



DATA STRUCTURES AND ITS APPLICATIONS

UE19CS202

Shylaja S S & Kusuma K V

Department of Computer Science
& Engineering

DATA STRUCTURES AND ITS APPLICATIONS

n-ary Tree Traversal

Shylaja S S

Department of Computer Science & Engineering

DATA STRUCTURES AND ITS APPLICATIONS

Tree Traversal



Structure of a treenode revisited

```
struct treenode{  
    int info;  
    struct treenode *child;  
    struct treenode *sibling;  
};
```

With the treenode implemented as having pointers to first child and immediate sibling, the traversal preorder, inorder and postorder for a tree are defined as follows:

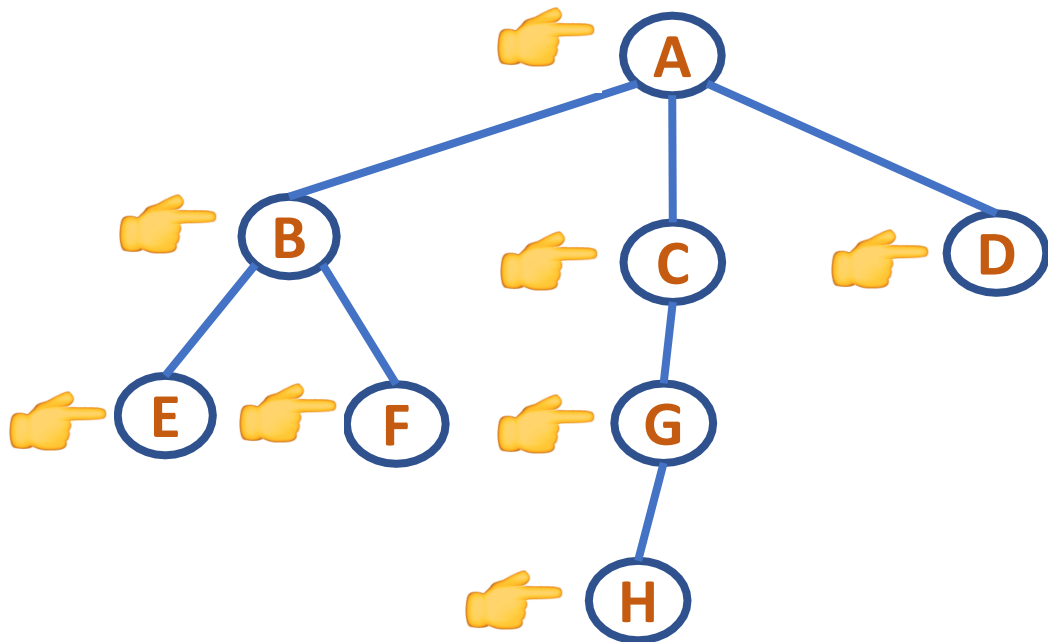
Preorder:

1. Visit the root of the first tree in the forest
2. Traverse in preorder the forest formed by the subtrees of the first tree, if any
3. Traverse in preorder the forest formed by the remaining trees in the forest, if any

DATA STRUCTURES AND ITS APPLICATIONS

Tree Traversal

Preorder Tree Traversal



ABEFCGHD

DATA STRUCTURES AND ITS APPLICATIONS

Tree Traversal



```
void preorder(TREE *root)
{
    if(root!=NULL)
    {
        printf(" %d ",root->info);
        preorder(root->child);
        preorder(root->sibling);
    }
}
```

