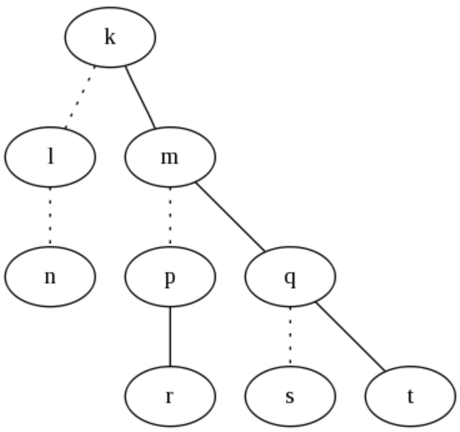
SET – IC

Binary Trees

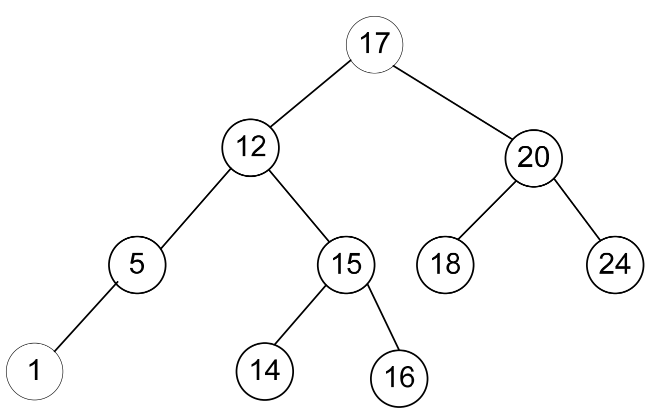
<table>
<tr>
<td>1.</td>
<td>Construct a binary tree whose preorder traversal is K L N M P R Q S T and in order traversal is N L K P R M S Q T

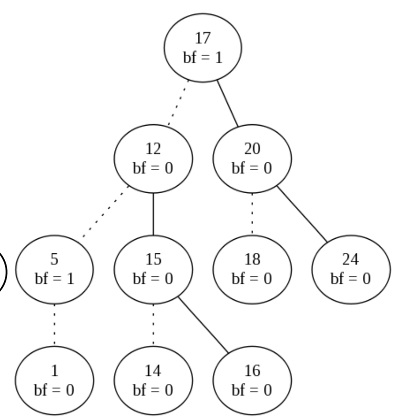

</td>
</tr>
<tr>
<td>2.</td>
<td>(i) Define the height of a binary tree or subtree and also define a height balanced (AVL) tree.

The height of a tree or a sub tree is defined as the length of the longest path from the root node to the leaf.

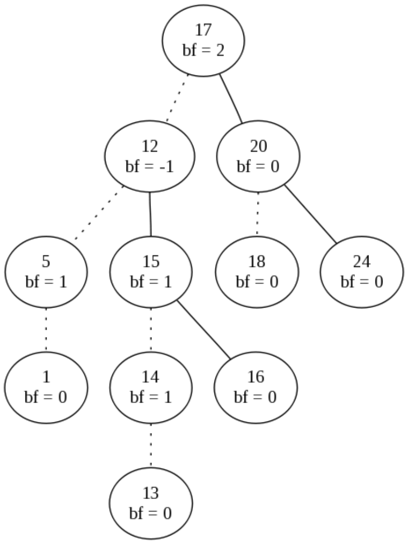
A tree is said to be height balanced if all the nodes are having a balance factor -1,0 or 1. The balance factor is the height of the left sub tree minus height of the right sub tree

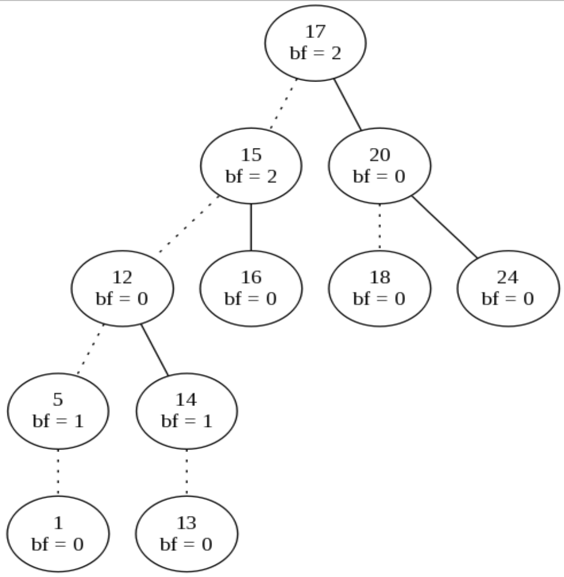
(ii) Mark the balance factor of each mode on the tree given in fig 7.1 and state whether it is height-balanced

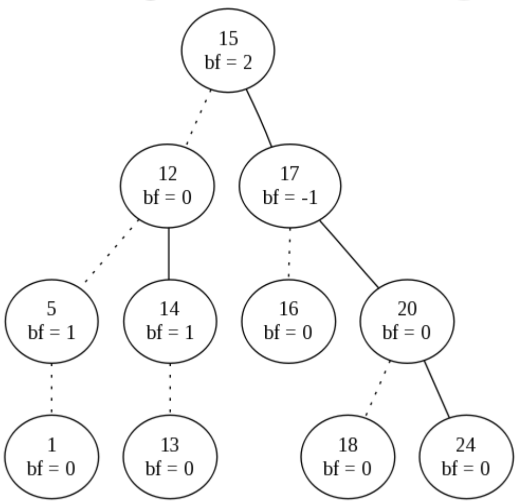




(iii) Into the same tree given in 7(ii) above, insert the integer 13 and show the new balance factors that would arise if the tree is no rebalanced. Finally, carry the required rebalancing of the tree and show the new tree with the balance factors on each mode.</td>
</tr>
</table>







3. The maximum number of nodes in a binary tree of level k, $k \geq 1$ is

(A) $2^k + 1$
(B) $2^k - 1$
(C) 2^{k-1}
(D) $2^{k-1} - 1$

[B]
Here k starts from 1

For the root node k=1, no. of nodes in root node is 1

For k=2, no.of nodes in the binary tree is 3,

For k=3,no. of nodes is 7

For k=4, no of nodes is 15.

………

………

So for any value k it is 2^k – 1

4. Consider the height balanced tree T_1, with values stored at only the leaf nodes, shown in Fig. 4

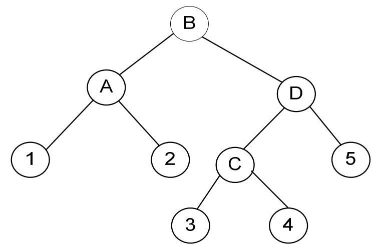


Fig 4

(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

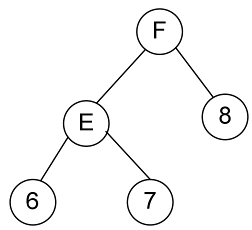


Fig 5

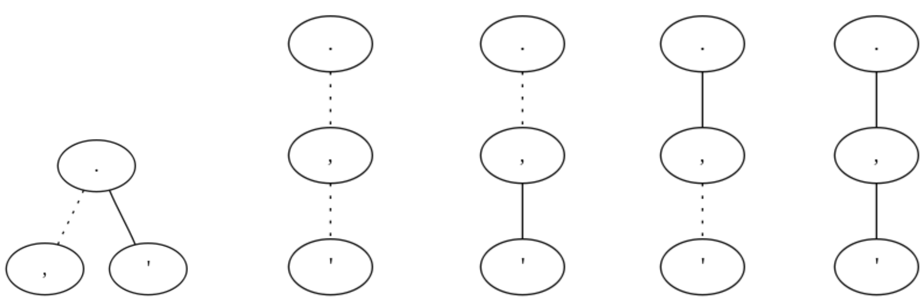
(ii) What is the time complexity of a merge operation on balanced trees T_1 and T_2 where T_1 and T_2 are of height h_1 and h_2 respectively, assuming rotation scheme are given. Give reasons

5. The number of different ordered trees with 3 nodes labeled Y, Y, Z are

(A) 16
(B) 8
(C) 12
(D) 24

Ans: 15

There are 5 tree unordered tree structures possible.



