Q.1) Max Marks: 1

In a hash function h(k)= key mod m ,where m is the size of the table. Then choose the best possible m value in 3, 7, 9, 11 and then find the number of collisions while inserting keys into hash table. Keys: 4, 12, 14, 18, 23, 44, 59, 87.

A 1

B 2 Correct Option

Solution: (B)

Solution: Ans is (B)2

Prime number is best choice but don't choose m values in terms of power of 2 because –

If m=2^k Then h(key) =LSB (k) bits

So here, choosing 11 is the best choice. H(k) = key mod 11

Then inserting keys into hash table

0	44
1	12
2	23
3	14
4	4
5	59
6	
7	18
8	
9	
10	87

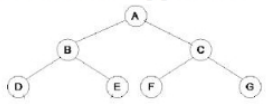
4 mod 11 = 4
12 mod 11 =1
14 mod 11 = 3
18 mod 11 = 7
23 mod 11 = 1 : collision
44 mod 11 = 0
59 mod 11 = 4 : collision
87 mod 11 = 10
So there are two collisions while inserting keys into hash table.

C 3

D 4

Q.2) Max Marks: 1

Consider the following given tree.



Choose the correct option of which elements are changing their position in level order traversal from preorder traversal in the above given tree.

A A, B, C

B B, C, D

C C, D, E Correct Option

Solution: (C)
Solution: Ans is (C)

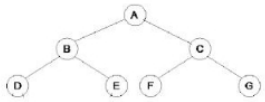
Pre-order traversal : (root, left, right) : A, B, D, E, C, F, G
Level order traversal : (level by level): A, B, C, D, E, F, G

We can see here, four elements which are at the same position in both the traversal.
Those are A, B, F, G. So, here node elements C, D, E are changing their positions in level order traversal from preorder traversal.

D D, E, F

Q.3) Max Marks: 1

Binary search tree with nodes (A, B, C, D, E, F, G) is given below.



Find pre-order and level order traversal of the given tree and arrange these elements in order of their traversal then find out which elements are at the same position in both the traversal.

A A, B, C, D

B A, B, D, E

C A, B, E, F

D A, B, F, G — Correct Option

Solution: (D)
Solution: Ans is (D)
Pre-order traversal : (root, left, right) : A, B, D, E, C, F, G
Level order traversal : (level by level): A, B, C, D, E, F, G
We can see here, four elements which are at the same position in both the traversal.
Those are A, B, F, G. so answer will be (D).

Q.4) Max Marks: 1

The keys 12, 18, 13, 23 and 15 are inserted into an initially empty hash table of length 10 using open addressing with the hash function h (k) = k mod 10 and linear probing. Then what will be the expected number of probes possible in a successful search of any open addressing technique.

A 1.5290

B 1.7529

C 1.3862 — Correct Option

Solution: (C)

Solution: Ans is (C)

The expected number of probes possible in a successful search of any open addressing technique = $(1/\propto) * \ln 1/(1-\propto)$ where $\propto$ is the load factor with the value $\propto = n/m$, where n is the number of keys and m is the size of the hash table.

In this question n is 5 because there are 5 keys and hash table length is 10 so m value is 10.
Therefore $\propto = 5/10 = 0.5$

Then, the expected number of probes possible in a successful search of any open addressing technique = $(1/\propto) * \ln 1/(1-\propto)$
$= (1/0.5) * \ln (1/1-0.5)$
$= 2 * \ln (1/.5)$
$= 2 * \ln 2 = 2 * 0.6931 = 1.3862$

D 1.9674

Q.5) Max Marks: 1

If there is a rooted tree with total n nodes, in which each node is having 0 or 3 children. Answer the following questions.
Choose the correct option for getting No. of leaf nodes in the tree

A n/2

B (n-1) /3

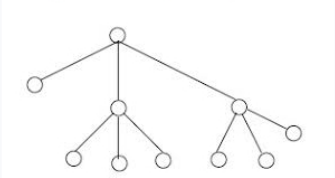
C (n-1) /2

D (2n+1)/3 — Correct Option

Solution: (D)

Solution: Ans is (D). (2n+1)/3

We can get the answer by drawing a tree with the the given condition. As it is given that, there is a tree in which every node has either 0 or 3 nodes. So tree can be like this:



In this tree, total node => n= 10, leaf nodes = 7, internal nodes = 3
and if we check with options then
n-1/3 = 10-1/3 = 9/3 = 3 not correct
n-1/2 = 10-1/2 = 9/2 = 4.5 not correct
n/2 = 10/2 = 5 not correct
(2n+1)/3 = 21/3 = 7 correct.
So option C is the answer.