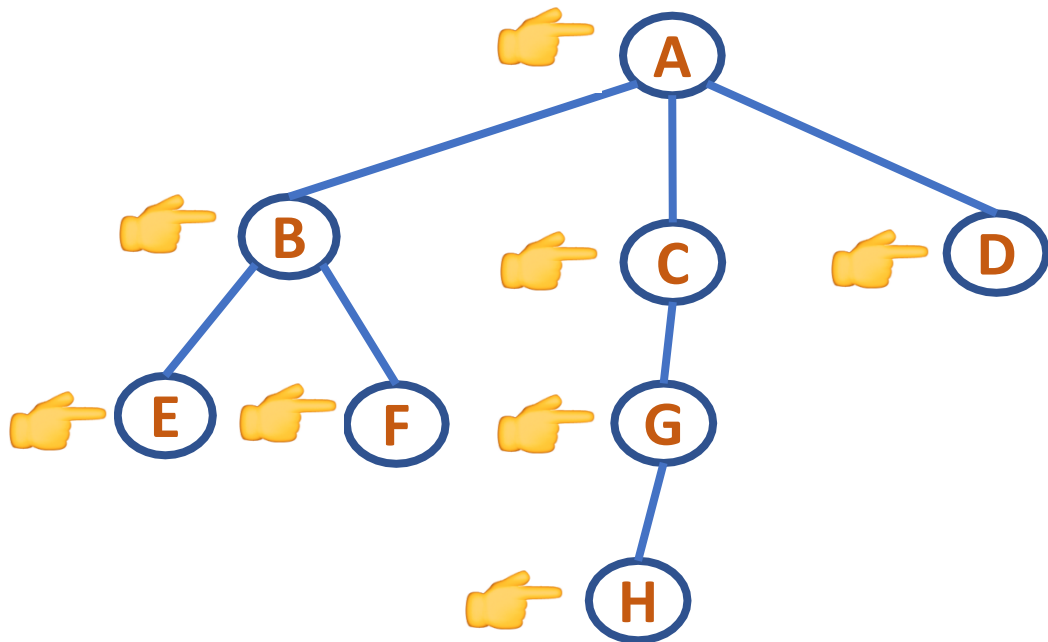
Inorder

1. Traverse in inorder the forest formed by the subtrees of the first tree, if any
2. Visit the root of the first tree in the forest
3. Traverse in inorder the forest formed by the remaining trees in the forest, if any

DATA STRUCTURES AND ITS APPLICATIONS

Tree Traversal

Inorder Tree Traversal



E F B H G C D A

DATA STRUCTURES AND ITS APPLICATIONS

Tree Traversal



```
void inorder(TREE *root)
{
    if(root!=NULL)
    {
        inorder(root->child);
        printf(" %d ",root->info);
        inorder(root->sibling);
    }
}
```

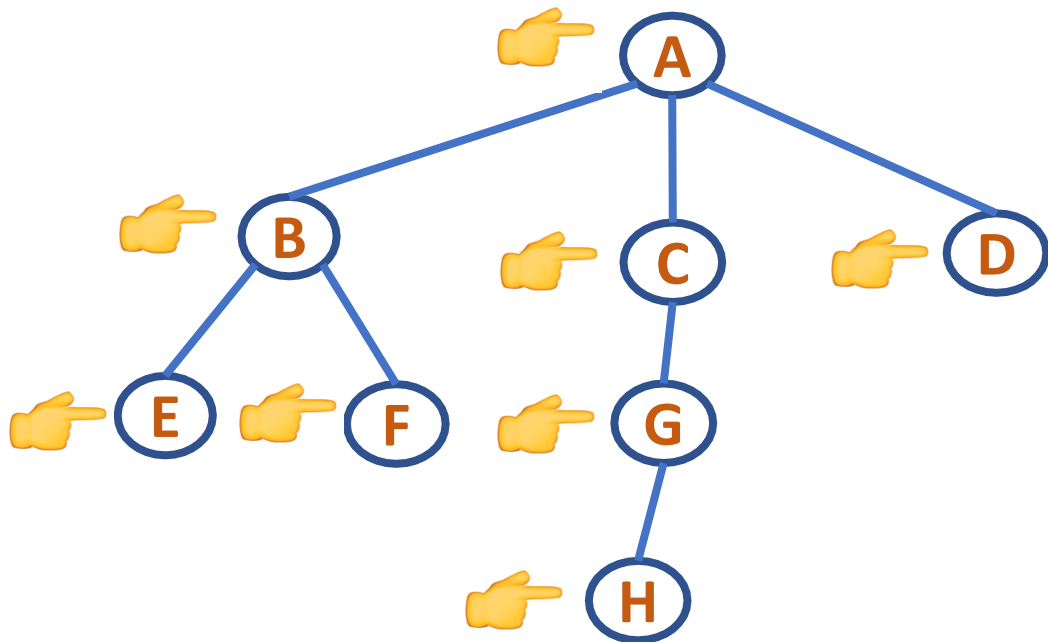
Postorder

1. Traverse in postorder the forest formed by the subtrees of the first tree, if any
2. Traverse in postorder the forest formed by the remaining trees in the forest, if any
3. Visit the root of the first tree in the forest

DATA STRUCTURES AND ITS APPLICATIONS

Tree Traversal

Postorder Tree Traversal



F E H G D C B A

DATA STRUCTURES AND ITS APPLICATIONS

Tree Traversal



```
void postorder(TREE *root)
{
    if(root!=NULL)
    {
        postorder(root->child);
        postorder(root->sibling);
        printf(" %d ", root->info);
    }
}
```



THANK YOU

Shylaja S S

Department of Computer Science
& Engineering

shylaja.sharath@pes.edu