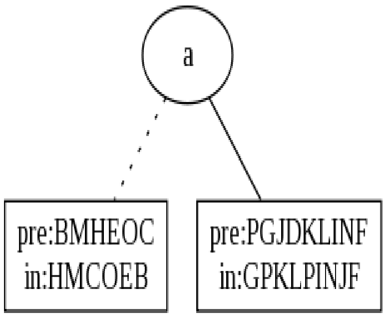
(a) (b) (c) (d) (e)

However, we want to order the nodes based on Y, Y and Z, there are 3 possibilities associated with every tree structure.

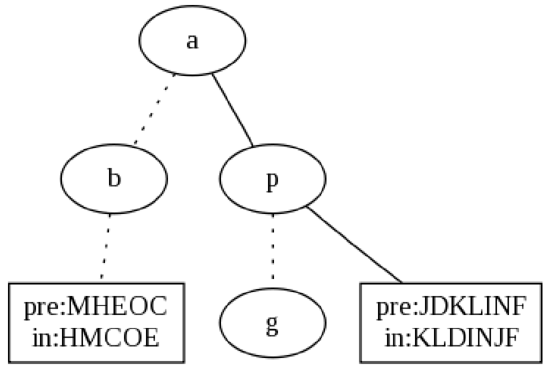
Hence, 5 * 3 = 15

6. Construct a binary tree whose preorder and inorder sequences are A B M H E O C P G J D K L I N F and H M C O E B A G P K L

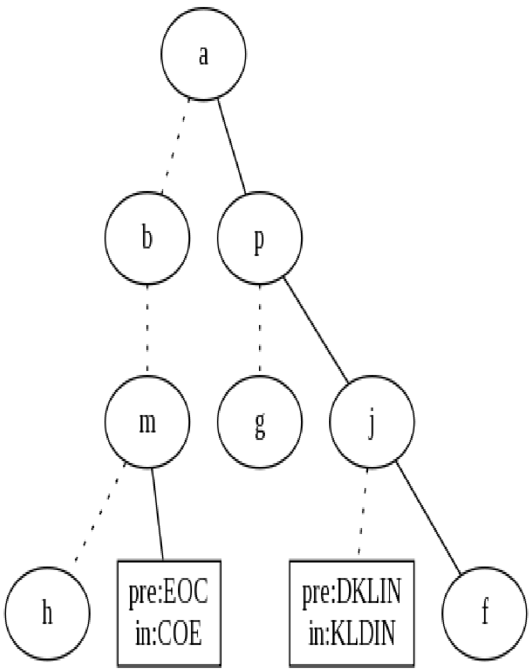
D I N J F respectively, where A, B, C, D, E, ………. are the labels of the tree nodes. Is it unique?



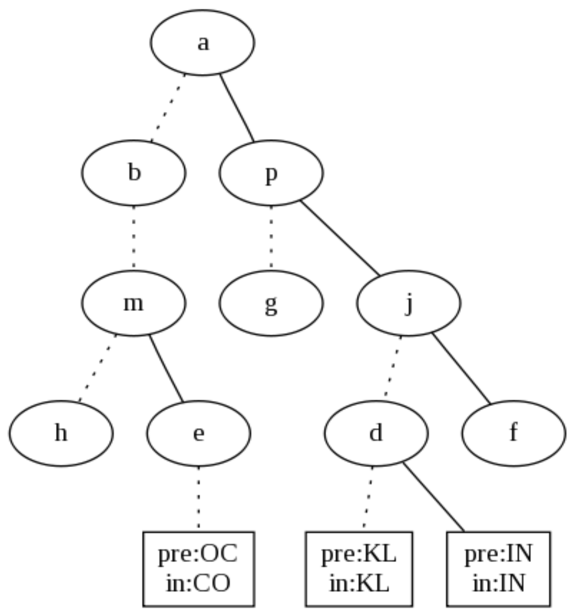
(a)



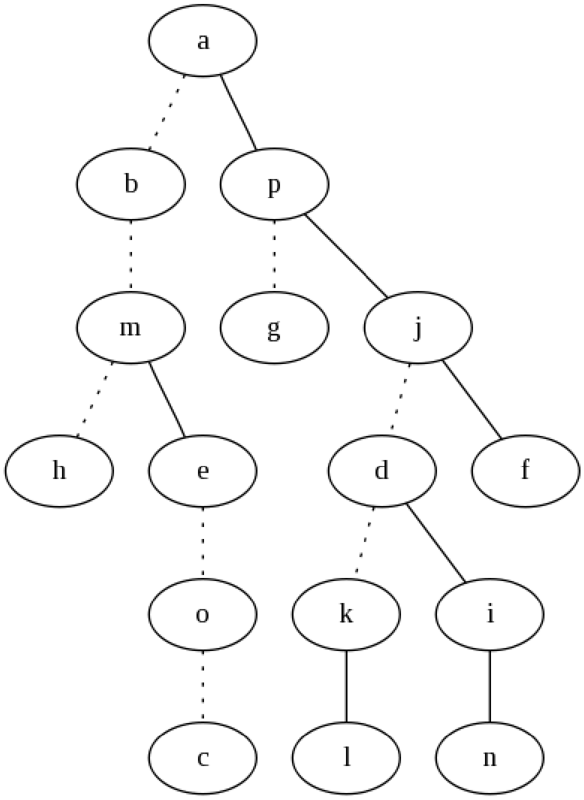
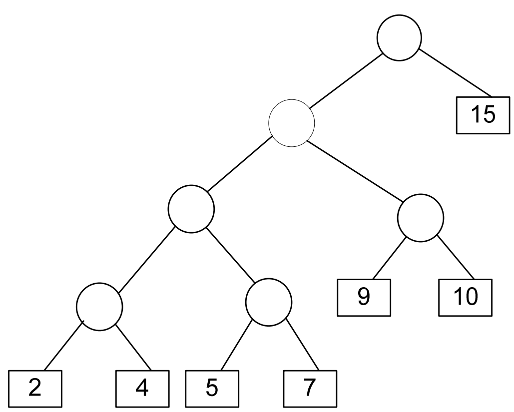
(b)



(c)



(d)

	 (e)
7.	The weighted external path length of the binary tree in Fig. 2 is ______________.  **Fig 2** weighted external path = (2*4) + (4*4) + (5*4) + (7*4) + (9*3) + (10*3) + (15*1) = 8 + 16 + 20 + 28 + 27 + 30 + 15 =144
8.	If the binary tree in Fig. 3 is traversed in inorder, then the order in which the nodes will be visited is ______________________

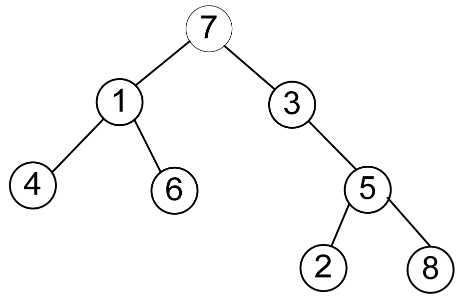


Fig 3

4 1 6 7 3 2 5 8

The in order sequence is left, root, right .

9. Consider the binary tree in Fig. 7:

(a) What structure is represented by the binary tree?
(b) Give the different steps for deleting the node with key 5 so that the structure is preserved.
(c) Outline a procedure in the pseudo-code to delete an arbitrary node from such a binary tree with n nodes that preserves the structure. What is the worst case complexity of your procedure?

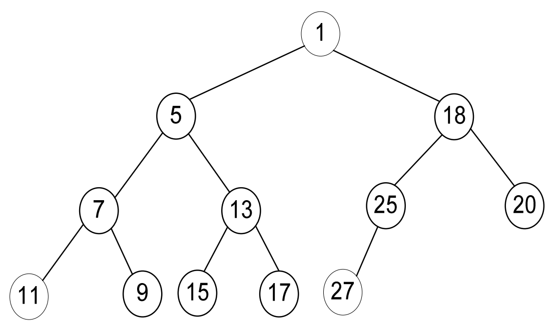


Fig. 7

(a)Min-Heap

(b)

i. Swap(27,5)

ii. Delete 5

iii. Minheapify

(c)

Alogithm delMinHeap()

Step 1: let I th node is to be deleted.

Step 2: copy last element to this location

Step 3: remove last element of array

Step 4: perform Min-Heapify on I th node

Algorithm Min-Heapify()

A-array, i-ith node

Step 1: if 2*i<=size[A] and A[2i]<A[i] /*left child*/

Then smallest=2i

Else smallest =i

Step 2: if 2i+1 <=size[A] and A[2i+1] < A[smallest]

Then smallest=2i+1

Step3: If smallest != i

Then swap A[i] and A[smallest]

Step 3: Min-Heapify(A,smallest)

10. if a tree has n_1 nodes of degree 1, n_2 nodes of degree 2, …. n_m nodes of degree m, give a formula for the number of terminal nodes n0 of the tree in terms of n_1, n_2, …., n_m.

Sum of degree of all the nodes will be-$1*n_1+2*n_2.....m*n_m$-------(1)

We know sum of degree of all the nodes is equal to number of edges.-----(2)

And number of edges= number of total node-1.-------(3)

Total number of nodes= $n_0+n_1+n_2....n_m$.-----------(4)

From 1,2,3,4 we have

$1*n_1+2*n_2......m*n_m= (n_0+n_1+n_2......n_m)-1$

We get

Ans: $n_0 = 1+[1*n_2(2-1) +2*n_{3..}...+(m-1)*n_m]$

11. A 2-3 tree is a tree 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

A and D

At leaf-level:

1. group the nodes into 3 sets of 3 …… it gives 4 internal nodes

2. group the nodes into 3 sets of 2 and 1 set of 3 …… it gives 7 internal nodes

12. A *K*-ary tree is such that every node has either *K* sons or no sons. If *L* and *I* are the number of leaves and internal nodes respectively, then express *L* in terms of *K* and *I*.

If there is only one 1 internal node then it have k leaf nodes.

Adding one more internal node means making 1 leaf node as internal and adding k leaf nodes.

So if there are I internal nodes it have **I*k-(I-1)** leaf nodes

L=K*(I-1)+1

13. A 3-ary tree is a tree in which every internal node has exactly three children. Use the induction to prove that the number of leaves in a 3-ary tree with n internal nodes is 2(n-1) + 3.

When,

$n = 0 \rightarrow l = 2(0 - 1) + 3 = 1$

$n = 1 \rightarrow l = 2(1 - 1) + 3 = 3$

In this way let the rule be true for $n = k$

Hence $l = 2(k - 1) + 3$

For $n = k + 1$

However, from observation: $l(k + 1) = l(k) + 3 – 1$

i.e. $l(k + 1) = 2(k - 1) + 3 + 3 - 1$

$= 2k + 3$

$= 2((k + 1)-1) + 3$

Thus, by induction, the rule is true for all k.

14. 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)

[A]

Degree of internal nodes (except root) = n + 1

Degree of root = n

Degree of leaf nodes = 1

Also,

No of edges = no. of nodes – 1

But n = x + L

	Thus, no. of edges = x + L -1 From graph theory: sum of degree = 2 * (no. of edges) Case 1: the tree has only 1 node. Sum of degree = 2 * (no of edges) = 0 Case 2: The tree has more than one node. (n + 1) * (x - 1) + n + L = x + L -1 **Ans: L = x(n - 1) + 1**
15.	
16.	The minimum number of interchanges needed to convert the array 89, 19, 40, 17, 12, 10, 2, 5, 7, 11, 6, 9, 70 into a heap with the maximum element at the root node is (A) 0 (B) 1 (C) 2 (D) 3 Only element 70 violates the rule. Hence, it must be shifted to its proper position. Step1: swap(10, 70) Step2: swap(40, 70) Hence, only 2 interchanges are required.
17.	Heap allocation is required for languages (A) that support recursion (B) that support dynamic data structures (C) that use dynamic scope rules (D) none of the above [B] Dynamic data structures have the capacity to grow. Hence, they need some sort of memory allocation during runtime. The heap is required to provide this memory.
18.	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 [B] Total no. of nodes in the tree n = n0 + n1 + n2

Where ni is the number of nodes with i children.

Total no. of edges in the tree

e = n0*0 + n1*1 + n2*2

But e = n - 1

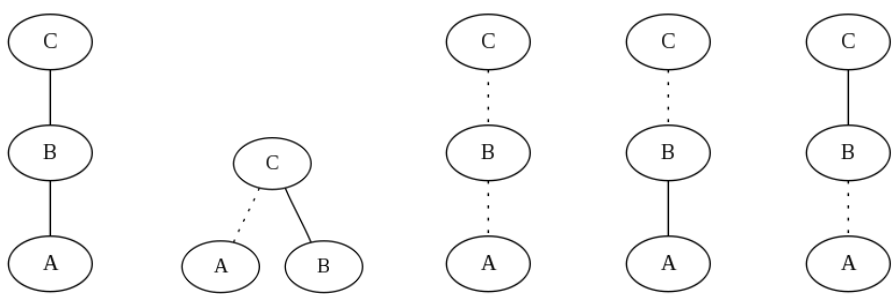
n1 + 2*n2 = n0 + n1 + n2 – 1

n2 = n0 – 1

n2 = n - 1

19. 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.

Ans: 5



(a) (b) (c) (d) (e)

20. In the balanced binary tree in Fig. 1.14 given below, how many nodes will become unbalanced when a node is inserted as a child of the node "g"?

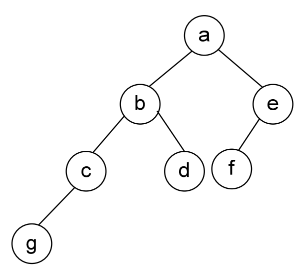


Fig. 1.14

(A) 1
(B) 3
(C) 7
(D) 8

[B]

After insertion

Bf(a) = 2

	Bf(b) = 2 **Bf(c) = 2** Bf(d) = 0 Bf(e) = 1 Bf(f) = 0 Bf(g) = 1 Which of the following sequences denotes the post-order traversal sequence of the tree of FIG 1.14? (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 [C] Post-order = Left sub-tree – right sub-tree – self
21.	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 is respectively is (A) (4, 7) (B) (7, 4) (C) (8, 3) (D) (3, 8) [B] Nodes to in the left subtree are less than 50, while nodes in the right sub-tree are greater than 50. No. of nodes less than 50 = 7 No. of nodes greater than 50 = 4
22.	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 Possible:[A,C, D] Not possible: [B] – at 50 the left branch is taken. Hence, all nodes in the list after 50 must be less than it. But 60 is not similarly in[D]18 is smaller than 27
23.	A binary search tree contains the values 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

Valid: d

Invalid: a, b, c

A preorder list has the following format –

<root> <left sub-tree (smaller elements)> <right sub-tree (larger elements)>

This format is applied recursively to the sub-trees.

24. 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 maximum distance of a leaf node from the root.

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

When

h = 0 ……… least no. of nodes = 2 ^ 0 = 1

h = 1 ……… least no. of nodes = 2 ^ 1 = 2

h = 2 ……… least no. of nodes = 2 ^ 2 = 4

Assume that the rule is true for h = k

Then the min no. of nodes = 2 ^ k nodes

If we increase the height by 1 by adding a node, we must also add nodes to fill the (max level -1) level.

