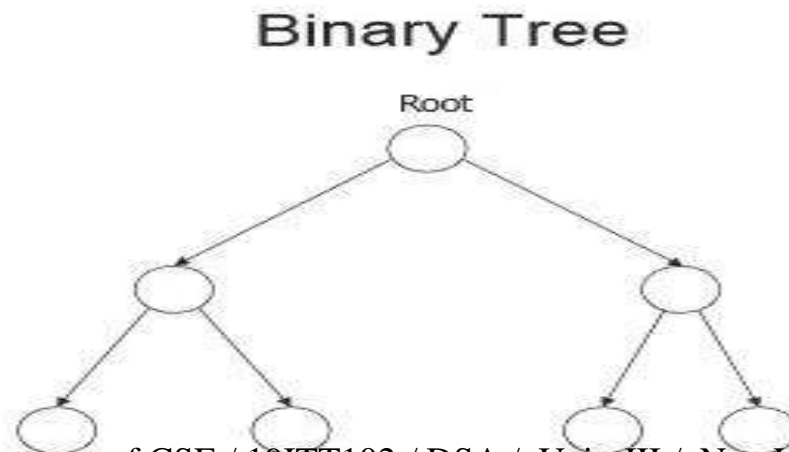




Binary tree

- A Binary tree is a data structure in that each node has at most **two nodes** left and right
- In binary tree, root has **in-degree 0** and maximum **out-degree 2**.
- In binary tree, each node have in-degree **one** and maximum out-degree **2**.
- Height of a binary tree is : $\text{Height}(T) = \{ \max (\text{Height}(\text{Left Child}) , \text{Height}(\text{Right Child}) + 1 \}$





Representation of Binary Tree

1. Array Representation
2. Linked List Representation.



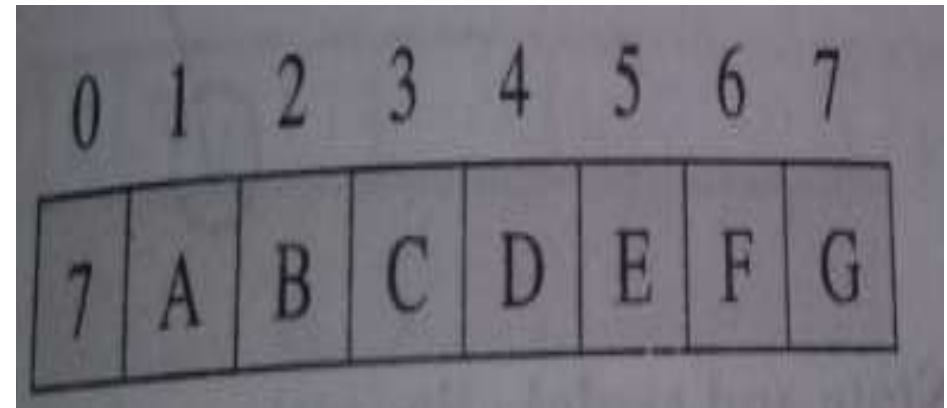
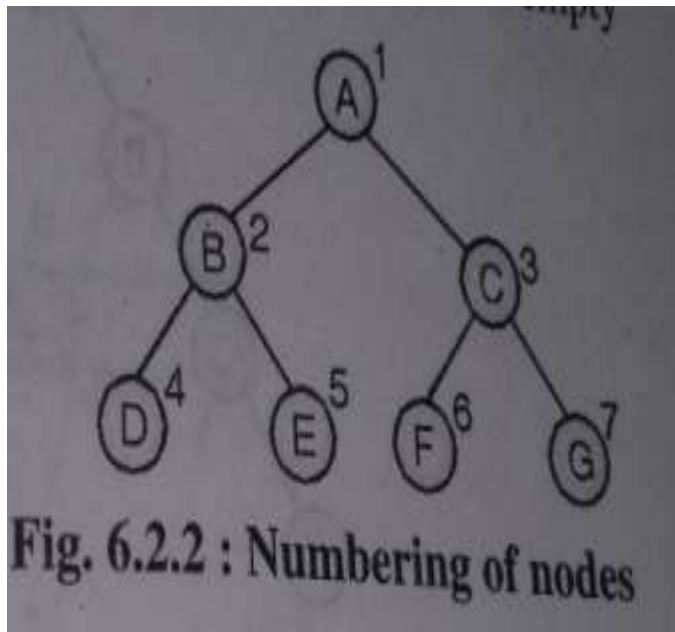
Representation of Binary Tree

```
Struct node  
{  
  
int data;  
struct node * left,right;  
};
```



