[GATE - 2005]

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

[GATE - 1993]

The following Pascal program segments finds the largest number in a two-dimensional integer array $A[0 \ldots n-1, 0 \ldots n-1]$ using a single loop. Fill up the boxes to complete the program and write against A, B, C and 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
```

[GATE - 1994]

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

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

[GATE - 1994]

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

[GATE - 1997]

An array A contains $n \geq 1$ positive integers in the locations $A[1], A[2], \ldots A[n]$. The following program fragment prints the length of a shortest sequence of consecutive elements of A, $A[i], A[i+1], \ldots, A[j]$ such that the sum of their values is $\geq M$, a given positive number. It prints '$n+1$' if no such sequence exists. Complete the program by filling in the boxes. In each case use the simplest possible expression. Write only the line number and the contents of the box.

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

[GATE - 1998]

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. $10j + i + 89$

[GATE - 2000]

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

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

The sum of the elements of the array v is

A. 0

B. $n-1$

C. $n^2 - 3n + 2$

D. $n^2 \frac{(n+1)}{2}$

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.

[GATE - 2005]

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

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. $oldc = \text{“}abc\text{”}, newc = \text{“}dab\text{”}$
2. $oldc = \text{“}cde\text{”}, newc = \text{“}bcd\text{”}$
3. $oldc = \text{“}bca\text{”}, newc = \text{“}cda\text{”}$
4. $oldc = \text{“}abc\text{”}, newc = \text{“}bac\text{”}$

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

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

[GATE - 2014]

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 $listA$.

[GATE - 2015]

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 _____.

[GATE - 2011]

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.

[GATE - 1996]

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. $(3, 8)$

[GATE - 1996]

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

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 _______;

}
```

[GATE - 2009]

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

[GATE - 2003]

Let $T(n)$ be the number of different binary search trees on n distinct elements. Then $T(n) = \sum_{k=1}^{n} T(k-1)T(x)$, where x is

A. $n-k+1$ B. $n-k$ C. $n-k-1$ D. $n-k-2$

[GATE - 2003]

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

[GATE - 2004]

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

[GATE - 2004]

A program takes as input a balanced binary search tree with n leaf nodes and computes the value of a function $g(x)$ for each node x. If the cost of computing $g(x)$ is:

$$\min\left(\begin{array}{c}\text{number of leaf-nodes}\\ \text{in left-subtree of } x\end{array}, \begin{array}{c}\text{number of leaf-nodes}\\ \text{in right-subtree of } x\end{array}\right)$$

Then the worst-case time complexity of the program is?

A. $\Theta(n)$

B. $\Theta(n \log n)$

C. $\Theta(n^2)$

D. $\Theta(n^2 \log n)$

[GATE - 2005]

The numbers $1,2,\ldots.n$ are inserted in a binary search tree in some order. In the resulting tree, the right subtree of the root contains p nodes. The first number to be inserted in the tree must be

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

[GATE - 2005]

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. $2, 1, 4, 3, 7, 8, 6, 5$

[GATE - 2006]

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. $\{79, 14, 72, 56, 16, 53, 55\}$

[GATE - 2007]

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

[GATE - 2008]

You are given the postorder traversal, P, of a binary search tree on the n elements $1, 2, \ldots, n$. You have to determine the unique binary search tree that has P as its postorder traversal. What is the time complexity of the most efficient algorithm for doing this?

A. $\Theta(\log n)$

B. $\Theta(n)$

C. $\Theta(n \log n)$

D. None of the above, as the tree cannot be uniquely determined

[GATE - 2008]

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 $O(n)$

[GATE - 2008]

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

[GATE - 2008]

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

[GATE - 2008]

How many distinct BSTs can be constructed with 3 distinct keys?

A. 4 B. 5 C. 6 D. 9

[GATE - 2009]

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

[GATE - 2012]

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. $\Theta(\log n)$

[GATE - 2013]

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. $15, 10, 23, 25, 20, 35, 42, 39, 30$

[GATE - 2013]

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. $O(n \log n)$

[GATE - 2014]

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 _____.

[GATE - 2015]

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

[GATE - 2015]

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

[GATE - 2015]

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

[GATE - 2016]

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.

[GATE - 2017]

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.

[GATE - 2017]

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. $7, 6, 2, 10, 9, 8, 15, 16, 17, 20, 19, 12$

[GATE - 2010]

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.

[GATE - 1987]

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?

[GATE - 1987]

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.

[GATE - 1987]

Construct a binary tree whose preorder traversal is

- K L N M P R Q S T

and inorder traversal is

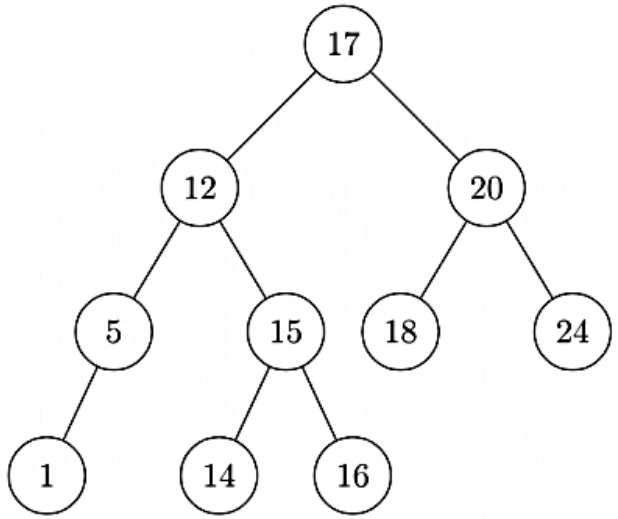
- N L K P R M S Q T

[GATE - 1988]

Define the height of a binary tree or subtree and also define a height-balanced (AVL) tree.

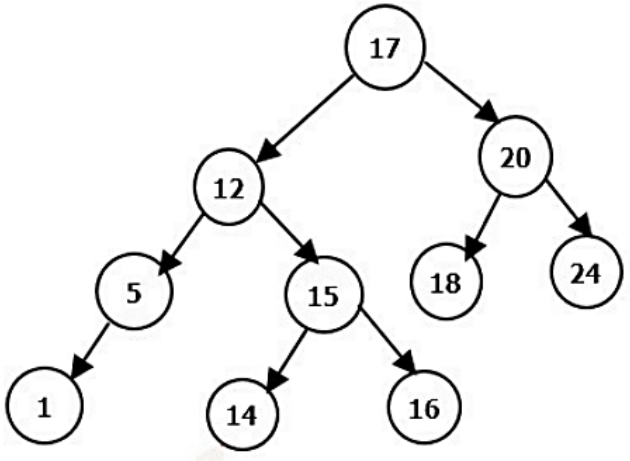
[GATE - 1988]

Mark the balance factor of each on the tree given on the below figure and state whether it is height-balanced.



[GATE - 1988]

Consider the tree given in the below figure, insert 13 and show the new balance factors that would arise if the tree is not rebalanced. Finally, carry out the required rebalancing of the tree and show the new tree with the balance factors on each mode.