Q.6) Max Marks: 1

Build heap operation is used for constructing a min-heap or max-heap from the given elements. Then find out upper bound of time complexity of constructing a heap using build heap operation.

A **O (n)** Correct Option

Solution: (A)

Solution: Ans is (A). O(n)
Following is an algorithm for building a heap of an input array of A
Build_heap(A)
Heapsize = size(A);
For i= floor (heapsize/2)downto 1
Do heapify(A,i);
End for
END
Although the worst case complexity looks like O (n log n), upper bound of time complexity is O (n).

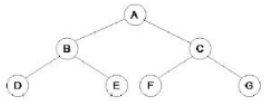
B **O (n^2)**

C **O (log n)**

D **O (n log n)**

Q.7) Max Marks: 1

Binary search tree with nodes (A, B, C, D, E, F, G) is given below.



Find out the correct option of getting exact mid element of pre-order, in-order and post -order traversal of the above tree.

A **E, A, F** Correct Option

Solution: (A)

Solution: Ans is (A) E, A F

Pre-order, In-order and Post-order are the different types of traversal of binary tree.

Pre-order: (root, left, right) A, B, D, E, C, F, G

In-order: (left, root, right) D, B, E, A, F, C, G

Post-order:(left, right, root) D, E, B, F, G, C, A

Finding mid element is very simple as there are odd number of nodes in the given tree. so mid of preorder, inorder and postorder are E, A and F

B **B, A, C**

C **D, A, G**

D **A, D, D**

Q.8) Max Marks: 1

What is the common in between pre-order, post-order and in-order traversals of the binary trees.

A Root is always visited before left subtree.

B Root is always visited after right subtree.

C Left sub-tree always visited before right subtree. Correct Option

Solution: (C)
Solution: Ans is (C)
As we know, different traversal sequences of binary search tree are as follows:
Pre-order: root ,left subtree, right subtree
Post-order: left subtree, right subtree, root
In-order: left subtree, root, right subtree
So first statement root is always visited before left subtree in only pre-order traversal.
Second statement, root is always visited after right subtree in post order traversal only.
Third statement, left subtree is always visited before right subtree in all the traversals.
Fourth statement, right subtree is never visited before left subtree.

D Right subtree is always visited before left subtree

Q.9) Max Marks: 1

In a hash function h(k)= key mod m ,where m is the size of the table. Then choose the best possible m value in 3, 7, 9, 11 and then find the number of collisions while inserting keys into hash table. Keys: 4, 12, 14, 18, 23, 44, 59, 87.What is the value of the equation if empty slots are denoted by E and occupied slots are denoted by O and number of collision are denoted by C equation is 2C *E + O.

A 12

B 20 Correct Option

Solution: (B)
Solution: Ans is (B) 20
Prime number is best choice but don't choose m values in terms of power of 2
because –
If m=2^k Then h(key) =LSB (k) bits
So here, choosing 11 is the best choice. H(k) = key mod 11
Then inserting keys into hash table:

0	44
1	12
2	23
3	14
4	4
5	59
6	
7	18
8	
9	
10	87

4 mod 11 = 4
12 mod 11 =1
14 mod 11 = 3
18 mod 11 =7
23 mod 11 = 1
44 mod 11 = 0
59 mod 11 = 4
87 mod 11 = 10

So there are two collisions while inserting keys into hash table.
Values of variables are: C =2, E=3, O=8
Then,
= 2C *E + O
= 2*2 * 3 +8
= 20

C 32

D 45

Q.10) Max Marks: 1

If there is a rooted tree with total n nodes, in which each node is having 0 or 3 children. Choose the correct option for getting No. of internal nodes in the given tree.

