for that TM to "process" any input symbol beyond the c th symbol!

\implies Length of input string is less than 2016. (If i can decide for these strings then L_2 is Recursive otherwise not Recursive) And there are finite strings having length less than 2016.

Now my task reduces to : "Take each string in this finite set and run M for finite number of steps"

The number of possible inputs is finite, and the number of steps M runs on each input is finite, therefore M is guaranteed to halt and decide the language. Hence L_2 is recursive.

(If we can decide for all inputs then we can also decide for some inputs therefore L_1 is also recursive (Reduction), but we can think of L_1 as an independent problem)

$L_1 = \{(M) | M \text{ takes at least 2016 steps on some input}\}$, I want to check if there exist any string in Σ^* for which M takes more than or equal to 2016 steps.

With same reasoning i can say that we will run M for finite number of steps and input string set is also finite. The only difference is, we can stop giving input once we find any string taking atleast 2016 steps, whereas in L_2 we have to check for all set of input strings length less than 2016.

Therefore L_1 is also recursive.

L_1, L_2 Recursive

L_3 : Not Recursive.

C is answer.

Ref: Problem number one in this pdf: https://www.cs.rice.edu/~nakleh/COMP481/final_review_sp06_sol.pdf.

23 votes

-- Sachin Mittal (6k points)

5.15.2 Turing Machine: GATE1987-2m [top](#)

<http://gateoverflow.in/80606>



Selected Answer

Yes The problem as to whether a Turing machine M accepts input w is undecidable which is the well known Halting Problem.

If string w is going to loop then we can not determine if it will be eventually accepted by TM or not.

2 votes

-- Prashant Singh (47.6k points)

5.15.3 Turing Machine: GATE2001-7 [top](#)

<http://gateoverflow.in/748>



Selected Answer

The question asks if M loops forever on w . If M loops forever on w , M wouldn't halt on w . And if M doesn't halt on w , M should loop forever. So, this problem is exactly same as asking if " M doesn't halt on w ", which is the complement of halting problem and is not even partially decidable. So, X is not even partially decidable.

13 votes

-- Arjun Suresh (287k points)

5.15.4 Turing Machine: GATE2002-14 [top](#)

<http://gateoverflow.in/867>



Selected Answer

(a) M erases its input w and simulates the moves of M' on x . Thus if M' halts on x , M accepts any input (Σ^*) and if M' doesn't halt on x , M accepts no string (\emptyset)

(b) Give the description of M - $\langle M \rangle$ to the TM that decides L . If TM accepts $\langle M \rangle$, M halts on all inputs $\Rightarrow M'$ accepts x . If TM rejects $\langle M \rangle$, M doesn't halt on some input $\Rightarrow M'$ doesn't halt on x , due to our construction of M in 1st step. Thus we decide halting problem

(c) M halting on all inputs w is the key property relating to M' which is halting on a given input x

2 votes

-- Arjun Suresh (287k points)

5.15.5 Turing Machine: GATE2003-53 [top](#)

<http://gateoverflow.in/941>



Selected Answer

option A. or epsilon is only accepted i.e tape contain B as the first character

10 votes

-- Supromit Roy (689 points)

5.15.6 Turing Machine: GATE2003-54 [top](#)

<http://gateoverflow.in/355>



Selected Answer

Both L and L' are undecidable and not even semi-decidable (not recursively-enumerable). Because halting problem can be solved with both L and L' .

Halting problem can be stated as follows: A machine M and a word w are given. You have to tell, if M halts on w .

So, to solve halting problem $\langle M, w \rangle$ using L , just give $\langle M, w, 0 \rangle$ and $\langle M, w, 1 \rangle$ to two instances of T which is the assumed Turing machine for L . If T accepts the triplet $\langle M, w, 0 \rangle$, it means M halts on $w \Rightarrow$ we have solved halting problem. If T accepts the triplet $\langle M, w, 1 \rangle$, it means M doesn't halt on $w \Rightarrow$ we have solved halting problem. We know that either $\langle M, w, 0 \rangle$ or $\langle M, w, 1 \rangle$ is in L . So, if L is recursively enumerable, T is bound to stop on at least one of these inputs (TM for a recursively enumerable language stops and accepts, when provided with a word in its language).