[GATE - 1990]

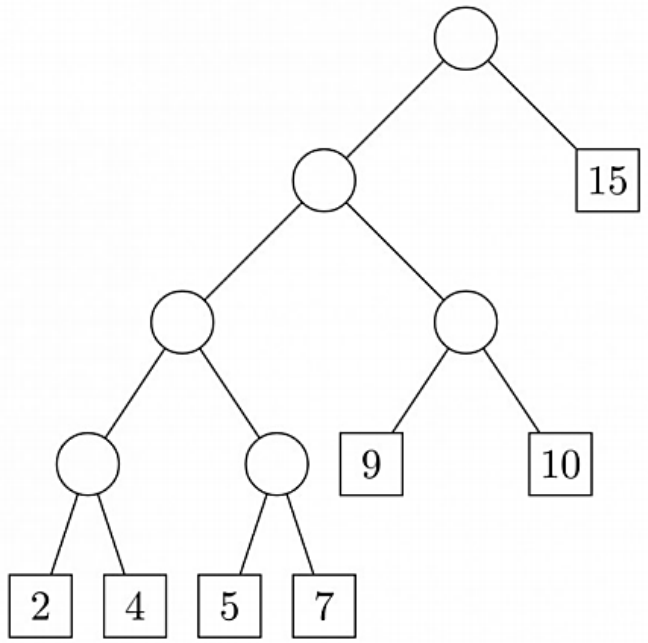
Choose the correct alternatives (More than one may be correct).

The total external path length, EPL, of a binary tree with n external nodes is, $EPL = \sum_w Iw$, where I_w is the path length of external node w),

A. $\leq n^2$ always.
B. $\geq n \log_2 n$ always.
C. Equal to n^2 always.
D. $O(n)$ for some special trees.

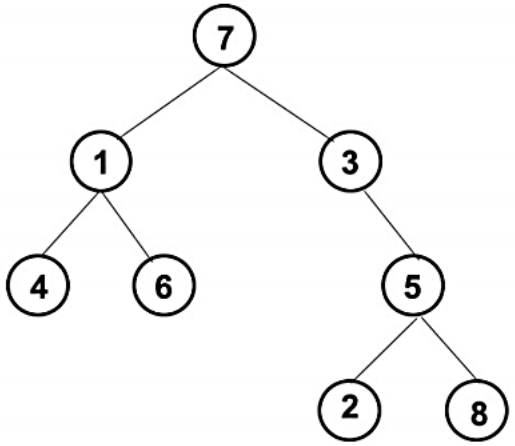
[GATE - 1991]

The weighted external path length of the binary tree in figure is _____



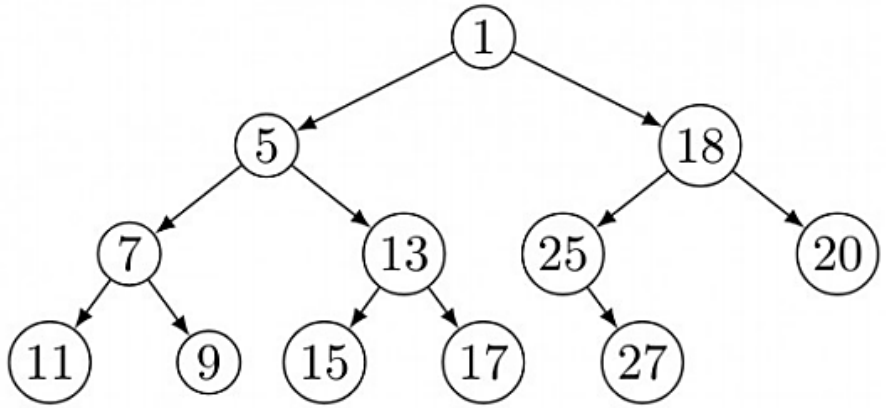
[GATE - 1991]

If the binary tree in figure is traversed in inorder, then the order in which the nodes will be visited is _____



[GATE - 1991]

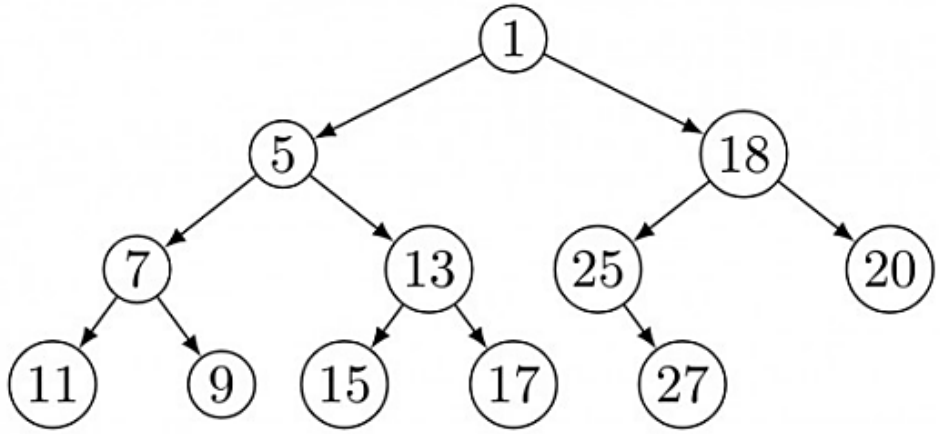
Consider the binary tree in the figure below:



(a). What structure is represented by the binary tree?

[GATE - 1991]

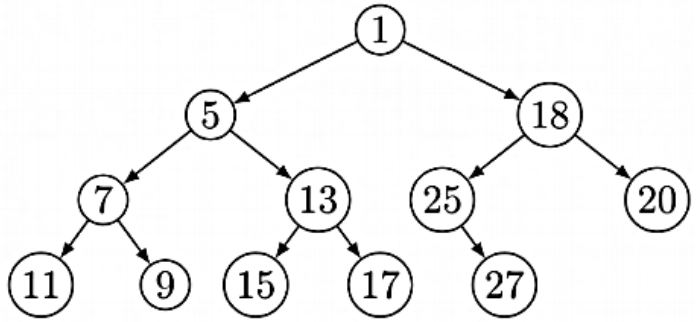
Consider the binary tree in the figure below:



Give different steps for deleting the node with key 5 so that the structure is preserved.

[GATE - 1991]

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?

[GATE - 1993]

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.

[GATE - 1994]

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.