Array Representation

1. To represent a tree in one dimensional array nodes are marked sequentially from left to right start with root node.
2. First array location can be used to store no of nodes in a tree.



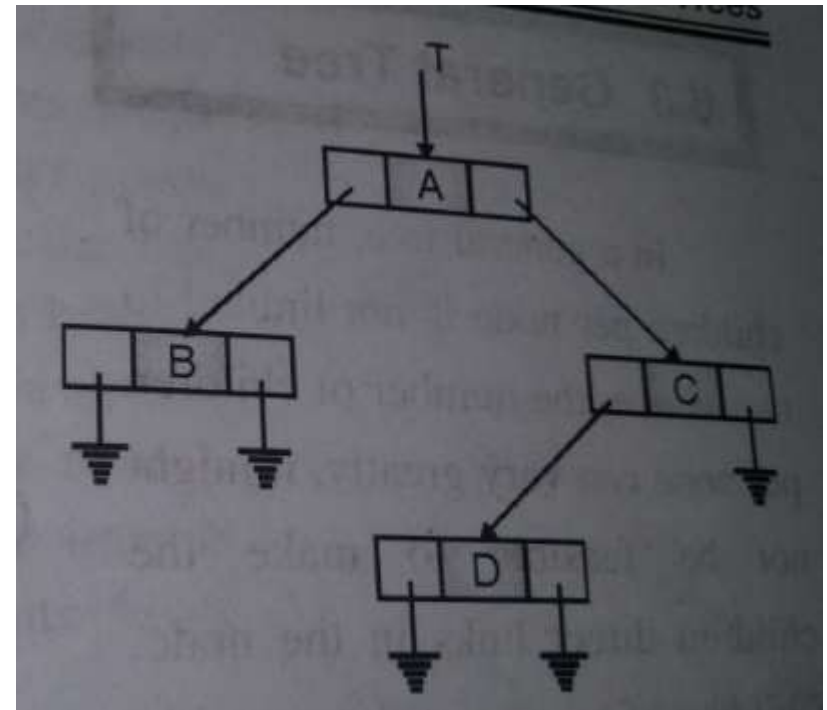
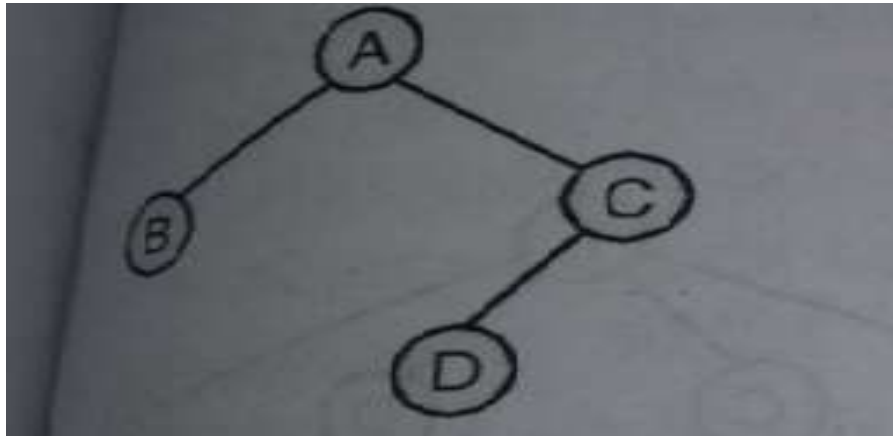


Linked Representation

1. This type of representation is more efficient as compared to array.
2. Left and right are pointer type fields left holds address of left child and right holds address of right child.