Hence, if L is recursively enumerable we can solve halting problem $\Rightarrow L$ is not recursively enumerable. Similarly, we can also show that halting problem can be solved with L' (shown at end)

Hence, neither L nor L' is recursively enumerable.

To solve halting problem $\langle M, w \rangle$ using L' , just give $\langle M, w, 0 \rangle$ and $\langle M, w, 1 \rangle$ to two instances of T' which is the assumed Turing machine for L' . If T' accepts the triplet $\langle M, w, 0 \rangle$, it means M does not halt on $w \Rightarrow$ we have solved halting problem. If T accepts the triplet $\langle M, w, 1 \rangle$, it means M halt on $w \Rightarrow$ we have solved halting problem. We know that either $\langle M, w, 0 \rangle$ or $\langle M, w, 1 \rangle$ is in L' . So, if L' is recursively enumerable, T' is bound to stop on at least one of these inputs (TM for a recursively enumerable language stops and accepts, when provided with a word in its language).

Hence, if L' is recursively enumerable we can solve halting problem $\Rightarrow L'$ is not recursively enumerable.

PS: If the bit part of the triplet is absent then L_0 is halting problem and L_1 is its complement and $L_0 \cup L_1 = \Sigma^*$, which is regular. Lets see how it happens.

Let the alphabet set be $\{0, 1\}$. Now for any string like 0010101 there are only two options- belong to L or belong to L' as this is what complement says. Now, lets take the case for L_0 and a string $001\dots10 - 01 - 1$, ("-" shown for notation purpose only) where the first component describes a TM M followed by input " $w = 01$ " and last bit "1". Now suppose M halts on "01". Still the given input is not in L_0 as the last bit is "1" and not "0" as required by L_0 . So, this input must be in L'_0 . But since M halts on w , this input is not in L_1 either. Similarly, we can get an infinite set of strings which does not belong to both L_0 and L_1 and this makes their union not Σ^* but an irregular (not r.e. as proved earlier) set. If the last bit is removed from the definition of L_0 and L_1 , then any string should be present in either L_0 or L_1 and their union would be Σ^* .

29 votes

-- gatecse (13.1k points)

5.15.7 Turing Machine: GATE2004-89 [top](#)

<http://gateoverflow.in/1083>



S_1 is TRUE.

If L_1 is recursive L_2 must also be recursive. Because to check if a word $w = w_i \# w_j$ belong to L_2 , we can give w_i and w_j to the decider for L_1 and if both are accepted then w belong to L_1 and not otherwise.

S_2 is TRUE.

With a decider for L_2 we can make a decider for L_1 as follows. Let w_1 be the first string enumerated by algorithm A for L_1 . Now, to check if a word w belongs to L_1 , make a string $w' = w_1 \# w$ and give it to the decider for L_2 and if accepted, then w belongs to L_1 and not otherwise.

So, answer must be A.

14 votes

-- Arjun Suresh (287k points)

5.15.8 Turing Machine: GATE2014-2-35 [top](#)

<http://gateoverflow.in/1994>



There are only a finite number of strings of length 2014. So, we can give all those strings to TM simulating each string for 1 step, then 2 step and so on (dovetailing), and if the TM accepts any of them ("yes" case of TM), we can say "yes". So, L is recursively enumerable.

(If the TM doesn't accept any string of length 2014, it can go to an infinite loop ("no" case of TM), and hence we can't say the method is decidable).

Now, to prove whether the problem is decidable or not we can make use of Rice's theorem. Rice's theorem (I) states that any non-trivial property of $L(TM)$ is undecidable. $L(TM)$ has a string of length 2014 is a non-trivial property as there are TMs whose language contains such a string and there are TMs whose language doesn't have such a string. So, the given problem is undecidable.

http://gatecse.in/wiki/Rice%27s_Theorem_with_Examples

29 votes

-- Arjun Suresh (287k points)

5.15.9 Turing Machine: GateBook Mock Test_2(TOC) [top](#)

<http://gateoverflow.in/116951>

Answer is (A)

L is the collection of all strings for which M₁, M₂, ..., M_n Halts on. i.e , They come to stop either by accepting or rejecting. As we can see , there is a chance that it can also go to infinite loop. This is same as halting problem and the language is R.E but not Recursive.

2 votes

-- junk_mayavi (499 points)

Brought to you by GATE Overflow
<http://gateoverflow.in/>