[GATE - 1995]

A binary tree T has n leaf nodes. The number of nodes of degree 2 in T is

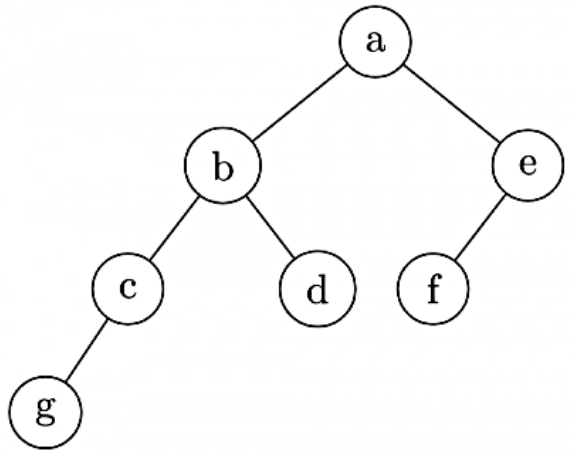
A. $\log_2 n$ B. $n-1$ C. n D. 2^n

[GATE - 1995]

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.

[GATE - 1996]

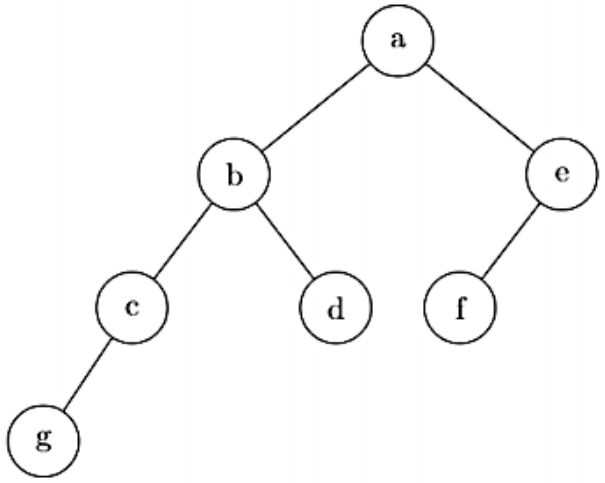
In the balanced binary tree in the below figure, how many nodes will become unbalanced when a node is inserted as a child of the node "g"?



A. 1 B. 3 C. 7 D. 8

[GATE - 1996]

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

[GATE - 1997]

A size-balanced binary tree is a binary tree in which for every node the difference between the number of nodes in the left and right subtree is at most 1. The distance of a node from the root is the length of the path from the root to the node. The height of a binary tree is the maximum distance of a leaf node from the root.

A. Prove, by using induction on h, that a size-balance binary tree of height h contains at least 2^h nodes.

B. In a size-balanced binary tree of height $h \geqslant 1$, how many nodes are at distance $h-1$ from the root? Write only the answer without any explanations.

[GATE - 1997]

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

[GATE - 1998]

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: a f b c d g e

Postorder: a f c g e d b

[GATE - 2000]

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. $(1\,(2\,3\,NULL)\,(4\,5))$

[GATE - 2000]

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

[GATE - 2002]

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. $\log_{\frac{3}{2}} n$

[GATE - 2002]

Draw all binary trees having exactly three nodes labeled A, B and C on which preorder traversal gives the sequence C, B, A.

[GATE - 2004]

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

Consider the following C program segment

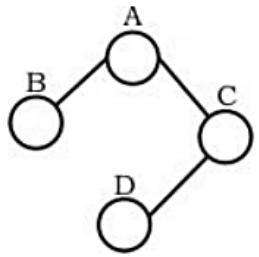
```
struct CellNode{
    struct CellNode *leftChild
    int element;
    struct CellNode *rightChild;
    };

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);
```

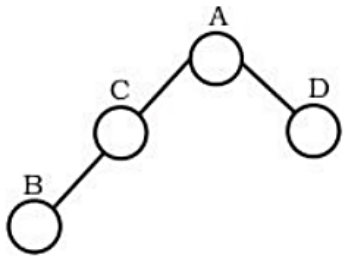
The value returned by the function DoSomething when a pointer to the root of a non-empty tree is passed as argument is

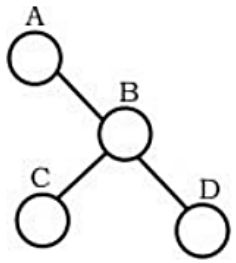
A. The number of leaf nodes in the tree
B. The number of nodes in the tree
C. The number of internal nodes in the tree
D. The height of the tree

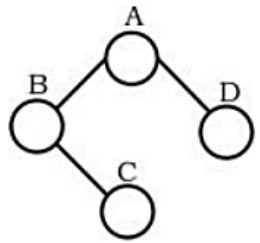
[GATE - 2004]

Which one of the following binary trees has its inorder and preorder traversals as $BCAD$ and $ABCD$, respectively?

A. 

B. 

C. 

D. 

[GATE - 2005]

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. $95, 50, 60, 40, 27, 23, 22, 25, 10, 9, 15, 29$

[GATE - 2005]

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

[GATE - 2006]

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. $2^n - 1$

[GATE - 2006]

An array X of n distinct integers is interpreted as a complete binary tree. The index of the first element of the array is 0. The index of the parent of element $X[i], i \neq 0$, is?

A. $\left\lfloor \frac{i}{2} \right\rfloor$

B. $\left\lceil \frac{i-1}{2} \right\rceil$

C. $\left\lceil \frac{i}{2} \right\rceil$

D. $\left\lceil \frac{i}{2} \right\rceil - 1$

[GATE - 2006]

An array X of n distinct integers is interpreted as a complete binary tree. The index of the first element of the array is 0. If the root node is at level 0, the level of element $X[i]$, $i \neq 0$, is

A. $\lfloor \log_2 i \rfloor$
B. $\lceil \log_2 (i+1) \rceil$
C. $\lfloor \log_2 (i+1) \rfloor$
D. $\lceil \log_2 i \rceil$

[GATE - 2006]

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

[GATE - 2007]

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}

[GATE - 2007]

The maximum number of binary trees that can be formed with three unlabeled nodes is:

A. 1 B. 5 C. 4 D. 3

[GATE - 2007]

The inorder and preorder traversal of a binary tree are

d b e a f c g and a b d e c f g , respectively

The postorder traversal of the binary tree is:

A. d e b f g c a

B. e d b g f c a

C. e d b f g c a

D. d e f g b c a

[GATE - 2007]

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

[GATE - 2008]

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

[GATE - 2008]

A binary tree with $n > 1$ nodes has n_1, n_2 and n_3 nodes of degree one, two and three respectively. The degree of a node is defined as the number of its neighbours.

n_3 can be expressed as

A. $n_1 + n_2 - 1$
B. $n_1 - 2$
C. $[((n_1 + n_2)/2)]$
D. $n_2 - 1$

[GATE - 2008]

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. $n_2 + n_1 - 2$

[GATE - 2010]

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. $n-1$

[GATE - 2011]

We are given a set of n distinct elements and an unlabeled binary tree with n nodes. In how many ways can we populate the tree with the given set so that it becomes a binary search tree?