3. *Struct node*

```
{  
  
int data;  
struct node * left,*right;  
};
```





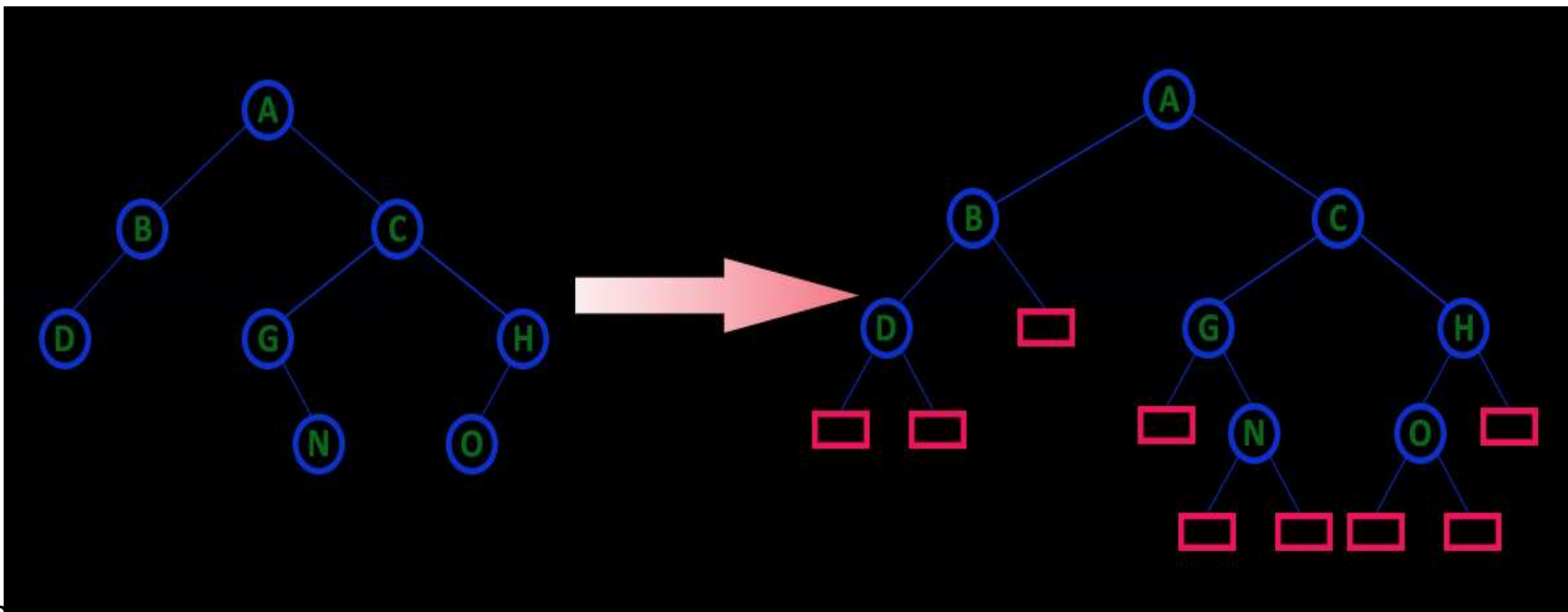
Binary Tree Types

1. Extended Binary Tree
2. Complete Binary Tree
3. Full Binary Tree
4. Skewed Binary Tree
5. Strictly Binary Tree
6. Expression Binary tree



Extended Binary Tree

An **extended binary tree** is a transformation of any **binary tree** into a complete **binary tree**. This transformation consists of replacing every null subtree of the original **tree** with “special nodes” or “failure nodes”. The nodes from the original **tree** are then internal nodes, while the “special nodes” are external nodes.

