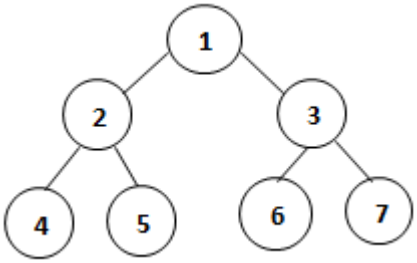
PROGRAMMING AND DATA STRUCTURES

TREES (Set-1)

SOLUTION

1. Consider the following tree:



If the post-order traversal gives ab-cd*+ then the label of the nodes 1, 2, 3,…..will be:

(a) +, -, *, a, b, c, d
(b) a, -, b, +, c, *, d
(c) a, b, c, d, -, *, +
(d) -, a, b, +, *, c, d

Solution: Option (a)

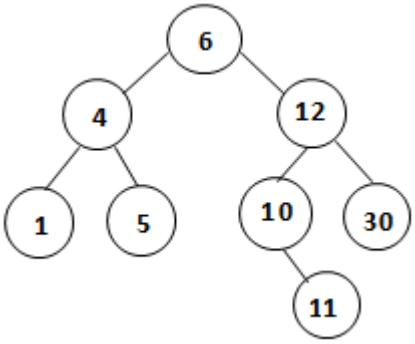
Explanation:

The post order traversal of a tree is recursively defined as visiting the left child then the right child and parent(LRP) L- left child, R- Right Child ,P-Root. In this manner if we perform post order traversal of above tree we will get

4 5 2 6 7 3 1

This matches option (a)

2. Consider the following tree:



If this tree is used for sorting, then a new number 8 should be placed as:

(a) left child of the node labeled 30
(b) right child of the node labeled 5
(c) right child of the node labeled 30
(d) left child of the node labeled 10

Solution: Option (d)

Explanation:

A tree is used for sorting means that the inorder traversal of the tree should give numbers in sorted order. So 8 should be inorder successor of node labelled 10 , therefore it should be left child of 10.

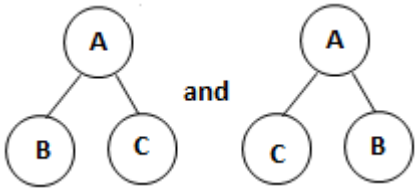
3. The number of possible ordered trees with 3 nodes A, B, C is:

(a) 16 (b) 12
(c) 6 (d) 10

Solution: Option (b)

Explanation:

The tree maybe of depth 2 or 1. If depth is 2, we have 6 possible trees. This is because one of the 3 nodes A, B, C may be the root and the next level may be one of the remaining 2 nodes. If the depth is 1, the root maybe one of the 3 nodes A, B, C. Corresponding to a root, say A, 2 trees are possible as this.



This gives us 6 more possibilities.

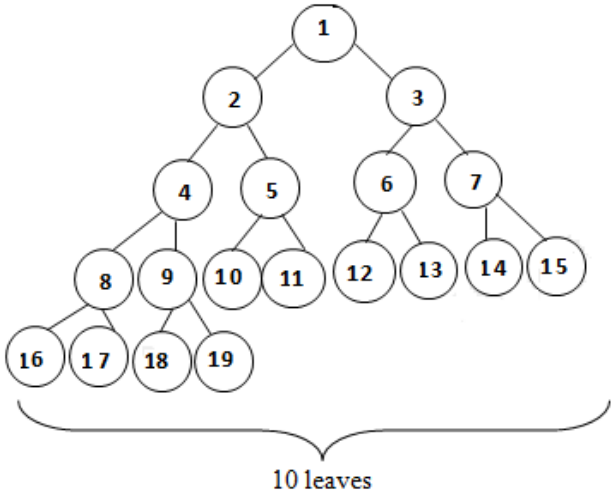
4. A binary tree in which every non-leaf node has non-empty left and right subtrees is called a strictly binary tree. Such a tree with 10 leaves.

(a) cannot have more than 19 nodes (b) has exactly 19 nodes
(c) has exactly 17 nodes (d) cannot have more than 17 nodes

Solution: Option (b)

Explanation:

The configuration of 10 leaves can only be of the following way:



Any tree with n-leaves, for strict binary tree has (2n-1) nodes.

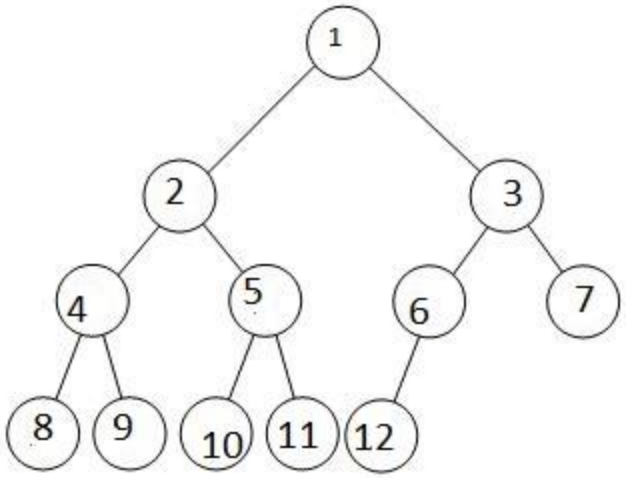
5. The depth of a complete binary tree with 'n' nodes is (log is to the base 2):

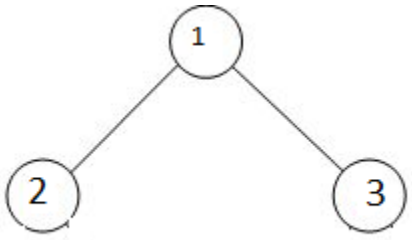
(a) log (n+1) -1
(b) log (n)
(c) log (n-1) +1
(d) log (n) +1

Solution: Option (a)

Explanation:

A complete binary tree is a binary tree where all levels except possibly the last is completely filled, and and all nodes are as far left as possible .