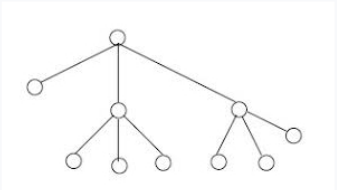
A floor [n/2]

B floor [(n-1) /3] Correct Option

Solution: (B)

Solution: Ans is (B). floor [(n-1) /3]

By drawing a ternary tree with given conditions we can find out the correct output of the given question. As it is given that, there is a tree in which every node has either 0 or 3 nodes. So tree can be like this:



In this tree, total node => n= 10, leaf nodes = 7, internal nodes = 3

and if we check with options then

floor [n/2] = floor [10/2] = 5 not correct

floor [(n-1) /3] = floor [(10-1) /3] = 3 correct

floor [(n-1)/2] = floor [(10-1)/2] 4 not correct

floor [(2n+1)/3] = floor [(2*10+1)/3] = 7 not correct

So option B is the correct answer.

C floor [(n-1) /2]

D floor [(2n+1)/3]

Q.11) Max Marks: 2

What are the minimum number of nodes in an AVL tree of depth 9 if we are considering root at depth 1.

A 143

B 88 Correct Option

Solution: (B)

Solution: Ans is (B). 88

Here, we are just making confusion by writing depth and starting root at level 1, so here total height of avl tree will be 8 so we need to calculate nodes for this height. We can use formula for that as we know that min nodes for height 5 and height 6 are 20 and 33.

1, if h= 0 : level 1: h(0) => 1 node

2, if h=1 : level 2: h(1) => 2 nodes

Min no. of nodes (h) = min. no. of nodes(h-1) + min. no. of nodes(h-2) + 1

level 3: h(2) => 4 nodes

level 4: h(3) => 7 nodes

level 5: h(4) => 12 nodes

level 6: h(5) => 20 nodes

h(7) = h(6) + h(5) + 1 = 33+20+1 = 54

h(8) = h(7) + h(6) + 1 = 54+33+1 = 88

So, min nodes for depth 9, AVL tree while considering root at depth 1 instead of 0 so here at h=8 it is calculated and min nodes will be 88.

C 54

D 33

Q.12) Max Marks: 2

When elements of priority queue are used for creating max-heap. And initially there are 6 elements in this. 20, 16, 10, 6, 4, 2. Then after constructing max heap find. No of swaps after inserting 2 more elements '3' and '9' and then deletion of root element.

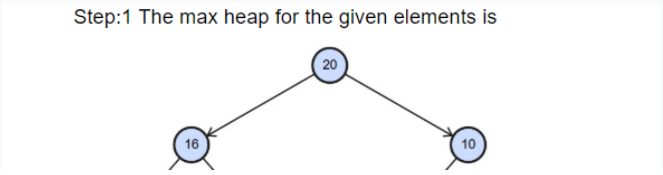
A 1

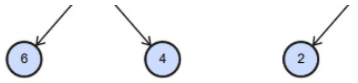
B 2

C 3

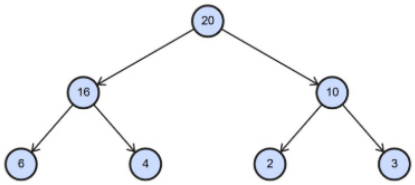
D 4 Correct Option

Solution: (D)

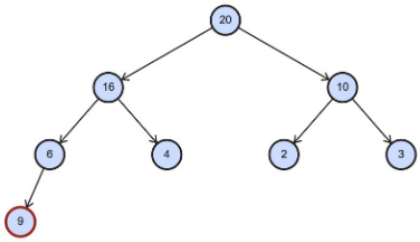




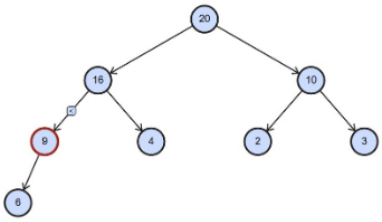
Step: 2 Insert 3 in the heap



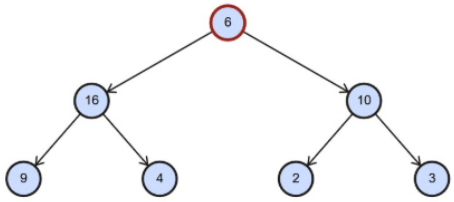
Step: 3 insert 9



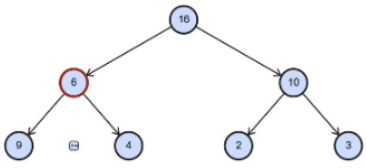
Swap 6 and 9



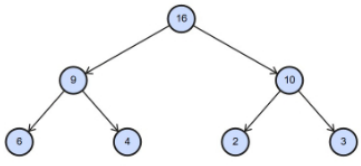
After deleting the root node the heap is



Swap 6 and 16



Swap 6 and 9



Please note that for deletion of the root node we need to swap the leaf node with root node. There after as mentioned we will have 2 swaps.

Total number of swaps are 1(insertion) + 1(deletion) + 2(swaps after deletion) = 4.

Q.13) Max Marks: 2

Consider an AVL tree or height balanaced tree with n nodes. Then calculate the toal time utilized for performing the following given steps:
step 1: delete the root node.
step 2: finding replacement for root.
step 3: balancing the tree.

A O(1)

B O(log n) Correct Option

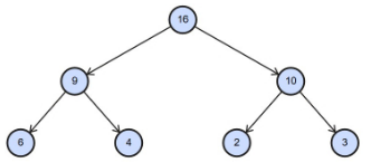
Solution: (B)
Solution: Ans is (B). O(log n)
step 1: delete the root node. // As deleting root node takes O(1) time.
step 2: finding replacement for root. // while finding replacement mean we are traversing tree upto depth so it takes O(log n) time.
step 3: balancing the tree. // updating height and getting balanced factor also takes constant time.
so total time will be O(log n).

C O(n)

D O(nlogn)

Q.14) Max Marks: 2

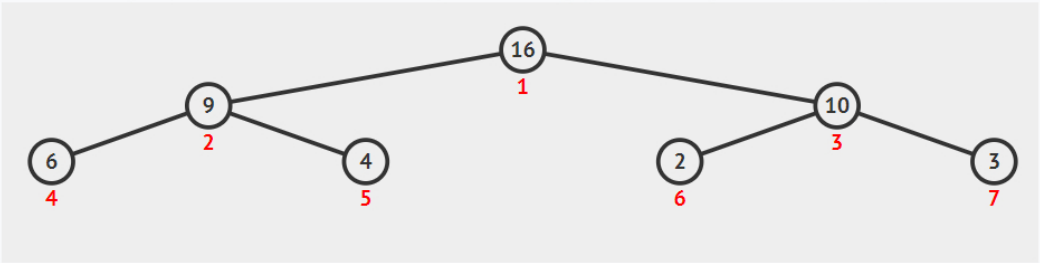
Find the preorder traversal of the given max heap.



A 10, 16, 9, 6, 4, 2, 3

B 16, 9, 6, 4,10, 2, 3 Correct Option

Solution: (B)
Solution: Ans is (B) 16, 9, 6, 4, 10, 2, 3
Here final max-heap will be like 16 as root element and 9 as left parent and 10 as right parent and then 6, 4 are children of 9 and 2, 3 are children of 10.



So its pre-order traversal will be like this 16, 9, 6, 4, 10, 2, 3.

C 16, 4, 2, 3, 10, 6, 9

D 16, 10, 4, 9, 6, 2, 3

Q.15) Max Marks: 2

find the sum of the total nodes of AVL tree of height 3 with minimum nodes and maximum nodes.

A 7

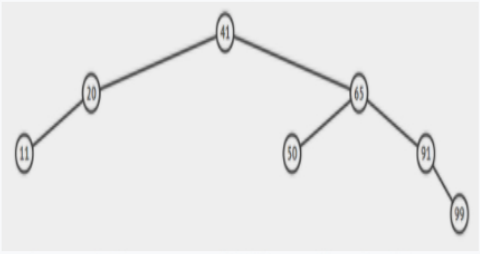
B 15

C 22 Correct Option

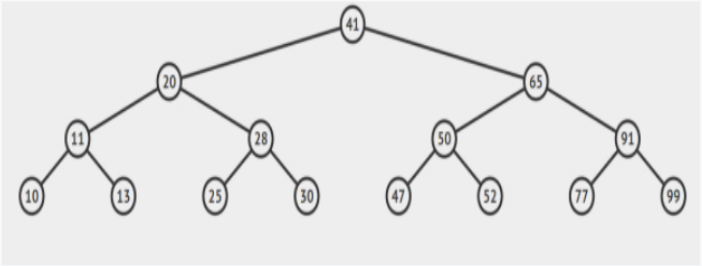
Solution: (C)

Solution: (C). 22

Minimum nodes in an AVL tree of height 3 = 7 (for satisfying balancing)



Maximum nodes in an AVL tree of height 3 = 15 (complete binary tree which is balance)



so sum is 7+15 = 22

D 30

close