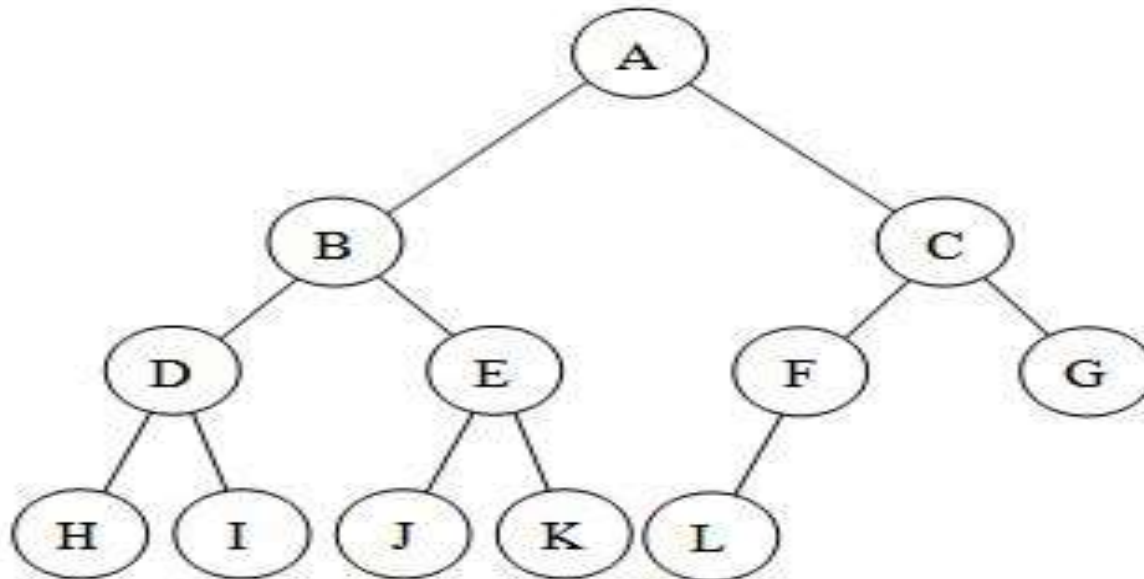




Complete Binary Tree

A **complete binary tree** is a **tree** in which

1. All leaf nodes are at n or $n-1$ level
2. Levels are filled from left to right



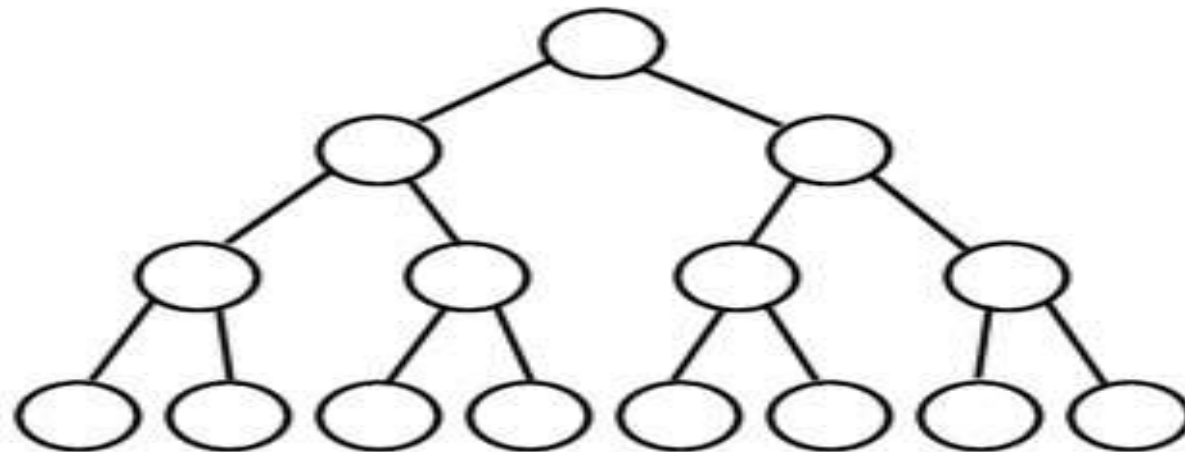


Full Binary Tree

A **full binary tree** (sometimes proper **binary tree** or **2-tree**) is a **tree** in which every node other than the leaves has two children or no children. Every level is completely filled up.

$$\text{No of nodes} = 2^{h+1} - 1$$

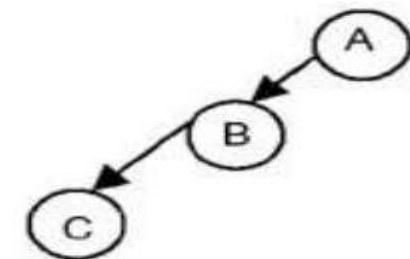
Full Binary Tree



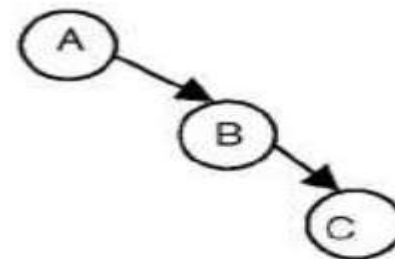


Skewed Binary Tree

A binary tree is said to be *Skewed Binary Tree* if every node in the tree contains either only left or only right sub tree. If the node contains only left sub tree then it is called *left-skewed binary tree* and if the tree contains only right sub tree then it is called *right-skewed binary tree*.



Left Skewed

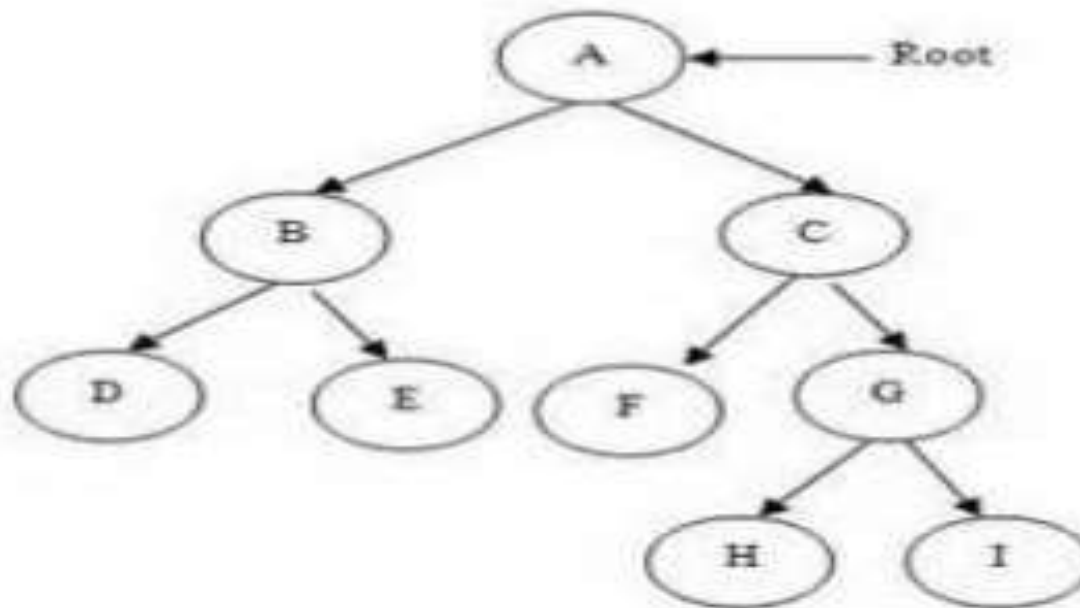


Right Skewed



Strictly Binary Tree

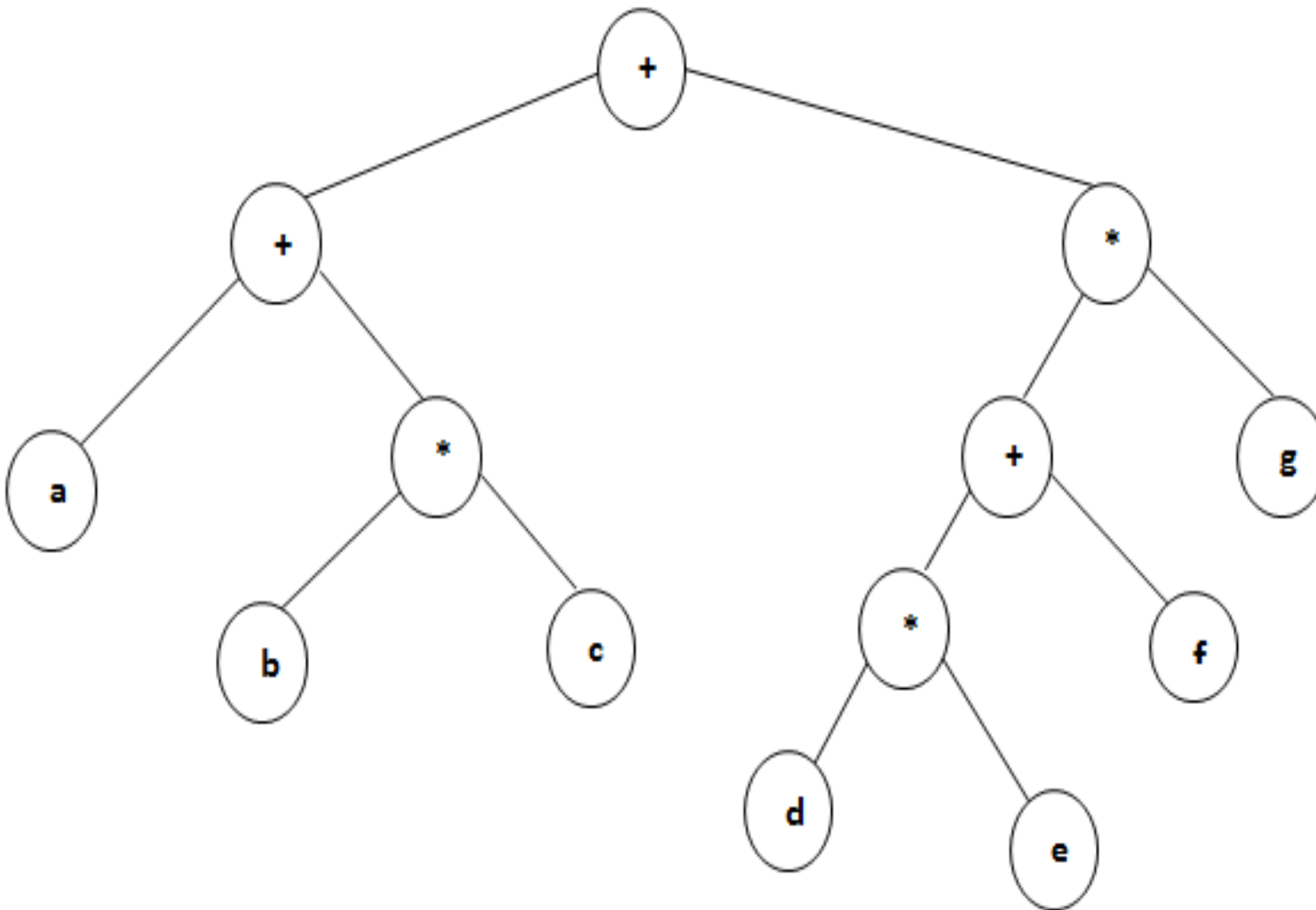
A node will have either two children or no child at all.





Expression Binary tree

- Expression trees are a special kind of binary tree used to evaluate certain expressions.
- Two common types of expressions that a binary expression tree can represent are algebraic and boolean.
- These trees can represent expressions that contain both unary and binary operators.
- The leaves of a binary expression tree are operands, such as constants or variable names, and the other nodes contain operators.
- Expression tree are used in most compilers.





Binary Tree Traversal

1. Preorder traversal:-In this traversal method first process root element, then left sub tree and then right sub tree.

Procedure:-

Step 1: Visit root node

Step 2: Visit left sub tree in preorder

Step 3: Visit right sub tree in preorder



2. Inorder traversal:-

In this traversal method first process left element, then root element and then the right element.

Procedure:-

Step 1: Visit left sub tree in inorder

Step 2: Visit root node

Step 3: Visit right sub tree in inorder



3. Postorder traversal:-

In this traversal first visit / process left sub tree, then right sub tree and then the root element.

Procedure:-

Step 1: Visit left sub tree in postorder

Step 2: Visit right sub tree in postorder

Step 3: Visit root node