TREE TRAVERSALS

Traversing means visiting each node only once. Tree traversal is a method for visiting all the nodes in the tree exactly once.

There are three types of tree traversal techniques, namely:

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

1. In order Traversal

The inorder traversal of a binary tree is performed as

- Traverse the left subtree in order
- Visit the root
- Traverse the right subtree in order

Example

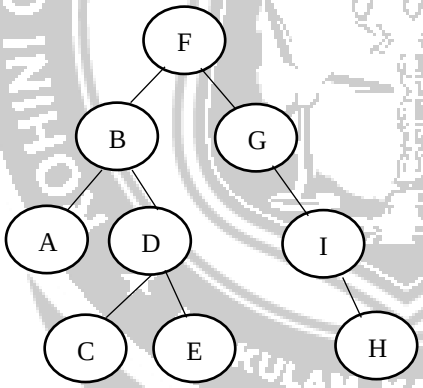

Fig 6. Binary Tree

In order traversal : A,B,C,D,E,F,,G,H,I

The inorder traversal of the binary tree for an arithmetic expression is in infix form.

Recursive Routine for inorder traversal

```
void inorder(Tree T)
{
if(T!=NULL)
{
inorder(T→left);
```

```
printf("%d",T→data);
inorder(T→right);
}
}
```

2. Pre order traversal

The pre order traversal of a binary tree is performed as follows:

- Visit the root
- Traverse the left subtree in pre order
- Traverse the right subtree in pre order

Preorder traversal of the Fig is

F,B,A,D,C,E,G,I,H

Recursive routine for Pre order traversal

```
void preorder(Tree T)
{
if(T!=NULL)
{
printf("%d",T→data);
preorder(T→left);
preorder(T→right);
}
}
```

3. Post order traversal

Post order traversal of a binary tree is performed by the following steps:

- Traverse the left subtree in post order
- Traverse the right subtree in post order
- Visit the root

Post order traversal of the Fig is

A,C,E,D,B,H,I,G,F

Recursive routine for Post order traversal

```
void postorder(Tree T)
{
if(T!=NULL)
{
postorder(T→left);
postorder(T→right);
printf("%d",T→data);
}
}
```