Linked Representation:

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

Shortcut :

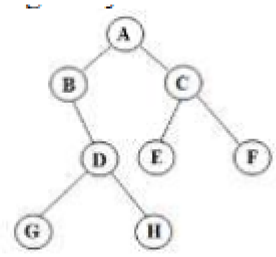
Pre Order	Print (Rt)	L	R
In Order	L	Print (Rt)	R
Post Order	L	R	Print (Rt)

Binary tree traversed:

- Preorder
- Inorder
- Postorder

Traversing a tree is a process of visiting every node of the tree and exactly once.

- **Preorder:**

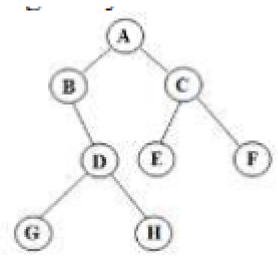


Order=ABDGHCEF

Steps:
1)Visit root node
2)Visit left subtree in preorder
3)Visit right subtree in preorder

```
void preorder (node *T)
{
if(T!=NULL)
{
printf(“\n %d”,T→data);
preorder(T→left);
preorder(T→right);
}
}
```

- **Inorder:**

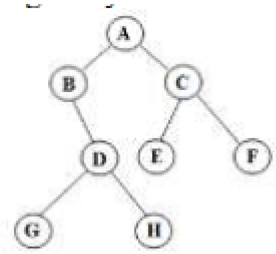


For above tree: Order =BGDHAECF

Steps:
1) Traverse left subtree in Inorder
2) Visit the root node
3) Traverse right subtree in Inorder

```
void inorder(node *T)
{
if(T!=NULL)
{
Inorder (T→left);
printf(“\n%d”,T→data);
Inorder(T→right);
}
}
```

- **Postorder:**

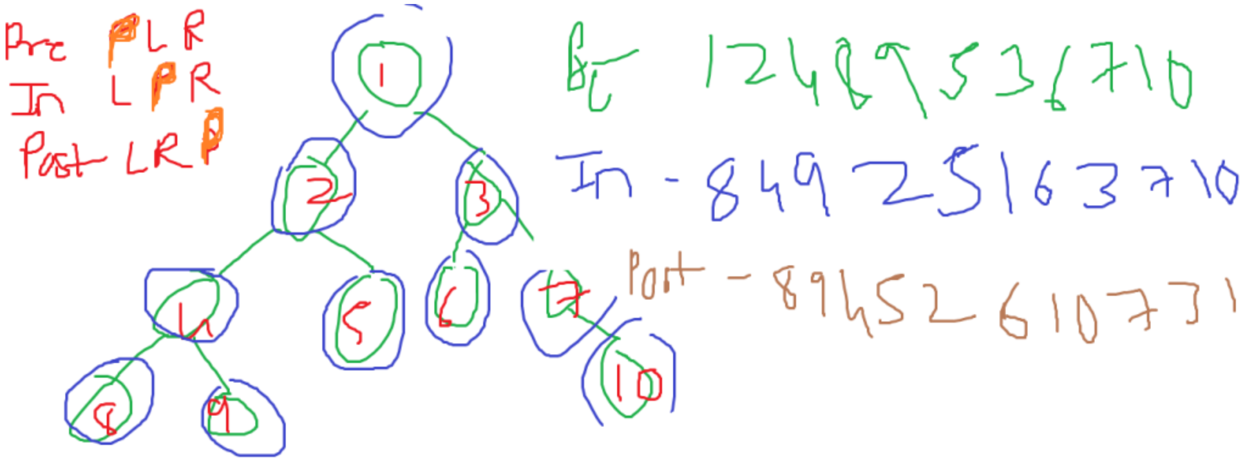


For above tree: Order =GHDBEFCA

Steps:
1) Traverse left subtree in postorder
2) Traverse right subtree in postorder
3) At last, visit the root node

```
void Postorder(node *T)
{
if(T!=NULL)
{
Postorder (T→left);
Postorder (T→right);
printf(“\n%d”,T→data);
}
}
```

Example :



Pre PLR
In LPR
Post LRP

Pre 1 2 4 8 9 5 3 6 7 10

In - 8 4 9 2 5 1 6 3 7 10

Post - 8 9 4 5 2 6 10 7 3 1