A. 0 B. 1 C. $n!$ D. $\frac{1}{n+1}.^{2n}C_n$

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
            }
       }
}
```

The appropriate expressions for the two boxes **B1** and **B2** are:

A. **B1**: $(1 + \text{height}(n \rightarrow \text{right}))$; **B2**: $(1 + \max(h1, h2))$
B. **B1**: $(\text{height}(n \rightarrow \text{right}))$; **B2**: $(1 + \max(h1, h2))$
C. **B1**: $\text{height}(n \rightarrow \text{right})$; **B2**: $\max(h1, h2)$
D. **B1**: $(1 + \text{height}(n \rightarrow \text{right}))$; **B2**: $\max(h1, h2)$

[GATE - 2014]

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 _____.

[GATE - 2015]

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

[GATE - 2015]

A binary tree T has 20 leaves. The number of nodes in T having two children is _____.

[GATE - 2015]

Consider a binary tree T that has 200 leaf nodes. Then the number of nodes in T that have exactly two children are _____.

[GATE - 2016]

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. $+-167*2\wedge5-34*$
B. $-+1*67\wedge2-5*34$
C. $-+1*76\wedge2-5*43$
D. $176*+2543*-\wedge-$

[GATE - 2018]

The postorder traversal of a binary tree is $8, 9, 6, 7, 4, 5, 2, 3, 1$. The inorder traversal of the same tree is $8, 6, 9, 4, 7, 2, 5, 1, 3$. The height of a tree is the length of the longest path from the root to any leaf. The height of the binary tree above is _____

[GATE - 2019]

Let T be a full binary tree with 8 leaves. (A full binary tree has every level full.) Suppose two leaves a and b of T are chosen uniformly and independently at random. The expected value of the distance between a and b in T (ie., the number of edges in the unique path between a and b) is (rounded off to 2 decimal places) _____.

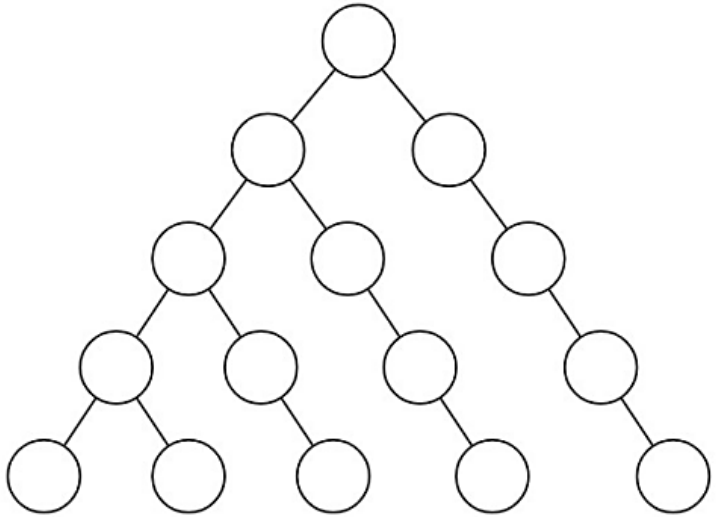
[GATE - 2012]

Consider a complete binary tree of height n, where each edge is one Ohm resistor. Suppose all the leaves of the tree are tied together. Approximately how much is the effective resistance from the root to this bunch of leaves for very large n?

a. Exponential in n.
b. Cubic in n.
c. Linear in n.
d. Logarithmic in n.
e. Of the order square root of n.

[GATE - 2013]

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.

[GATE - 2014]

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.

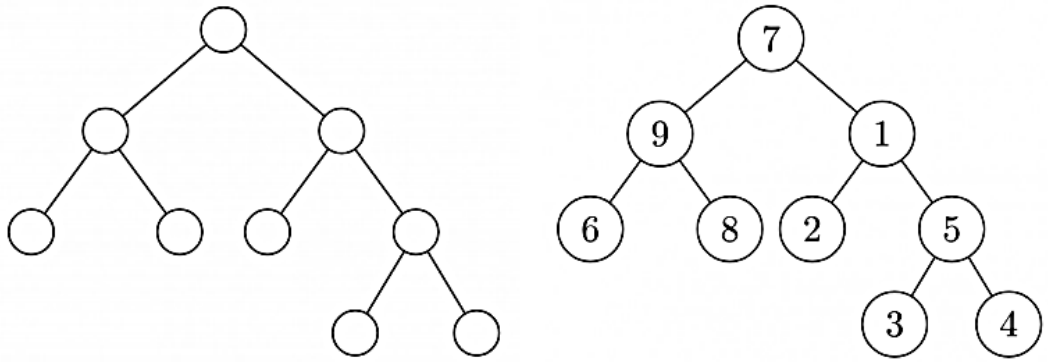
in-order: $a, b, c, d, e, f, g, h, i, j, k$

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.

First, consider the tree on the left.



On the right, the nine nodes of the tree have been assigned numbers from the set $\{1,2,\ldots,9\}$ so that for every node, the numbers in its left subtree and right subtree lie in disjoint intervals (that is, all numbers in one subtree are less than all numbers in the other subtree). How many such assignments are possible? Hint: Fix a value for the root and ask what values can then appear in its left and right subtrees.

A. $2^9 = 512$ B. $2^4.3^2.5.9 = 6480$ C. $2^3.3.5.9 = 1080$ D. $2^4 = 16$ E. $2^3.3^3 = 216$

[GATE - 2018]

Consider the following implementation of a binary tree data strucrure. The operator $+$ denotes list-concatenation. That is, $[a,b,c]+[d,e]=[a,b,c,d,e]$.

```
struct TreeNode:
 int value
 TreeNode leftChild
 TreeNode rightChild

function preOrder(T):
 if T == null:
  return []
 else:
  return [T.value] + preOrder(T.leftChild) + preOrder(T.rightChild)

function inOrder(T):
 if T == null:
  return []
 else:
  return inOrder(T.leftChild) + [T.value] + inOrder(T.rightChild)