This would mean doubling the nodes

Thus 2^ (k+1)

Hence, proved

(b) In a size-balanced binary tree of height h ≥ 1, how many nodes are at distance h-1 from the root? Write only the answer without any explanation

2^{h-1}

25. 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 inorder 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

[C] In order traversal of the binary search tree is always in ascending order

26. Which of the following statements is false?

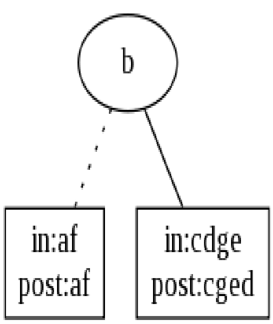
(a) A tree with n nodes has (n-1) edges
(b) A labeled rooted binary tree can be uniquely constructed given its post-order and pre-order traversal results.
(c) A complete binary tree with n internal nodes has (n+1) leaves
(d) The maximum number of nodes in a binary tree of height h is $(2^{h+1} - 1)$

[B and D] With inorder,preorder or with inorder, postorder it is possible for a unique combination

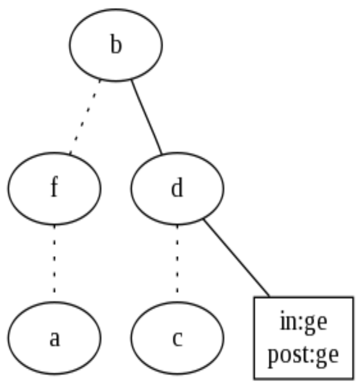
Maximum number of nodes in a binary tree of height h is $(2^{h} - 1)$

27. Draw the binary tree with the node labels a, b, c, d, e, f and g for which the inorder and post-order traversals result in the following sequences

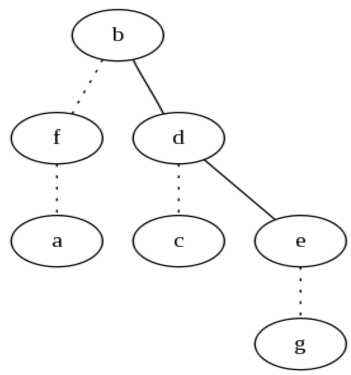
Inorder	a f b c d g e
Postorder	a f c g e d b



(a)

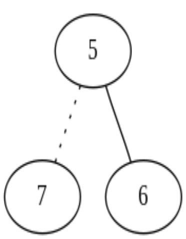


(b)

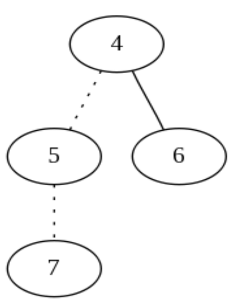


(c)

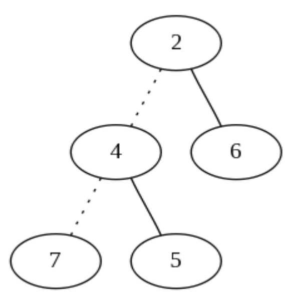
28. Draw the min-heap that results from insertion of the following elements in order into an initially empty min-heap: 7, 6, 5, 4, 2, 3, 1. show the result after the deletion of the root of this heap.



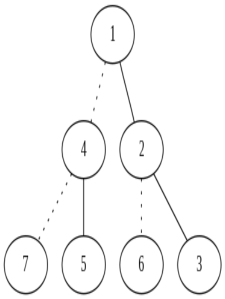
(a)



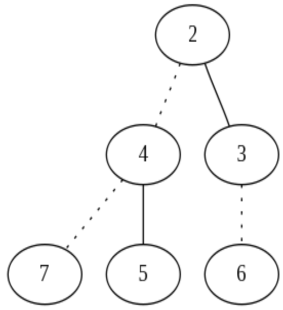
(b)



(c)



(d)



(e)

29. The number of leaf nodes in a rooted tree of n node, with each node having 0 or 3 children is:

A. n/2
B. (n – 1)/3
C. (n – 1)/2
D. (2n + 1)/3

[D]Degree of internal node (except root node) = 4

Degree of root = 3 or 0

Degree of leaf node = 1

No of edges = n – 1

From graph theory: sum of degree = 2 * (no. of edges)

Case 1: tree contains only root

Sum of degree = 2 * (no of edges) = 0

Case 2:

(n – L - 1) * 4 + 3 + L * 1 = 2 * (n - 1)

Ans: L = (2n + 1)/3

30. In a binary tree, a full node is defined to be a node with 2 children. Use the induction on the height of the binary tree to prove that the number of full nodes plus one is equal to the numbers of leaves.

1(root) element contains 2 children .no of full nodes=1,no. of leaves=2

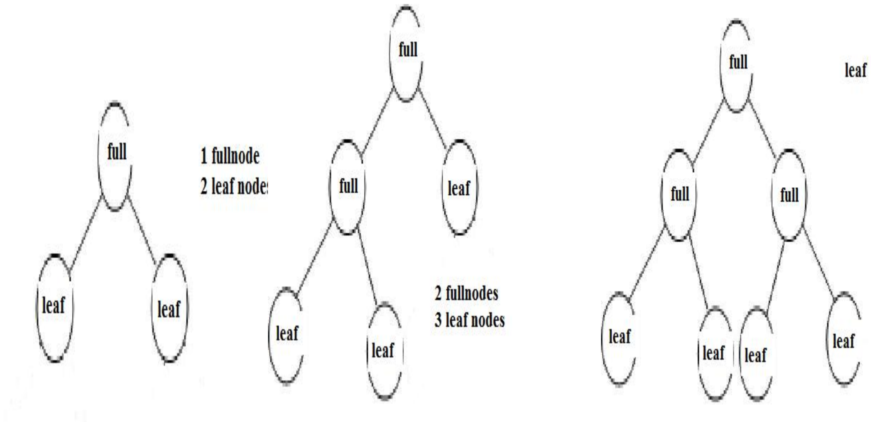
2 full nodes have 3 leaves

3 full nodes have 4 leaves

……………………………..

…………………………..

N full nodes have N+1 leaves



31. Consider the following nested representation of binary trees: (X Y Z) indicates Y 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))

[C] Every node contains two sub trees.

In [A] 4 contains 3 sub trees which violate the property.

[B] node 2 has 4 child 3 and 4 and 5 and 6

[D] node 4 right child is not specified NULL.

32. Let LASTPOST, LASTIN and LASTPRE denotes the last vertex visited in a post-order, inorder and pre-order traversal respectively, of a completely binary tree. Which of the following is always true?

(A) LASTIN = LASTPOST
(B) LASTIN = LASTPRE
(C) LASTPRE = LASTPOST
(D) None of the above

[B]

In-order = LNR

Pre-order = NLR

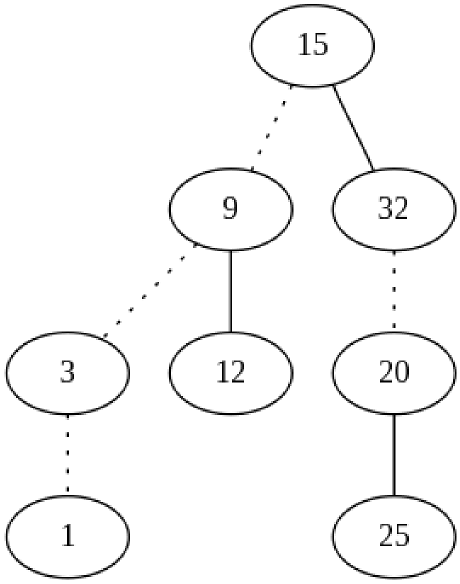
Post-order = LRN

The rightmost element will be the last element in both in and pre orders

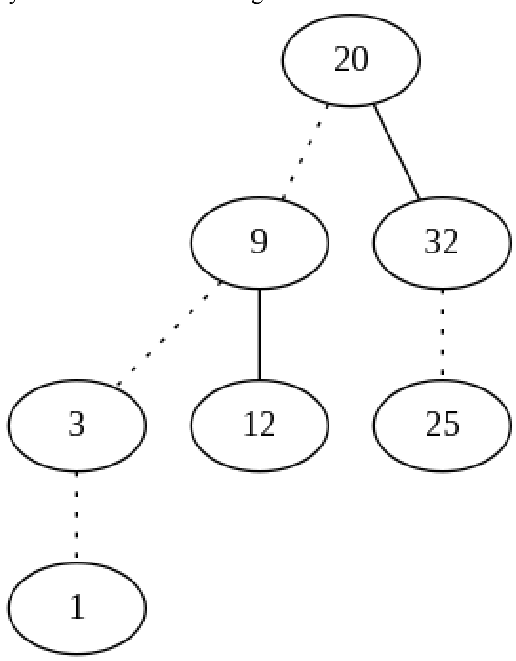
33. (a) Insert the following keys one by one into a binary search tree in order specified.