The diagrams above show complete binary trees with 12 nodes and 3 nodes respectively. Using this we can decude the answer.

6. Preorder is same as:

(a) depth-first order
(b) breadth-first search
(c) topological order
(d) linear order

Solution: Option (a)

7. Which of the following traversal techniques lists the nodes of a binary search tree in ascending order?

(a) Post-order
(b) In-order
(c) Pre-order
(d) None of these

Solution: Option (b)

8. The no. of possible Binary trees with 3 nodes are:

(a) 12 (b) 13
(c) 5 (d) 15

Solution: Option (c)

Explanation:

The number of different unlabelled trees we get with 3 nodes is equal to the C3 (where Cn is the nth Catalan number) .

$$Cn = \frac{2n_n}{(n+1)} = \frac{6_3}{(3+1)} = \frac{6_3}{4} = 5$$

9. The number of possible binary trees with 4 nodes are:

(a) 12 (b) 13
(c) 14 (d) 15

Solution: Option (c)

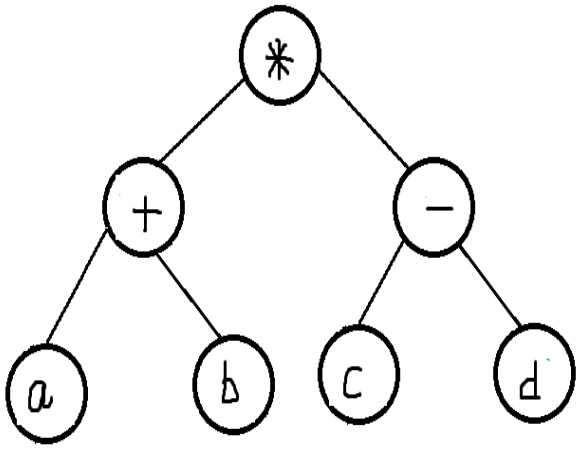
10. The postfix equivalent of the tree whose prefix order is → * + ab – cd is:

(a) ab + cd - * (b) ab cd + - *
(c) ab + cd * - (d) ab + - cd *

Solution: Option (a)

Explanation:

Using the given prefix order we get the infix order as (a+b)*(c-d) . Now we can find the tree using the prefix and infix expression. The tree is shown below.



From the tree we can find the postfix expression.

11. A binary tree has n leaf nodes. The no. of nodes of degree 2 in this tree is:

(a) $\log_2 n$ (b) n-1
(c) n (d) 2^n

Solution: Option (b)

Explanation:

Check for small values of n.

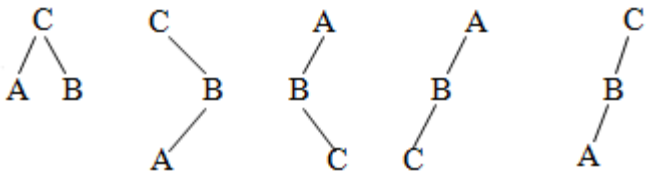
12. The no. of binary trees with 3 nodes which when traversed by post-order gives the sequence A, B, C is:

(a) 3 (b) 9
(c) 7 (d) 5

Solution: Option (d)

Explanation:

The possible configurations are:



13. A 3-ary tree is a tree in which every internal node has exactly 3 children. The no. of leaf nodes in such a tree with 6 internal nodes will be:

(a) 10 (b) 23
(c) 17 (d) 13

Solution: Option (d)

Explanation:

Let us first see what happens when number of internal nodes is 1, we have 3 leaf nodes. Let N(I) denote the number of leaf nodes such that there are I internal nodes. Now to increase the number of internal node by 1 , we select a leaf node and make it have 3 child. Therefore now we have decreased the leaf nodes by 1 and increased by 3. Therefore net increase in the number of leaf nodes is 2. Hence we get a recurrence as,

$$N(I) = N(I-1) + 2 \quad I>1$$
$$= 3 \quad I=1$$

Solving this we get N(I) = 2*I +1. We can use this to get the answer.

14. Which of the following need not be binary tree?

(a) Search tree (b) Heap
(c) AVL- Tree (d) B-Tree

Solution: Option (d)

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

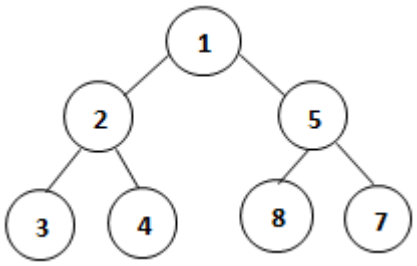
(a) 2^n-1 (b) $2^{n-1}-1$
(c) $2^{n+1}-1$ (d) 2^{n+1}

Solution: Option (c)

Explanation:

Check for small values for n,

For n=2



We get Option (c) $2^{2+1} - 1 = 2^3 - 1 = 7$

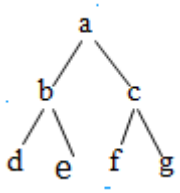
16. The inorder and preorder traversal of a binary tree are:
d b e a f c g and a b d e c f g respectively. The post order traversal of the binary tree is:

(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

Solution: Option (a)

Explanation:

The tree is:



17. The binary search tree contains the values—1, 2, 3, 4, 5, 6, 7 and 8. The tree is traversed in preorder 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

Solution: Option (d)

Explanation:

The tree for option (d) is: