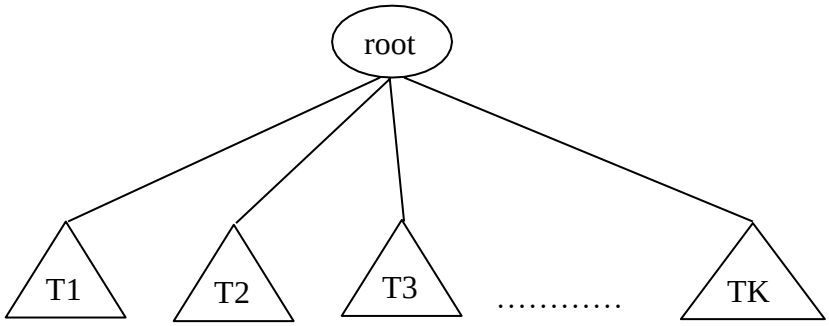
UNIT III: TREES

Part-A

1. Define Tree .Give an example.

A tree is a collection of nodes .The collection can be empty .Otherwise a tree consists of a distinguished node r called the root and 0 or more non empty sub-trees $T_1, T_2, T_3 \ldots\ldots\ldots\ldots\ldots T_k$ each of whose roots are connected by a directed edge from r.

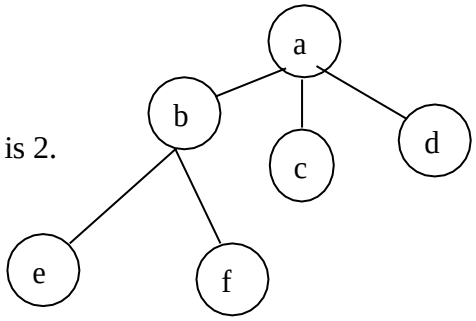


Eg: directory structure hierarchy

2. Define depth of a node in a tree. Give example.

For any node n_i the depth of n_i is the length of the unique path from the root to n_i

eg:

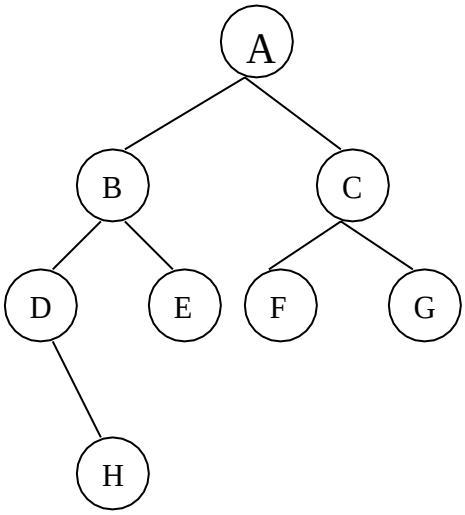


The depth of e is 2.

3. Write the routine for node declaration in trees.

```
typedef struct TreeNode *PtrToNode;
struct TreeNode
{
        ElementType Element;
        PtrToNode FirstChild;
        PtrToNode NextSibling;
};
```

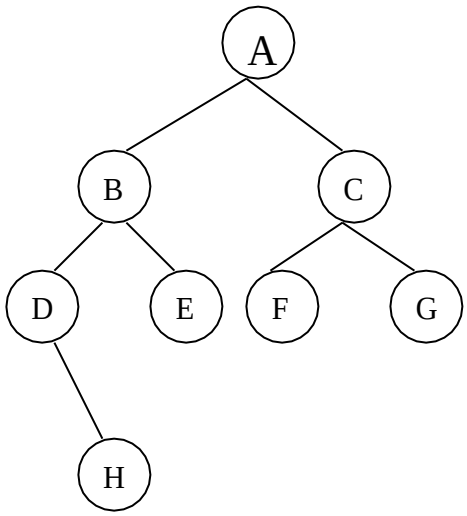
4. Define length of the path in a tree with an example.



Length of a path is the number of edges on the path. The length of the path from A-H is 3.

5. Define a path in a tree. Give example.

A path from a node n_1 to n_k is defined as the sequence of nodes n_1, n_2…..n_k such that n_i is the parent of n_{i+1} for $1 \leq i < k$.



Eg:

The path from A-H is A-B-D-H

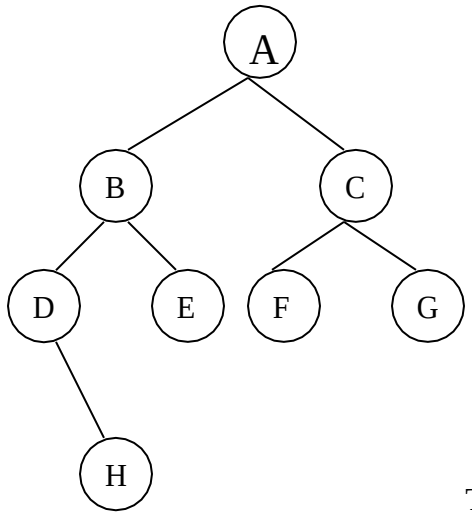
6. List the applications of trees.

- Binary search trees
- Expression trees
- Threaded binary trees

7. Define height of the node in a tree. Give example.

The height of node n_i the length of the longest path from n_i to a leaf

Eg:



The height of node B is 2.

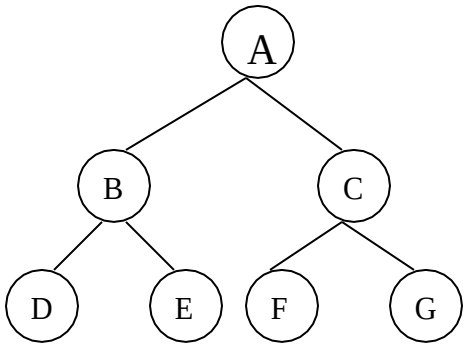
8. List the tree traversal applications.

1. Listing a directory in an hierarchal file system (preorder)
2. Calculating the size of a directory (post order)

9. Define binary tree ADT with an example.

A binary tree is a tree in which no node can have more than two children.

For Eg:,



10. Define binary search tree?

Binary Search tree is a binary tree in which each internal node *x* stores an element such that the element stored in the left sub tree of *x* are less than or equal to *x* and elements stored in the right sub tree of *x* are greater than or equal to *x*. This is called binary-search-tree

11. List the Types of binary search trees

i) Performance comparisons

ii) Optimal binary search trees

12. List the uses of binary tree.

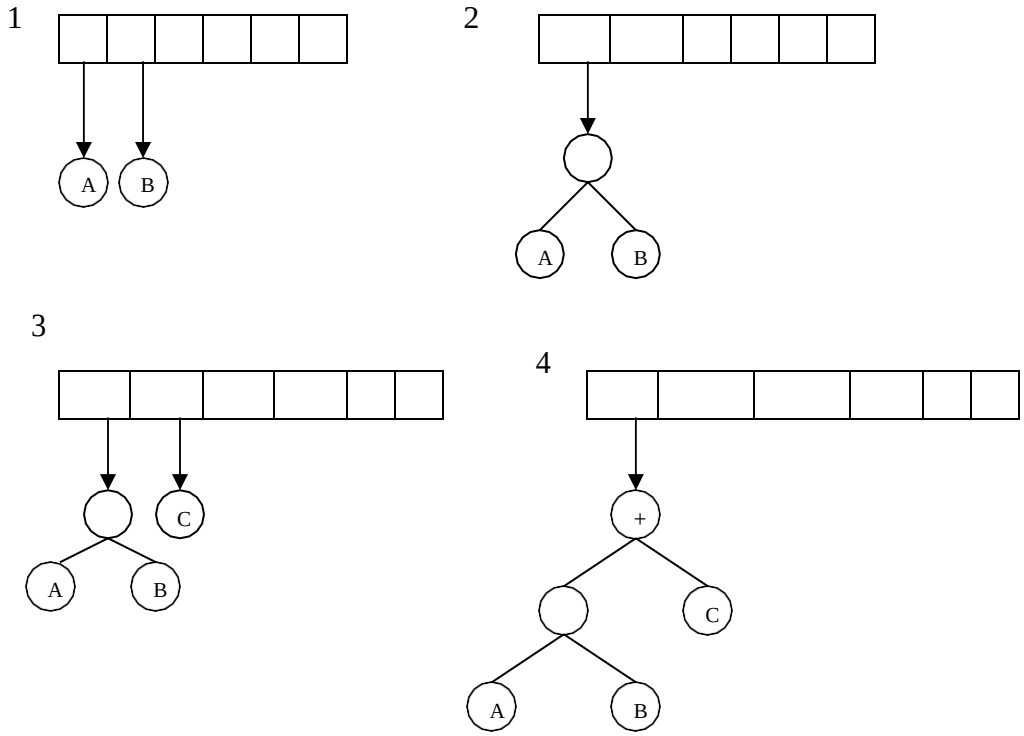
1. Searching. 2. Compiler design.

13. List the Operations of binary search tree?

- Make Empty
- Find
- Insert
- Delete
- Search
- Display

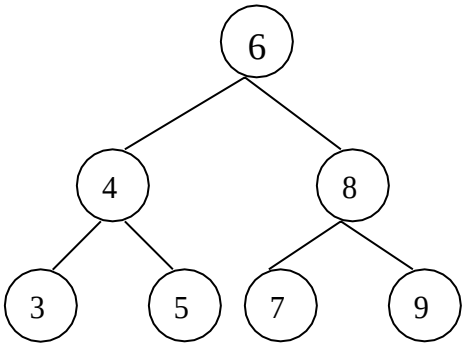
14. Draw the expression tree for the given postfix expression using stack.

AB*C+



15. Define binary search tree ADT with an example.

A binary search tree is a tree in which for every node X, the values of all the keys in its left sub tree are smaller than the key value in X and the values of all the keys in its right sub tree are larger than the key value in X.



16. Define internal path length.

It is the sum of the depths of all nodes in a tree.

17. How deletion is performed in a binary search tree.

Once the node to be deleted is found there are three possibilities

1. If the node is a leaf, it can be deleted immediately.

2. If the node has one child the node can be deleted after its parent adjusts a pointer to bypass the node.

3. If the node has two children the general strategy is to replace the data of this node with the smallest data of the right sub tree and recursively delete the node which is empty.

18. What is the average depth of all nodes in an equally likely tree?

The average depth of all nodes in an equally likely tree is O (log N).

19. List the disadvantages of Binary search tree.

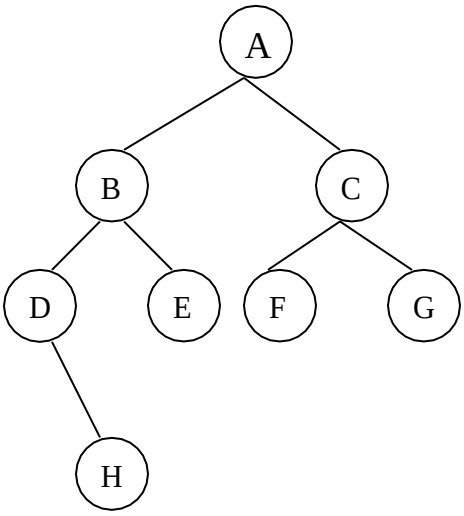
1. Deletions in a binary search tree lead to trees which are not equally likely.
2. Absence of balanced search tree.
3. The average depth is not O (log N) in trees which are not equally likely.

20. Define tree traversal, List out the types of Tree traversal?

Traveling through all the nodes of the tree in such a way that each node is visited exactly once. There are three types of tree traversal

1. Preorder traversal
2. In order traversal
3. Post order traversal

21. Perform preorder traversal for the given tree.



Preorder: **ABDHECFG**

22. Define the following.

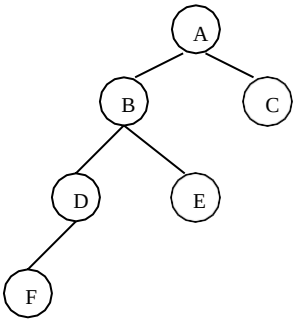
i) Leaf - Nodes at the bottommost level of the tree are called **leaf nodes**

ii) Sibling - The nodes with common parent are called Sibling

23. Define expression trees?

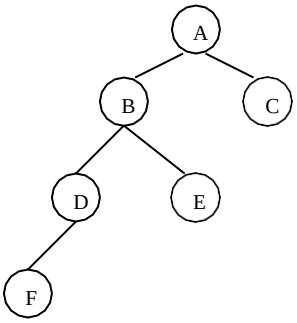
The leaves of an expression tree are operands such as constants or variable names and the other nodes contain operators.

24. Perform in order traversal for the given tree.



Inorder:FDBEAC

25. Perform postorder traversal for the given tree.



Postorder:FDEBCA

26. Define expression trees?

The leaves of an expression tree are operands such as constants or variable names and the other nodes contain operators.

27. Define strictly binary tree?

If every nonleaf node in a binary tree has nonempty left and right subtrees, the tree is termed as a strictly binary tree.

28. Define complete binary tree?

A complete binary tree of depth d is the strictly binary tree all of whose are at level d.

29. What is an almost complete binary tree?

A binary tree of depth d is an almost complete binary tree if :Each leaf in the tree is either at level d or at level d-1.For any node nd in the tree with a right descendant at level d,all the left descendants of nd that are leaves are at level d.

30. What is mean by Full Binary Tree?

A binary tree is said to be full, is all its leaves are at the same level and every interior node has two children.

31. Define right – in-threaded tree?

Right –in –threaded binary tree is defined as one in which threads replace NULL pointers in nodes with empty right sub trees.

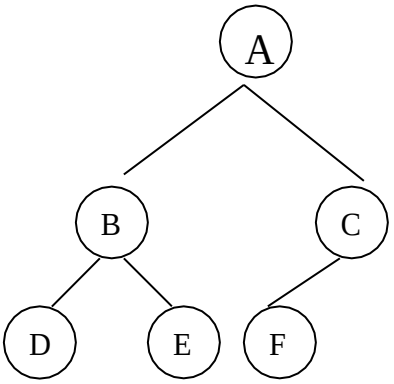
32. Define left – in –threaded tree?

A left-in-threaded binary tree may be defined as one in which each NULL pointers is altered to contain a thread to that node's in order predecessor.

33. Define AVL Tree. Give Example.

An AVL Tree is a binary search tree with a balance condition, which is easy to maintain and ensure that the depth of the tree is O(logN). Balance condition require that the left and the right sub trees have the same height.

Example:



34. Define Balance factor.

The balance factor of a node in binary tree is defined to be $|h_L-h_R|<=1$, where h_L and h_R are heights of left and right sub trees of T. For any node in AVL tree the balance factor should be 1,0 or -1.

35. What are the various transformation performed in AVL tree?

1. Single rotation:

-Single L rotation

-Single R rotation

2. Double rotation

-LR rotation

-RL rotation

36. When AVL tree property is violated and how to solve it?

After insertion of any node in an AVL tree if the balance factor of any node becomes other than -1, 0, or 1 then it is said that AVL property is violated. So the node on the path from the inserted node to the root needs to be readjusted. Check the balance factor for each node in the path from inserted node to the root node and adjust the affected sub tree such that the entire sub tree should satisfy the AVL property.

37. Mention the four cases to rebalance the AVL tree.

- An insertion of new node into Left sub tree of Left child (LL).
- An insertion of new node into Right sub tree of Left child (LR).

- An insertion of new node into Left sub tree of Right child (RL).
- An insertion of new node into Right sub tree of Right child (RR).

38. List the Operations on Binary heap.

- Adding to the heap
- Deleting the root from the heap

39. What is priority queue?

A Queue in which we are able to insert items or remove items from any position based on some priority is after referred to as a priority Queue.A priority queue is a data structure that allows at least the following two operations: insert which does the obvious thing; and Deletemin, which finds, returns, and removes the minimum element in the priority queue. The Insert operation is the equivalent of Enqueue.

40. Define min heap?

A heap in which the parent has a smaller key than the child's is called a min heap.

41. Define max heap?

A heap in which the parent has a larger key than the child's is called a max heap.

42. **Define binary heaps.**

A binary heap is a heap data structure created using a binary tree. It can be seen as a binary tree with two additional constraints:

- The *shape property*: the tree is an *almost complete binary tree*; that is, all levels of the tree, except possibly the last one (deepest) are fully filled, and, if the last level of the tree is not complete, the nodes of that level are filled from left to right.
- The *heap property*: each node is greater than or equal to each of its children according to some comparison predicate which is fixed for the entire data structure.

43. List the Applications of Binary heap

- Heap sort
- Selection Algorithm and Graph Algorithm

44. Application of priority queues?

1. Schedule the process in operating system.
2. It is used for external sorting.
3. To implement of greedy algorithm.

45. What are the main properties of a binary heap?

1. Structure property
2. Heap order property

46. List the Operations on B-Trees.

1. Search
2. Create
3. Insert

47. List the B-Trees Applications.

1. Databases
2. Concurrent Access to B-Trees

48. Define B-Tree.

A search tree that is not a binary tree is called B-Tree. That satisfies the following structural properties

1. Root is either a leaf or has between 2 and M children
2. All non leaf nodes except the root have between [M/2] and M children.
3. All leafs are at the same depth.

49.