15, 32, 20, 9, 3, 25, 12, 1

Show the final binary search tree after 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 __________;
}
```

S1: y=depth(t->right)

S2: x+1

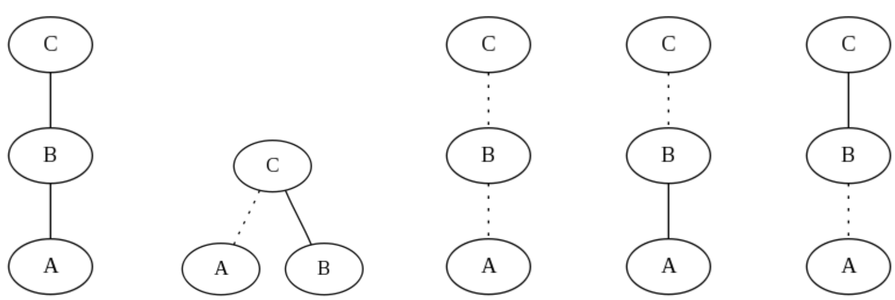
S3: y+1

34. A weight balanced tree is a binary tree in which for each node, the number of nodes in the left subtree is at least half and at most twice the number of nodes in the right subtree. 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. $\log_{4/3} n$
   C. $\log_3 n$
   D. $\log_{3/2} n$

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

Ans: 5



(a) (b) (c) (d) (e)

36. In heap with n elements with the smallest element at the root, the 7th smallest element can be found in time

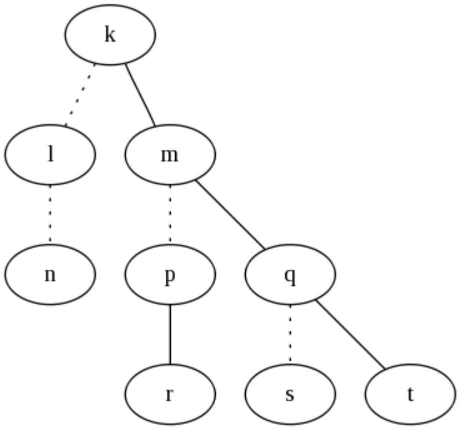
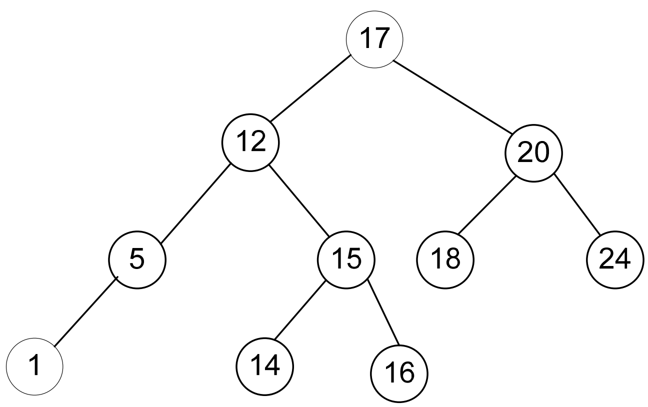
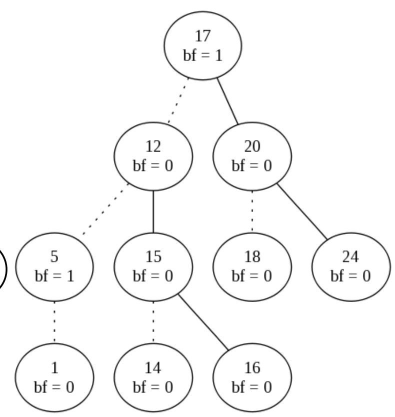
   (A) Θ(n log n)
   (B) Θ(n)
   (C) Θ(log n)
   (D) Θ(1)

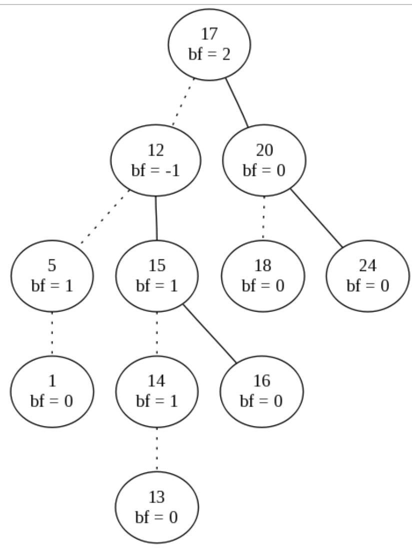
In order to find the 7th element one must remove elements one by one from the heap, until the 7th smallest element is found.

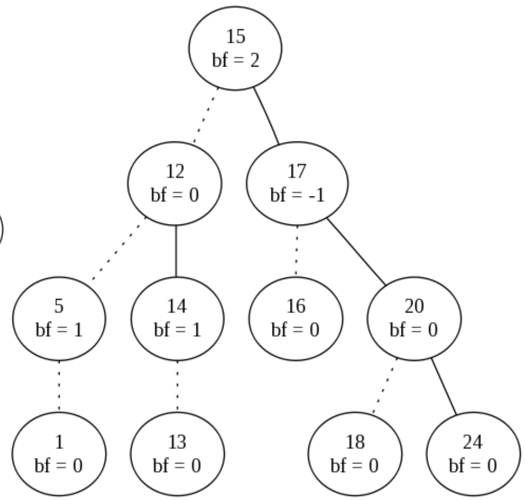
As a heap may contain duplicate values, we may need to remove more then 7 elements. At the worst case we will have to remove all n elements.

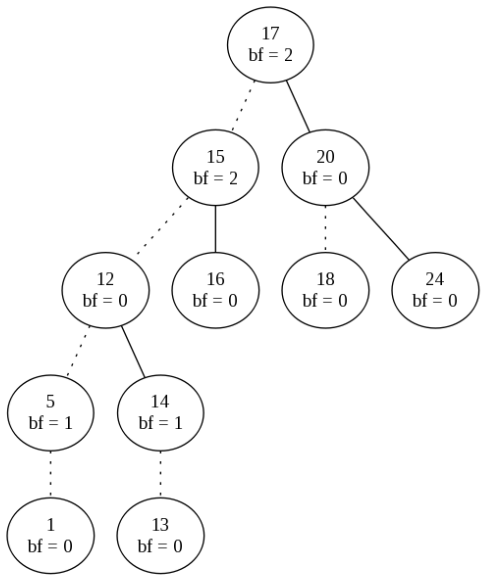
1 deletion takes O(log(n)). Hence ‘n’ deletes will take O(nlog(n))

37. 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

	
38.	(i) Define the height of a binary tree or subtree and also define a height balanced (AVL) tree. The height of a tree or a sub tree is defined as the length of the longest path from the root node to the leaf. A tree is said to be height balanced if all the nodes are having a balance factor -1,0 or 1. The balance factor is the height of the left sub tree minus height of the right sub tree (ii) Mark the balance factor of each mode on the tree given in fig 7.1 and state whether it is height-balanced   (iii) Into the same tree given in 7(ii) above, insert the integer 13 and show the new balance factors that would arise if the tree is no rebalanced. Finally, carry the required rebalancing of the tree and show the new tree with the balance factors on each mode.







39. The number of rooted binary trees with n nodes is,

(A) Equal to the number of ways of multiplying (n+1) matrices.
(B) Equal to the number of ways of arranging n out of 2n distinct elements.
(C) Equal to $\frac{1}{(n+1)}\binom{2n}{n}$
(D) Equal to n!

[C]

N=1 1 tree

N=2 2 trees

N=3 5 trees

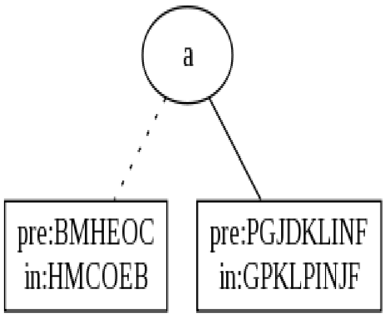
N=4 14 trees

1, 2 ,5, 14…………………..are catlan numbers

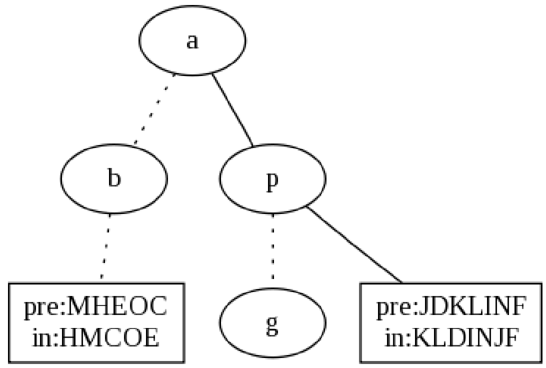
40. The maximum number of nodes in a binary tree of level k, $k \geq 1$ is

(E) $2^k + 1$
(F) $2^k - 1$
(G) 2^{k-1}
(H) $2^{k-1} - 1$

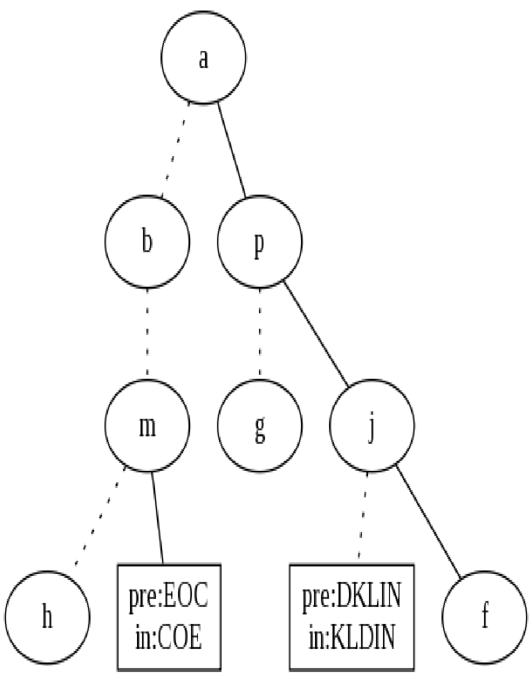
41. Construct a binary tree whose preorder and inorder sequences are A B M H E O C P G J D K L I N F and H M C O E B A G P K L D I N J F respectively, where A, B, C, D, E, ………. are the labels of the tree nodes. Is it unique?



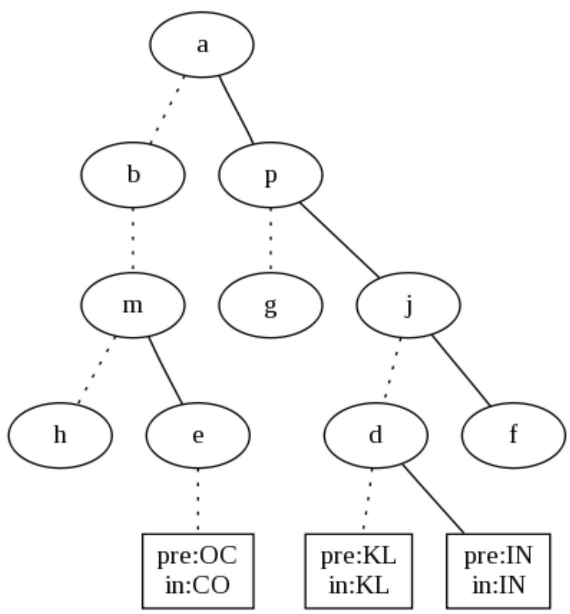
(a)



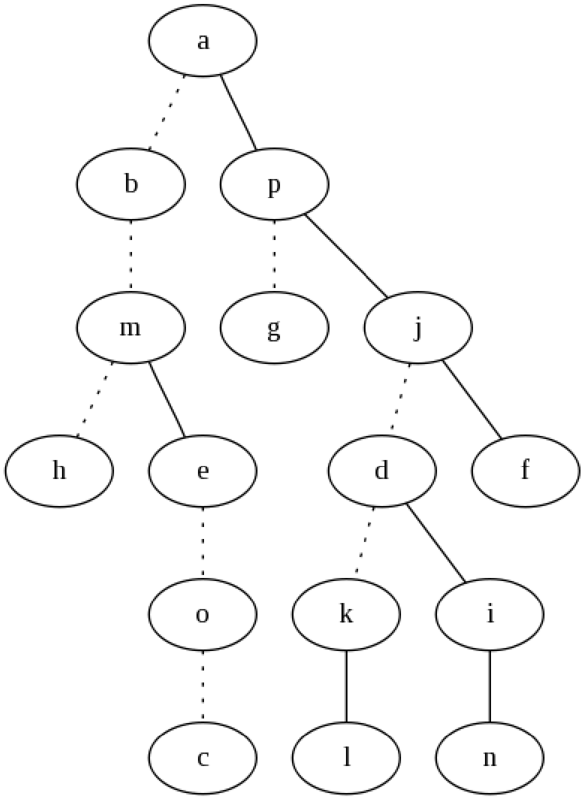
(b)



(c)



(d)



(e)

42. The minimum number of comparisons required to sort 5 elements is ________________.
4 using insertion sort when numbers are in sorted order

43. The weighted external path length of the binary tree in Fig. 2 is ______________.

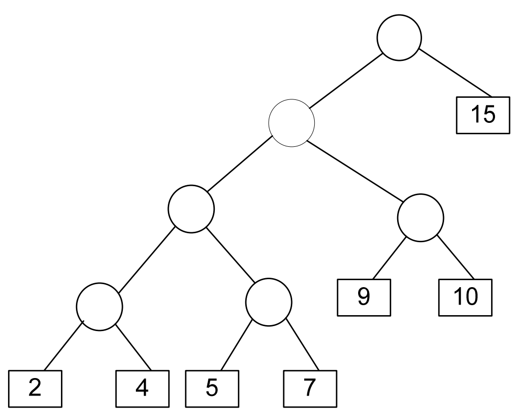


Fig 2

weighted external path = (2*4) + (4*4) + (5*4) + (7*4) + (9*3) + (10*3) + (15*1)

= 8 + 16 + 20 + 28 + 27 + 30 + 15

=144

44. If the binary tree in Fig. 3 is traversed in inorder, then the order in which the nodes will be visited is _________________________

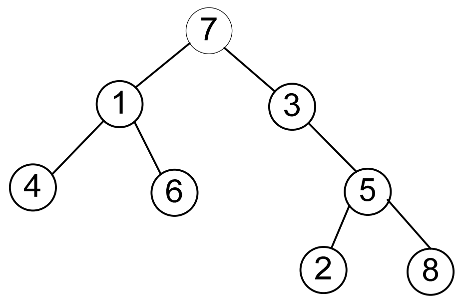


Fig 3

In-order traversal: 4 1 6 7 3 2 5 8

45. Consider the binary tree in Fig. 7:

(d) What structure is represented by the binary tree?
(e) Give the different steps for deleting the node with key 5 so that the structure is preserved.