function postOrder(T):
 if T == null:
  return []
 else:
  return postOrder(T.leftChild) + postOrder(T.rightChild) + [T.value]
```

For some T the functions inOrder(T) and preOrder(T) return the following:

inOrder(T) : $[12,10,6,9,7,2,15,5,1,13,4,3,8,14,11]$

preOrder(T) : $[5,2,10,12,9,6,7,15,13,1,3,4,14,8,11]$

What does postOrder(T) return ?

A. $[12,6,10,7,15,2,9,1,4,13,8,11,14,3,5]$
B. $[11,8,14,4,3,1,13,15,7,6,9,12,10,2,5]$
C. $[11,14,8,3,4,13,1,5,15,2,7,9,6,10,12]$
D. $[12,6,7,9,10,15,2,1,4,8,11,14,3,13,5]$
E. Cannot be uniquely determined from given information.

[GATE - 1989]

Answer the following:

Which one of the following statements (s) is/are FALSE?

A. Overlaying is used to run a program, which is longer than the address space of the computer.
B. Optimal binary search tree construction can be performed efficiently by using dynamic programming.
C. Depth first search cannot be used to find connected components of a graph.
D. Given the prefix and postfix walls over a binary tree, the binary tree can be uniquely constructed.

[GATE – 1992]

How many edges can there be in a forest with p components having n vertices in all?

[GATE - 1997]

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

[GATE - 2008]

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.

[GATE - 2008]

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

[GATE - 2014]

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 $E_3 = \{(u, v) |$ there is a path of length ≤ 2 from u to v in $E\}$
D. $G_4 = (V_4, E)$ where V_4 is the set of vertices in G which are not isolated

[GATE - 2016]

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 _____.

[GATE - 1989]

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

[GATE - 1996]

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

[GATE - 1996]

Insert the characters of the string $K\ R\ P\ C\ S\ N\ Y\ T\ J\ M$ into a hash table of size 10. Use the hash function

$$h(x) = (ord(x) - ord(\text{" } a \text{ "}) + 1) \mod 10$$

and linear probing to resolve collisions.

A. Which insertions cause collisions?

B. Display the final hash table.

[GATE - 1997]

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?

[GATE - 2004]

Given the following input $(4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199)$ and the hash function x mod 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

[GATE - 2005]

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

[GATE - 2006]

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

[GATE - 2007]

Consider a hash table of size seven, with starting index zero, and a hash function $(3x + 4) \mod 7$. Assuming the hash table is initially empty, which of the following is the contents of the table when the sequence $1, 3, 8, 10$ is inserted into the table using closed hashing? Note that − denotes an empty location in the table.

A. 8, –, –, –, –, –, 10

B. 1, 8, 10, –, –, –, 3

C. 1, –, –, –, –, –, 3

D. 1, 10, 8, –, –, –, 3

[GATE - 2007]

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

[GATE - 2008]

Consider a hash table of size 11 that uses open addressing with linear probing. Let $h(k) = k \mod 11$ be the hash function used. A sequence of records with keys

43 36 92 87 11 4 71 13 14

is inserted into an initially empty hash table, the bins of which are indexed from zero to ten. What is the index of the bin into which the last record is inserted?

A. 3 B. 4 C. 6 D. 7

The keys $12, 18, 13, 2, 3, 23, 5$ and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function $h(k) = k \mod 10$ and linear probing. What is the resultant hash table?

A.

0	
1	
2	2
3	23
4	
5	15
6	
7	
8	18
9	

B.

0	
1	
2	12
3	13
4	
5	5
6	
7	
8	18
9	

C.

0	
1	
2	12
3	13
4	2
5	3
6	23
7	5
8	18
9	15

D.

0	
1	
2	2, 12
3	13, 3, 23
4	
5	5, 15
6	
7	
8	18
9	

[GATE - 2010]

A hash table of length 10 uses open addressing with hash function $h(k) = k \mod 10$, and linear probing. After inserting 6 values into an empty hash table, the table is shown as below

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

Which one of the following choices gives a possible order in which the key values could have been inserted in the table?

A. $46, 42, 34, 52, 23, 33$
B. $34, 42, 23, 52, 33, 46$
C. $46, 34, 42, 23, 52, 33$
D. $42, 46, 33, 23, 34, 52$

[GATE - 2010]

A hash table of length 10 uses open addressing with hash function $h(k) = k \, mod \, 10$, and linear probing. After inserting 6 values into an empty hash table, the table is shown as below

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

How many different insertion sequences of the key values using the same hash function and linear probing will result in the hash table shown above?

A. 10 B. 20 C. 30 D. 40

[GATE - 2014]

Consider a hash table with 9 slots. The hash function is $h(k) = k \mod 9$. The collisions are resolved by chaining. The following 9 keys are inserted in the order: $5, 28, 19, 15, 20, 33, 12, 17, 10$. The maximum, minimum, and average chain lengths in the hash table, respectively, are

A. $3, 0$, and 1

B. $3, 3$, and 3

C. $4, 0$, and 1

D. $3, 0$, and 2

[GATE - 2014]

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. $(97 \times 96 \times 95/(3! \times 100^3)$

[GATE - 2015]

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. $h(i) = (12 * i^2) \bmod 10$

[GATE - 2015]

Given that hash table T with 25 slots that stores 2000 elements, the load factor a for T is _____.

[GATE - 1990]

Match the pairs in the following questions:

(a)	A heap construction	(p)	$\Omega(n \log_{10} n)$
(b)	Constructing Hashtable with linear probing	(q)	$O(n)$
(c)	AVL tree construction	(r)	$O(n^2)$
(d)	Digital trie construction	(s)	$O(n \log_{10} n)$

[GATE - 1996]

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

[GATE - 1999]

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.

[GATE - 2001]

Consider any array representation of an n element binary heap where the elements are stored from index 1 to index n of the array. For the element stored at index i of the array $(i \leq n)$, the index of the parent is

A. $i-1$ B. $\lfloor \frac{i}{2} \rfloor$ C. $\lceil \frac{i}{2} \rceil$ D. $\frac{(i+1)}{2}$

[GATE - 2003]

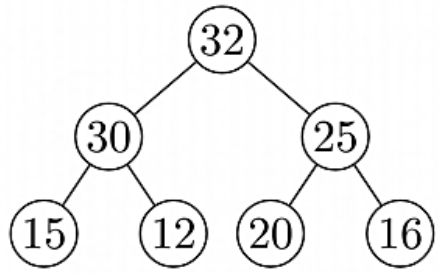
In a min-heap with n elements with the smallest element at the root, the 7^{th} smallest element can be found in time

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

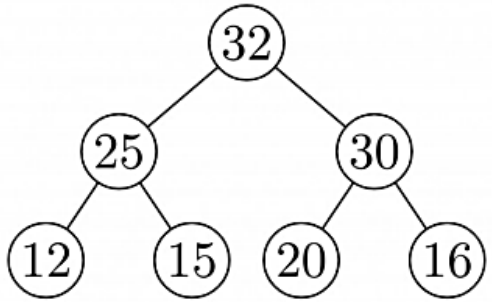
[GATE - 2004]

The elements $32, 15, 20, 30, 12, 25, 16,$ are inserted one by one in the given order into a maxHeap. The resultant maxHeap is

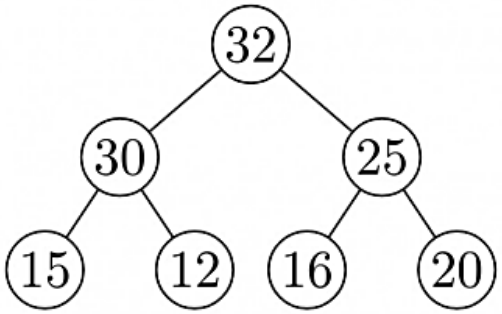
A.



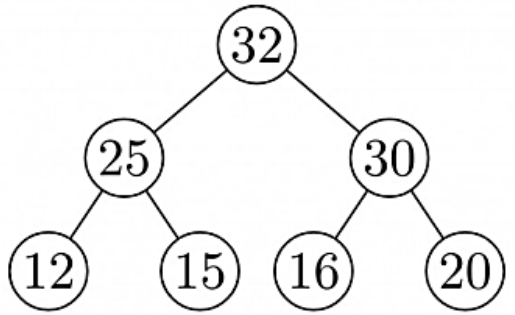
B.



C.



D.



[GATE - 2004]

An array of integers of size n can be converted into a heap by adjusting the heaps rooted at each internal node of the complete binary tree starting at the node $\lfloor (n-1)/2 \rfloor$, and doing this adjustment up to the root node (root node is at index 0) in the order $\lfloor (n-1)/2 \rfloor$, $\lfloor (n-3)/2 \rfloor$, ..., 0. The time required to construct a heap in this manner is

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

[GATE - 2005]

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. $10, 8, 7, 3, 2, 1, 5$

[GATE - 2006]

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. $O(1)$

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. $9,5,6,8,3,1$

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. $10, 8, 6, 9, 7, 2, 3, 4, 1, 5$

[GATE - 2006]

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. $\{23, 17, 14, 7, 13, 10, 1, 12, 5, 7\}$

[GATE - 2006]

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. $O(n \log \log n)$

[GATE - 2007]

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. $\Theta(n \log_2 n)$

[GATE - 2009]

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. $\{25, 14, 12, 13, 10, 8, 16\}$

[GATE - 2009]

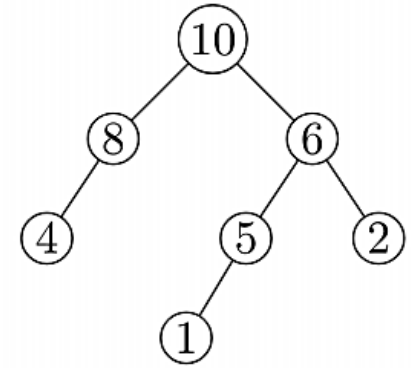
Consider a binary max-heap implemented using an array.
What is the content of the array after two delete operations on $\{25, 14, 16, 13, 10, 8, 12\}$

A. $\{14, 13, 12, 10, 8\}$
B. $\{14, 12, 13, 8, 10\}$
C. $\{14, 13, 8, 12, 10\}$
D. $\{14, 13, 12, 8, 10\}$

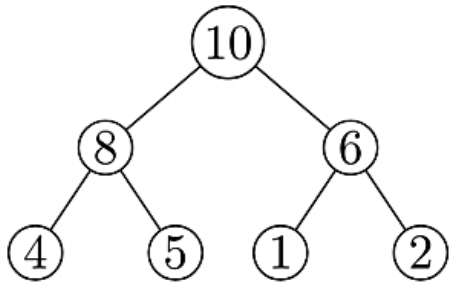
[GATE - 2011]

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?

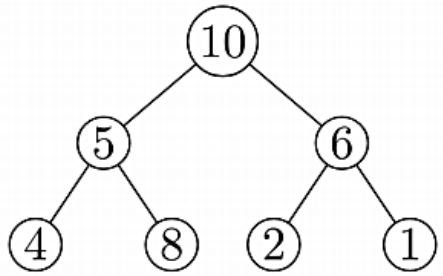
A.



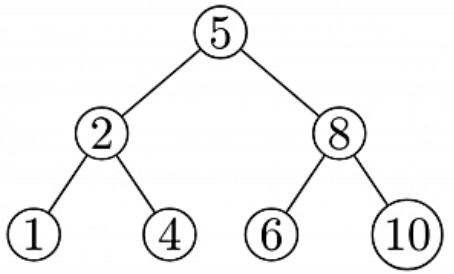
B.



C.



D.



[GATE - 2014]

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. $10, 8, 7, 5, 3, 2, 1$

[GATE - 2015]

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. $40, 35, 20, 10, 15, 16, 17, 8, 4, 30$

[GATE - 2015]

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. $\Omega(n^2)$

[GATE - 2015]

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

[GATE - 2016]

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 $O(2^d)$

[GATE - 2016]

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 _____.

[GATE - 2019]

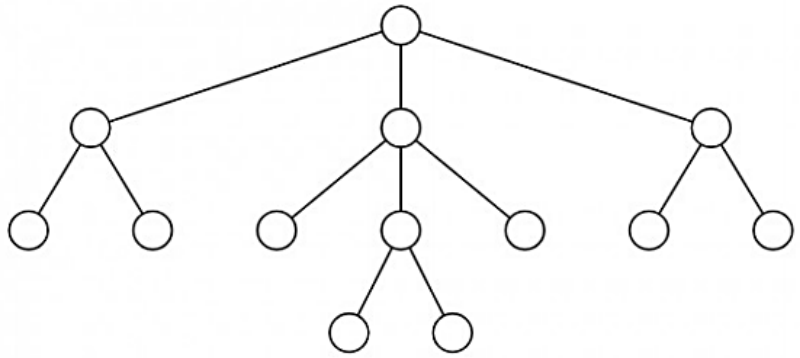
Consider the following statements:

I. The smallest element in a max-heap is always at a leaf node
II. The second largest element in a max-heap is always a child of a root node
III. A max-heap can be constructed from a binary search tree in $\theta(n)$ time
IV. A binary search tree can be constructed from a max-heap in $\theta(n)$ time

Which of te above statements are TRUE?

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

Consider the following tree with 13 nodes.



Suppose the nodes of the tree are randomly assigned distinct labels from $\{1,2,\ldots,13\}$, each permutation being equally likely. What is the probability that the labels form a min-heap (i.e., every node receives the minimum label in its subtree)?

A. $\left(\frac{1}{6!}\right)\left(\frac{1}{3!}\right)^2$

B. $\left(\frac{1}{3!}\right)^2\left(\frac{1}{2!}\right)^3$

C. $\left(\frac{1}{13}\right)\left(\frac{1}{6}\right)\left(\frac{1}{3}\right)^3$

D. $\frac{2}{13}$

E. $\frac{1}{2^{13}}$

[GATE - 1997]

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

[GATE - 1998]

Compute the post fix equivalent of the following expression $3^* \log(x+1) - \frac{a}{2}$

[GATE - 1987]

In a circular linked list oraganisation, insertion of a record involves modification of

A. One pointer.
B. Two pointers.
C. Multiple pointers.
D. No pointer.

[GATE - 1987]

A list of n elements is commonly written as a sequence of n elements enclosed in a pair of square brackets. For example. $[10,20,30]$ is a list of three elements and $[]$ is a nil list. Five functions are defined below:

- $car(l)$ returns the first element of its argument list l;
- $cdr(l)$ returns the list obtained by removing the first element of the argument list l;
- $glue(a,l)$ returns a list m such that $car(m) = a$ and $cdr(m) = l$.
- $f(x,y) \equiv$ if $x = []$ then y
  else $glue(car(x), f(cdr(x),y))$;
- $g(x) \equiv$ if $x = []$ then $[]$
  else $f(g(cdr(x)), glue(car(x),[]))$

What do the following compute?

(a) $f([32,16,8],[9,11,12])$

(b) $g([5,1,8,9])$

[GATE - 1993]

Consider a singly linked list having n nodes. The data items $d_1, d_2, \ldots d_n$ are stored in these n nodes. Let X be a pointer to the j^{th} node $(1 \leq j \leq n)$ in which d_j is stored. A new data item d stored in node with address Y is to be inserted. Give an algorithm to insert d into the list to obtain a list having items $d_1, d_2, \ldots, d_j, d, \ldots, d_n$ in order without using the header.

[GATE - 1994]

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

[GATE - 1995]

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

[GATE - 1997]

The concatenation of two lists is to be performed on $O(1)$ time. Which of the following implementations of a list should be used?

A. Singly linked list

B. Doubly linked list

C. Circular doubly linked list

D. Array implementation of list

[GATE - 1997]

Consider the following piece of 'C' code fragment that removes duplicates from an ordered list of integers.

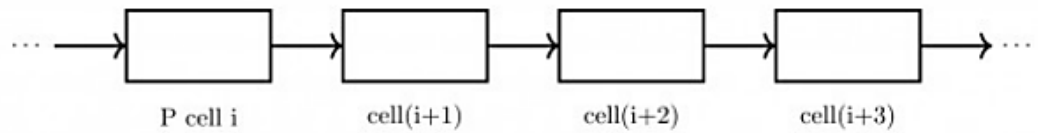
```
Node *remove-duplicates (Node* head, int *j)
{
    Node *t1, *t2; *j=0;
    t1 = head;
    if (t1! = NULL)
        t2 = t1 ->next;
    else return head;
    *j = 1;
    if(t2 == NULL) return head;
    while (t2 != NULL)
    {
        if (t1.val != t2.val) ----------------> (S1)
        {
            (*j)++;
            t1 -> next = t2;
            t1 = t2; -----> (S2)
        }
            t2 = t2 ->next;
    }
    t1 -> next = NULL;
    return head;
}
```

Assume the list contains n elements ($n \geq 2$) in the following questions.

a. How many times is the comparison in statement $S1$ made?
b. What is the minimum and the maximum number of times statements marked $S2$ get executed?
c. What is the significance of the value in the integer pointed to by j when the function completes?

[GATE - 1998]

a. Let p be a pointer as shown in the figure in a single linked list.



What do the following assignment statements achieve?

```
q: = p -> next
p -> next:= q -> next
q -> next:=(q -> next) -> next
(p -> next) -> next:= q
```

[GATE - 1999]

Write a constant time algorithm to insert a node with data D just before the node with address p of a singly linked list.

[GATE - 2002]

In the worst case, the number of comparisons needed to search a single linked list of length n for a given element is

A. $\log n$ B. $\frac{n}{2}$ C. $\log_2 n - 1$ D. n

[GATE - 2003]

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

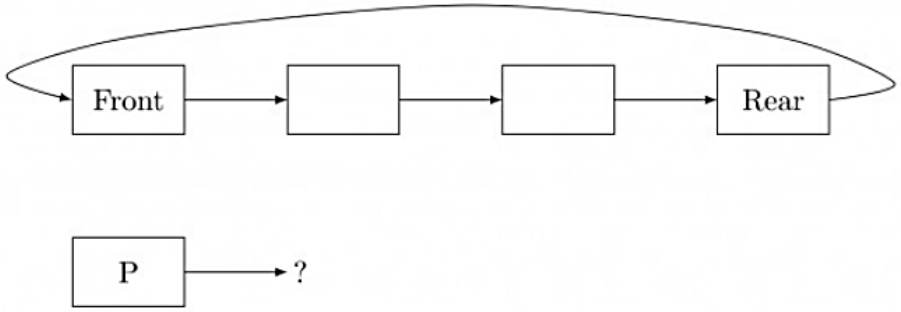
B. the elements in the list are sorted in non-decreasing order of data value

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

[GATE - 2004]

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

[GATE - 2004]

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

[GATE - 2004]

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. $O(1)$

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. $2,3,4,5,6,7,1$

[GATE - 2008]

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. $2, 3, 4, 5, 6, 7, 1$

[GATE - 2010]

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. $q = NULL; p \rightarrow next = head; head = p$;
B. $q \rightarrow next = NULL; head = p; p \rightarrow next = head$;
C. $head = p; p \rightarrow next = q; q \rightarrow next = NULL$;
D. $q \rightarrow next = NULL; p \rightarrow next = head; head = p$;

[GATE - 2016]

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. $\Theta(N^2 \log N)$

[GATE - 2017]

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.

[GATE - 1997]

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

[GATE - 1992]

Suggest a data structure for representing a subset S of integers from 1 to n. Following operations on the set S are to be performed in constant time (independent of cardinality of S).

i.	MEMBER (X) :	Check whether X is in the set S or not
ii.	FIND-ONE (S) :	If S is not empty, return one element of the set S (any arbitrary element will do)
iii.	ADD (X) :	Add integer X to set S
ii.	DELETE (X) :	Delete integer X from S

Give pictorial examples of your data structure. Give routines for these operations in an English like language. You may assume that the data structure has been suitable initialized. Clearly state your assumptions regarding initialization.

[GATE - 1994]

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.

[GATE - 1996]

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

[GATE - 2001]

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

[GATE - 2006]

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 $2m \leq y \leq 2n$

[GATE - 2007]

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

[GATE - 2012]

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: $REAR == FRONT$

[GATE - 2013]

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. $\Theta(n^2)$

[GATE - 2016]

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 $\Omega(\log n)$

[GATE - 2016]

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 _____.

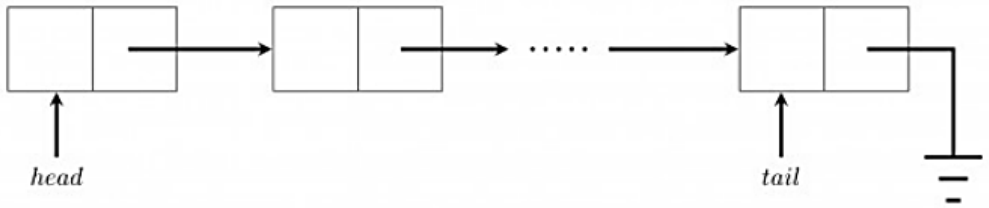
[GATE - 2017]

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).

A queue is implemented using a non-circular singly linked list. The queue has a head pointer and a tail pointer, as shown in the figure. Let n denote the number of nodes in the queue. Let 'enqueue' be implemented by inserting a new node at the head, and 'dequeue' be implemented by deletion of a node from the tail.



Which one of the following is the time complexity of the most time-efficient implementation of 'enqueue' and 'dequeue, respectively, for this data structure?

A. $\Theta(1), \Theta(1)$
B. $\Theta(1), \Theta(n)$
C. $\Theta(n), \Theta(1)$
D. $\Theta(n), \Theta(n)$

[GATE - 1991]

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

[GATE - 1994]

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. $5, 4, 3, 1, 2$

[GATE - 1995]

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. $A + *BCD/F * DE + +$

[GATE - 2000]

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

[GATE - 2003]

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. $Y+2X$

[GATE - 2004]

A single array $A[1..MAXSIZE]$ is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables $top1$ and $top2$ $(top < top2)$ point to the location of the topmost element in each of the stacks. If the space is to be used efficiently, the condition for "stack full" is

A. $(top1 = MAXSIZE/2)$ and $(top2 = MAXSIZE/2 + 1)$

B. $top1 + top2 = MAXSIZE$

C. $(top1 = MAXSIZE/2)$ or $(top2 = MAXSIZE)$

D. $top1 = top2 - 1$

[GATE - 2004]

Assume that the operators $+, -, \times$ are left associative and $\hat{}$ is right associative. The order of precedence (from highest to lowest) is $\hat{}, \times, +, -$. The postfix expression corresponding to the infix expression $a + b \times c - d \hat{} e \hat{} f$ is

A. $abc \times + def \hat{} \hat{} -$

B. $abc \times + de \hat{} f \hat{} -$

C. $ab + c \times d - e \hat{} f \hat{}$

D. $- + a \times bc \hat{} \hat{} def$

[GATE - 2004]

The best data structure to check whether an arithmetic expression has balanced parentheses is a

A. queue B. stack C. tree D. list

[GATE - 2004]

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. $cabd$

[GATE - 2005]

A function f defined on stacks of integers satisfies the following properties. $f(\emptyset) = 0$ and $f(push(S,i)) = max(f(S),0) + i$ for all stacks S and integers i.

If a stack S contains the integers $2,-3,2,-1,2$ in order from bottom to top, what is $f(S)$?

A. 6 B. 4 C. 3 D. 2

[GATE - 2007]

The following postfix expression with single digit operands is evaluated using a stack:

8 2 3 ^ / 2 3 * +

Note that ^ 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

[GATE - 2007]

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

[GATE - 2014]

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.

[GATE - 2015]

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 _____.

[GATE - 2015]

The result evaluating the postfix expression $10\ 5 + 60\ 6 / * 8 -$ is

A. 284 B. 213 C. 142 D. 71

[GATE - 2017]

We have an implementation that supports the following operations on a stack (in the instructions below, **s** is the name of the stack).

- **isempty(s)** : returns **True** if **s** is empty, and **False** otherwise.
- **top(s)** : returns the top element of the stack, but does not pop the stack; returns **null** if the stack is empty.
- **push(s,x)** : places **x** on top of the stack.
- **pop(s)** : pops the stack; does nothing if **s** is empty.

Consider the following code:

```
pop_ray_pop(x):
    s=empty
    for i=1 to length(x):
        if (x[i] == '('):
            push(s, x[i])
        else:
            while (top(s)=='('):
                pop(s)
            end while
            push(s, ')')
        end if
    end for
    while not isempty(s):
        print top(s)
        pop(s)
    end while
```

What is the output of this program when

```
pop_ray_pop("((((()(((()((((")
```

is executed?

A. ((((B.))) ((((C.))) D. ((((())) E. ()()

[GATE - 1990]

Consider the height-balanced tree T_t with values stored at only the leaf nodes, shown in Fig.4.

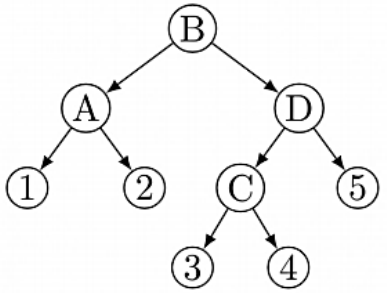


Fig.4

(i) Show how to merge to the tree, T_1 elements from tree T_2 shown in Fig.5 using node D of tree T_1.

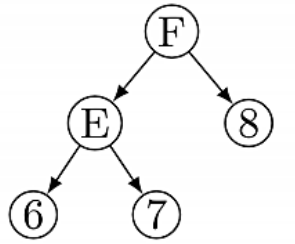


Fig.5

(ii) What is the time complexity of a merge operation of balanced trees T_1 and T_2 where T_1 and T_2 are of height h_1 and h_2 respectively, assuming that rotation schemes are given. Give reasons.

[GATE - 1992]

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

[GATE - 1994]

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 $2(n-1)$.

[GATE - 1998]

Which of the following statements is false?

A. A tree with a n nodes has $(n-1)$ edges
B. A labeled rooted binary tree can be uniquely constructed given its postorder and preorder traversal results.
C. A complete binary tree with n internal nodes has $(n+1)$ leaves.
D. The maximum number of nodes in a binary tree of height h is $2^{h+1} - 1$

[GATE - 1998]

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. $x(n+1)$

[GATE - 1998]

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?

[GATE - 2002]

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. $\frac{(2n+1)}{3}$

[GATE - 2004]

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

[GATE - 2005]

In a complete k-ary tree, every internal node has exactly k children. The number of leaves in such a tree with n internal node is:

A. nk B. $(n-1)k+1$ C. $n(k-1)+1$ D. $n(k-1)$

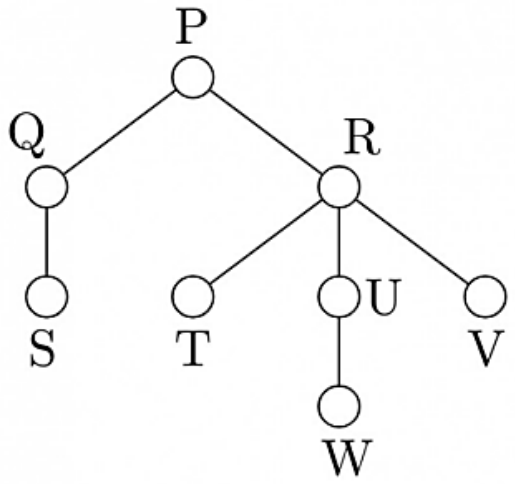
[GATE - 2007]

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

[GATE - 2014]

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. $SQPTRUWV$

[GATE - 2014]

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

[GATE - 2017]

Let T be a tree with 10 vertices. The sum of the degrees of all the vertices in T is _____

[GATE - 2012]

Let T be a tree of n nodes. Consider the following algorithm, that constructs a sequence of leaves $u_1, u_2 \ldots$. Let u_1 be some leaf of tree. Let u_2 be a leaf that is farthest from u_1. Let u_3 be the leaf that is farthest from u_2, and, in general, let u_{i+1} be a leaf of T that is farthest from u_i (if there are many choices for u_{i+1}, pick one arbitrarily). The algorithm stops when some u_i is visited again. What can u say about the distance between u_i and u_{i+1}, as $i = 1, 2, \ldots$?

A. For some trees, the distance strictly reduces in each step.
B. For some trees, the distance increases initially and then decreases.
C. For all trees, the path connecting u_2 and u_3 is a longest path in the tree.
D. For some trees, the distance reduces initially, but then stays constant.
E. For the same tree, the distance between the last two vertices visited can be different, based on the choice of the first leaf u_1.