(f) Outline a procedure in the pseudo-code to delete an arbitrary node from such a binary tree with n nodes that preserves the structure. What is the worst case complexity of your procedure?

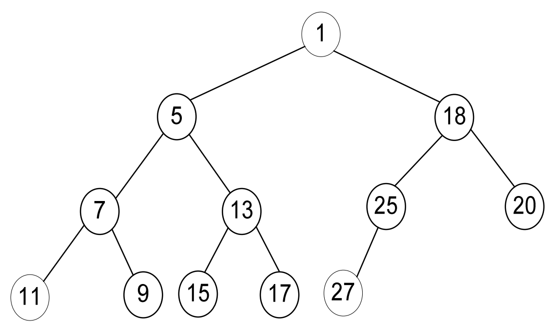


Fig. 7

(a)Min-Heap

(b)

iv. Swap(27,5)

v. Delete 5

vi. Minheapify

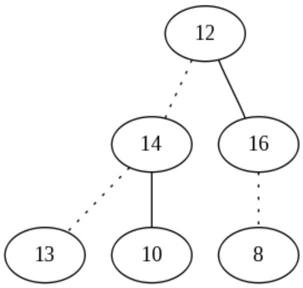
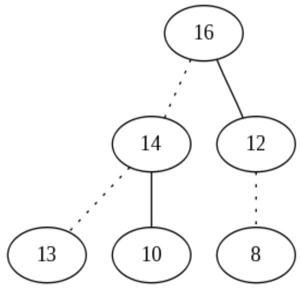
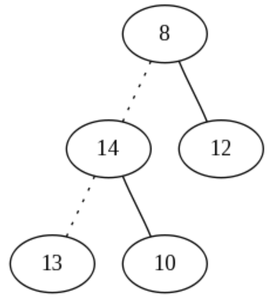
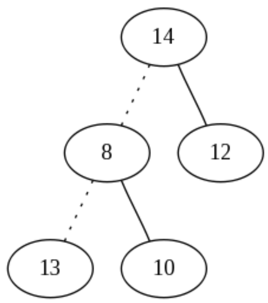
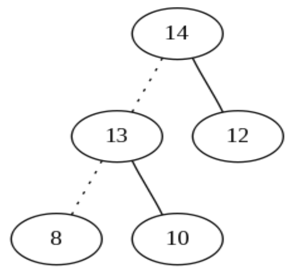
(c)

Alogithm delMinHeap()

Step 1: let I th node is to be deleted.

Step 2: copy last element to this location

	Step 3: remove last element of array Step 4: perform Min-Heapify on I th node Algorithm Min-Heapify() A-array, i-ith node Step 1: if 2*i<=size[A] and A[2i]<A[i] /*left child*/ Then smallest=2i Else smallest =i Step 2: if 2i+1 <=size[A] and A[2i+1] < A[smallest] Then smallest=2i+1 Step3: If smallest != i Then swap A[i] and A[smallest] Step 3: Min-Heapify(A,smallest)
46.	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 } [C] What is the content of the array after two delete operations on the correct answer to the previous question? (A) { 14, 13, 12, 10, 8 } (B) { 14, 12, 13, 8, 10} (C) { 14, 13, 8, 12, 10 } (D) { 14, 13, 12, 8, 10 } [D] (A) 12 has child 13 (B) 14 has child 16 (D) 12 has child 16 Hence, they are not max-heaps.

<table>
<tr><td></td><td>


(a) (b) (c)


(d) (e)</td></tr>
<tr><td>47.</td><td>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

[B]The maximum height of an AVL tree with n nodes is 1.44log n.

By substituting n=7 we get 4.03,since it is given that the height of a tree with a single node is 0.Therefore the answer is 3.</td></tr>
<tr><td>48.</td><td>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
[D]
II and III have the same last element. Thus they must be either pre-order or in-order.

Thus, I must be post-order.

But then K must be the root.</td></tr>
</table>

	K occurs as the first element tin II. Thus, II must be pre-order. Thus, III must be in-order.
49.	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 $O(n)$ [A]Searching in AVL tree has O(logn) complexity. And searching for binary search tree is also O(n), when it is skewed and is order of O(log(n)) when it is complete
50.	We have a binary heap on n elements and wish to insert n more elements (not necessarily one after another) into this heap. The total time required for this is (A) $\Theta(\log n)$ (B) $\Theta(n)$ (C) $\Theta(n \log n)$ (D) $\Theta(n^2)$ [C] For insertion of 1 node in heap will take O(log(n)) For insertion of n nodes will take O(nlog(n)).
51.	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. [C] To get unique tree we must have pre order and in order or post order and inorder. In order traversal of a BST is always sorted. I=Sort P(using merge sort or heap sort which takes O(nlogn)) Using I and P construct the tree.
52.	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 node containing the value 60? (A) 35 (B) 64 (C) 128 (D) 5040
53.	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) $\Theta(n \log_2 n)$ [B] n = no of elements in the tree.

lgn = height of the tree i.e. path length of the newly inserted node to the root.

Complexity of binary search = O(lgn) where n is the no. of elements searched.

Thus, O(ln(lg(n)))

54. 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

L = I(n - 1) + 1

Thus n = (L - 1) / I – 1

i.e n = (41 - 1) / 10 -1

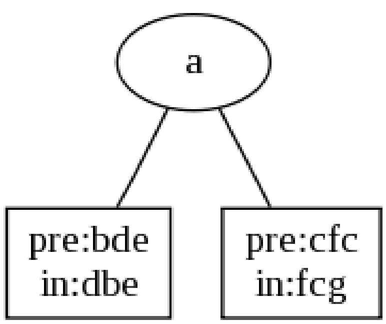
Ans: n = 3

55. The inorder and pre-order traversal of a binary tree are

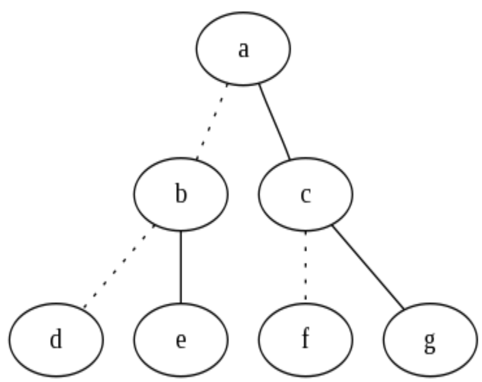
d b e a f c g and a b d e c f g, respectively.

The post-order 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
[A]



(a)



(b)

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

(A) 1 (B) 5 (C) 4 (D) 3

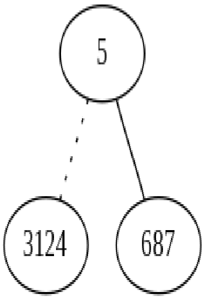
[B] Using formula (1/(n+1))* 2nCn , where n is the number of unlabeled node so 5

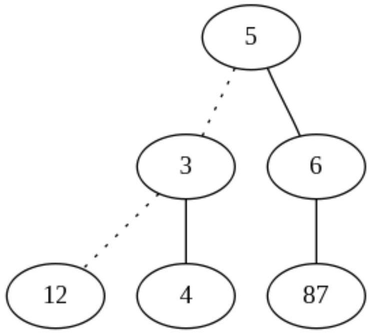
57. The height of a binary tree is the maximum number of edges in any root to leaf path. The maximum number of nodes in a binary tree of height h is:

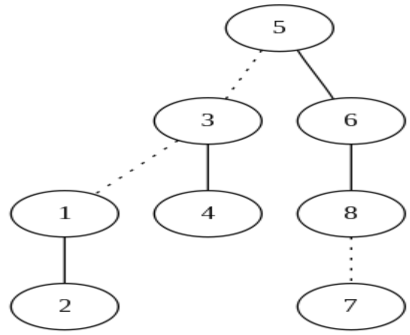
(A) 2^h (B) $2^{h-1} - 1$ (C) $2^{h+1} – 1$ (D) 2^{h+1}

Level 0: h = 0; nodes = 1

	Level 1: h = 1; nodes = 2 Level 2: h = 2; nodes = 4 Level n: h = n; nodes = 2^n Total no of nodes = 1 + 2 + 4 + … + 2^n = 2^(h+1) -1
58.	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} [C] On arriving at node 47, we take the right sub-tree. The right sub-tree can contain only those nodes which are larger than 47. But 43 < 47
59.	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} [C] In a max heap parent node is always greater than its children. Only [C] satisfies this property
60.	A scheme for sorting 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) $2^n - 1$ [B] In a complete binary tree to store n nodes an array of size n is needed.
61.	In a binary max heap containing n numbers, the smallest element can be found in time (A) Θ(n) (B) Θ(log n) (C) Θ(log log n) (D) Θ(1) [A]The smallest element can be found in last level of tree. So finding smallest we have to perform search in last level and that search can be performed in O(n).So total complexity will be O(logn+n)= O(n).
62.	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 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 [D]







(a) (b) (c)

63. 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) $2^{h-1} – 1$ (D) 2^h

N= h+[1/2*[(h-2)*(h-3)]]

64. The numbers 1, 2, …, 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) 134 (B) 133 (C) 124 (D) 123

1-----r-----n

Where r is the value of the root.

No. of elements between n and r is n – r = p

r = n - p

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

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

[C]Degree of internal nodes (except root) = k + 1

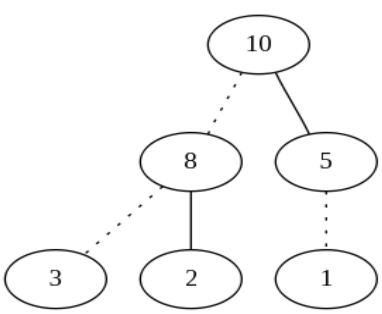
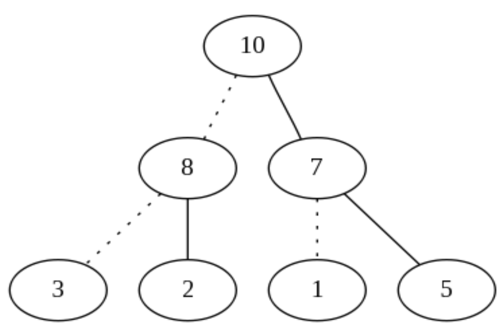
Degree of root = k

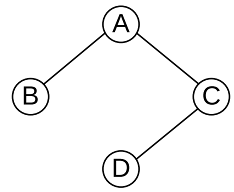
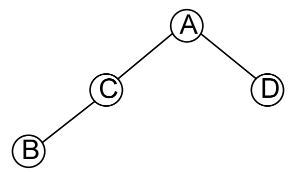
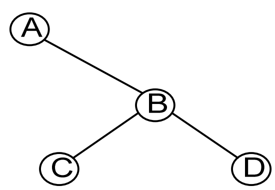
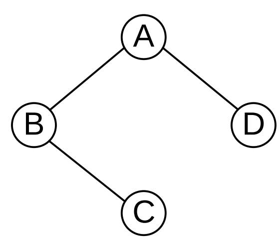
Degree of leaf nodes = 1

Also,

No of edges = no. of nodes – 1

But total no of nodes = n + L (L = leaf nodes)

	Thus, no. of edges = n + L -1 From graph theory: sum of degree = 2 * (no. of edges) Case 1: the tree has only 1 node. Sum of degree = 2 * (no of edges) = 0 Case 2: The tree has more than one node. (k + 1) * (n - 1) + k + L = n + L -1 **Ans: L = n(k - 1) + 1**
66.	How many distinct binary search tree can be created out of 4 distinct keys? (A) 5 (B) 14 (C) 24 (D) 42 [B] By formula $\frac{1}{(n+1)}\binom{2n}{n}$
67.	A Priority-Queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is given below: 10, 8, 5, 3, 2 Two new elements ‘1’ and ‘7’ are inserted in 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 [D]  (a)  (b)
68.	Which of the following binary trees has its inorder and preorder traversals as BCAD and ABCD, respectively?

	(A)  (B)  (C)  (D)  [D] With inorder and preorder,a tree can be uniquely identified.
69.	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 upto 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)

[B] Refer Cormen for proof

70. A program takes as input a balance 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 {no. of leaf nodes in left-subtree of x, no. of leaf-nodes in right-sub tree of x} then the worst time case complexity of the program is

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

[B] The recurrence relation for the recursive function is

T(N) = 2 * T(N/2) + n/2

Where N is the total no. of nodes in the tree.

T(N) = 2 * (2*T(N/2) + n/2) + n/2

= 4 * T(N/2) + 3(n/2)

Solve this till T(1) i.e. till we reach the root.

T(N) = c * T(N / 2^i) + (2*i - 1) * (n/2)

Where i = lg(N)

= lg((2n - 1) / 2)

O(c * T(N / 2^i) + (2*i - 1) * (n/2)) reduces to

O((2*i - 1) * (n/2))

O((2*(lg((2n - 1) / 2)) - 1) * (n/2)) …….sub the value of i.

O(n * ln(n))

71. 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);

                }

        }
```

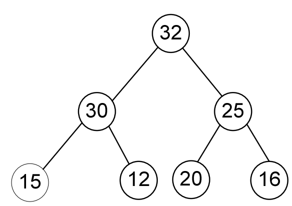
The value returned by the function DoSomething when a pointer to the root of a non-empty tree is passed as argument is

(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

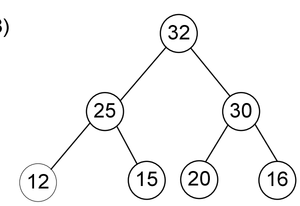
[D]Assuming that the function returns 'value'.The leaf nodes return 0. Each internal node adds 1 to the maximum returned value.

72. The elements 32, 15, 20, 30, 12, 25, 16 are inserted one by one in the given order into a maxheap. The resultant max heap is

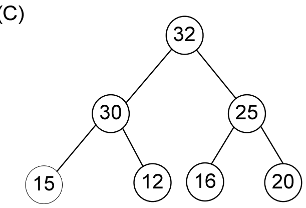
(A)



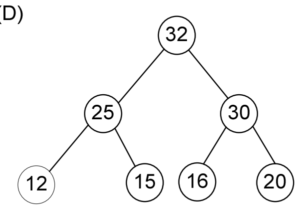
(B)



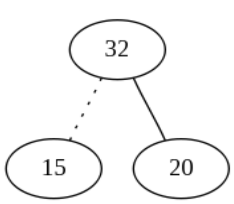
(C)

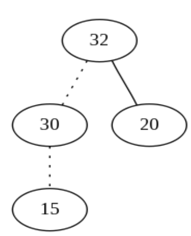


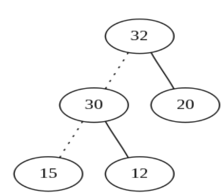
(D)



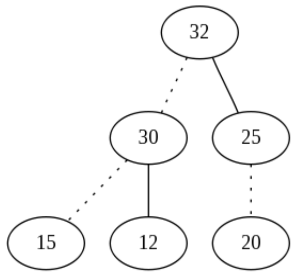
[A]

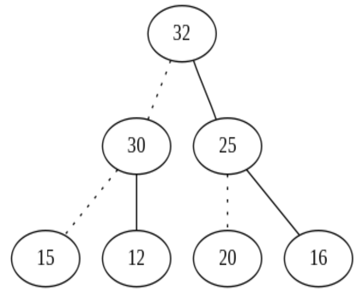






(a) (b) (c)





(d) (e)

73. 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) Pre-order and post-order
(ii) Inorder and post-order
(iii) Pre-order and inorder
(iv) Level order and post-order

(A) (i)only
(B) (i), (iii)
(C) (iii)only
(D) (iv)only

(ii) and (iii)

here [C] is also correct

74. Level order traversal of a rooted tree can be done by starting the root and performing

(A) Pre-order traversal
(B) Inorder traversal
(C) Depth first search
(D) Breadth first search

[D] A breath first search is implemented by enqueueing the root and the repeating the following-

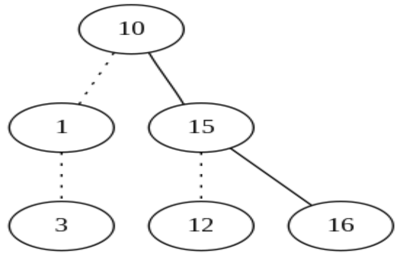
Step: dequeue a node and push its children in the queue from the left to right.

This will give a level-order traversal of the tree.

75. The following numbers are inserted into an empty binary search tree in the given order: 10, 1, 3, 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

[A]


10
1
15
3
12
16