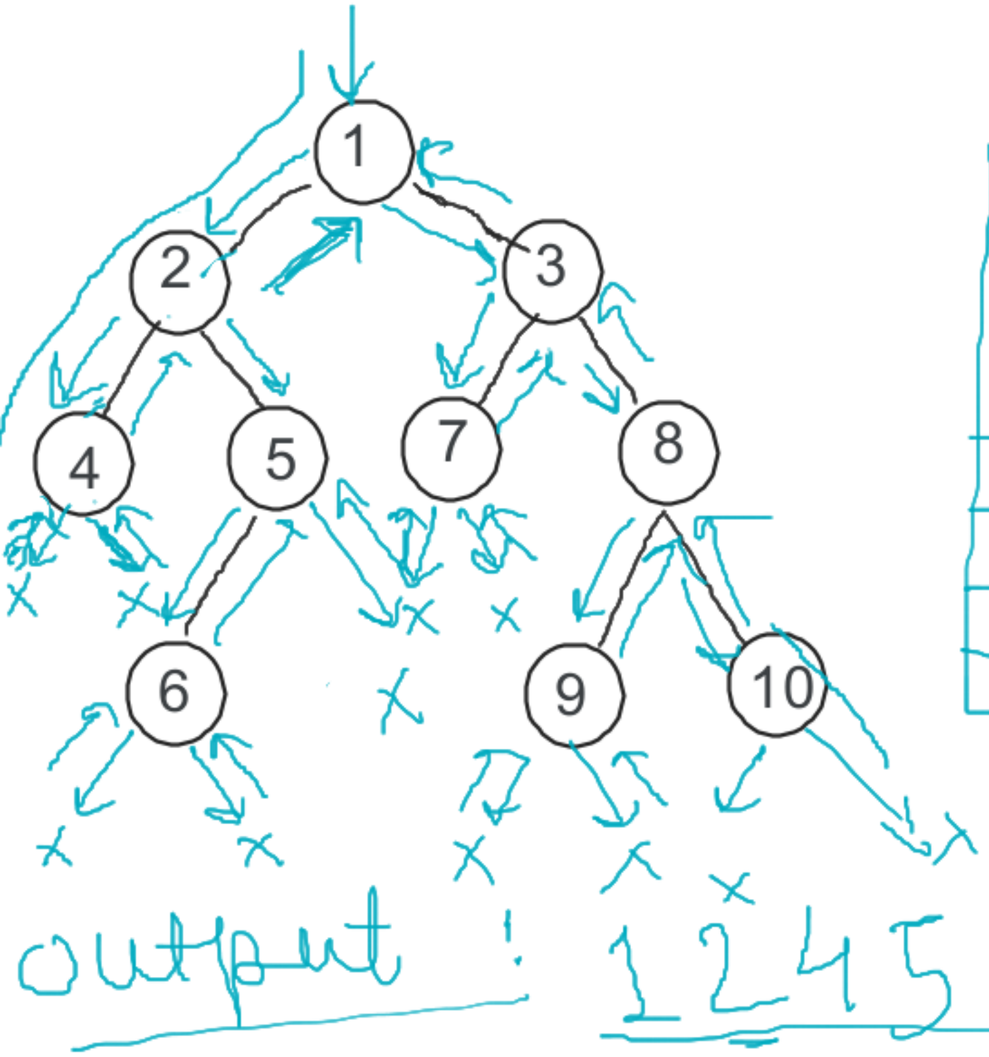
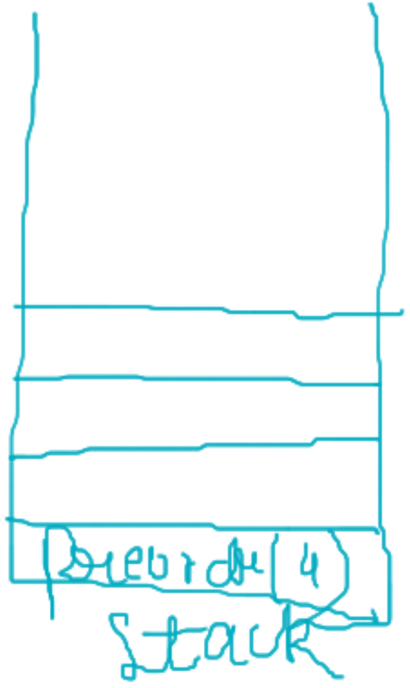
PreOrder Traversal

[Condition ->Root Left Right]



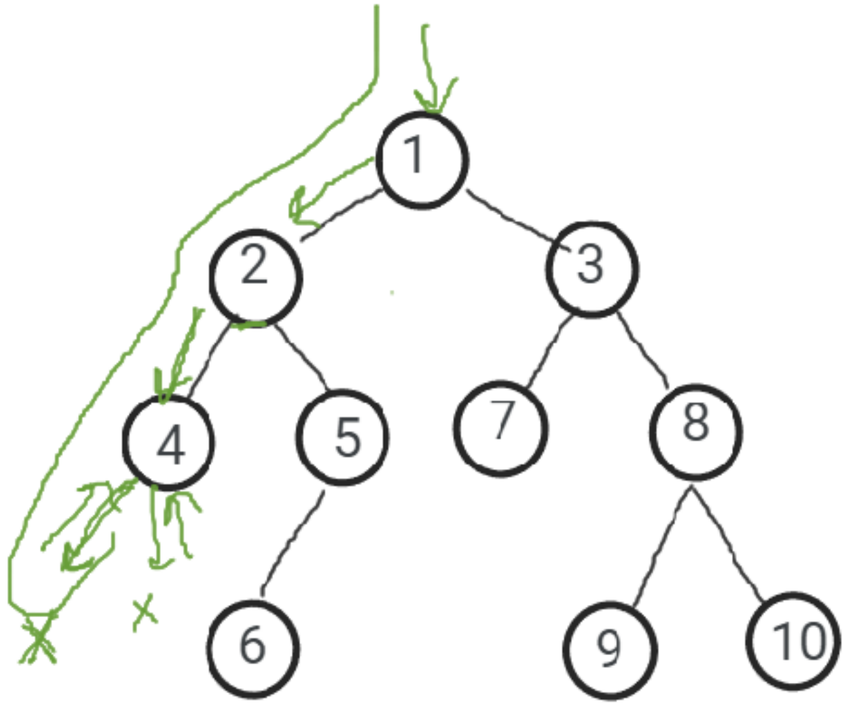



```
void preorder (node){
if(node == null)
    return;

print(node ->data)
  preorder(node->left)
  preorder(node->right)
  }
```

output : 1 2 4 5 6 3 7 9 10

InOrder Traversal

[Condition ->Left Root Right]

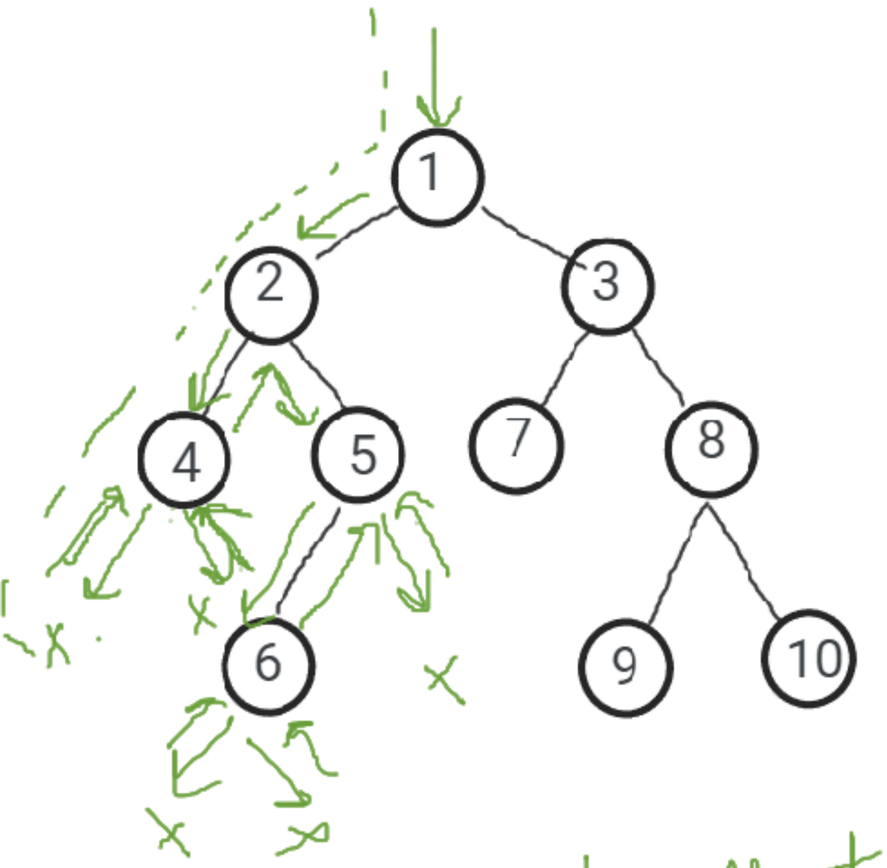



```
void inorder (node){
if(node == null)
    return;

    inorder(node->left)
    print(node ->data)
    inorder(node->right)
    }
```

Output: 4 2 6 5 1 7 3 9 8 10

PostOrder Traversal

[Condition ->Left Right Root]



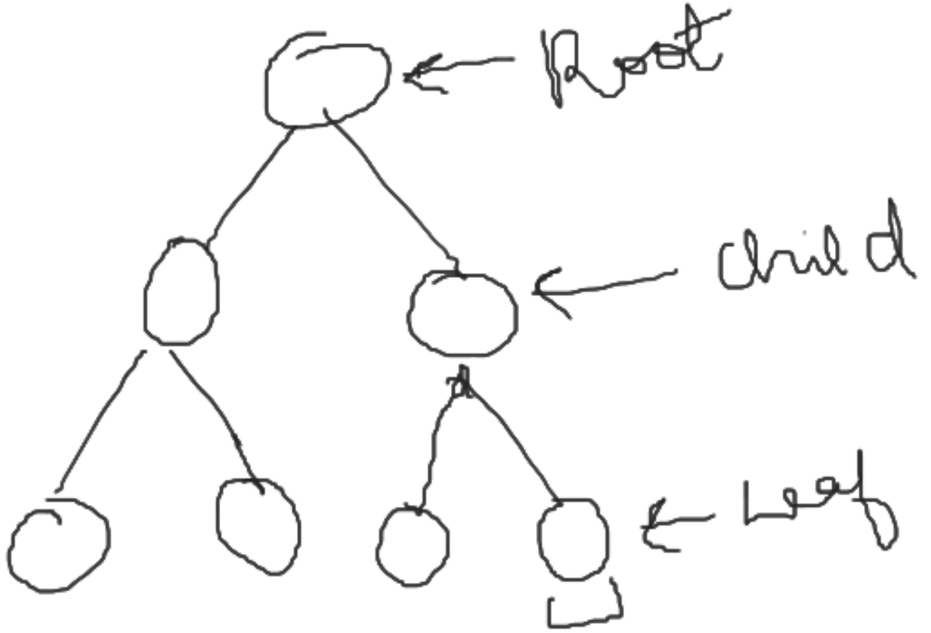

```
void postorder (node){
if(node == null)
    return;

    postorder(node->left)
    postorder(node->right)
    print(node ->data)
}
```

output : 4 6 5 2 7 9 10 8 3 1

Traversal Techniques:

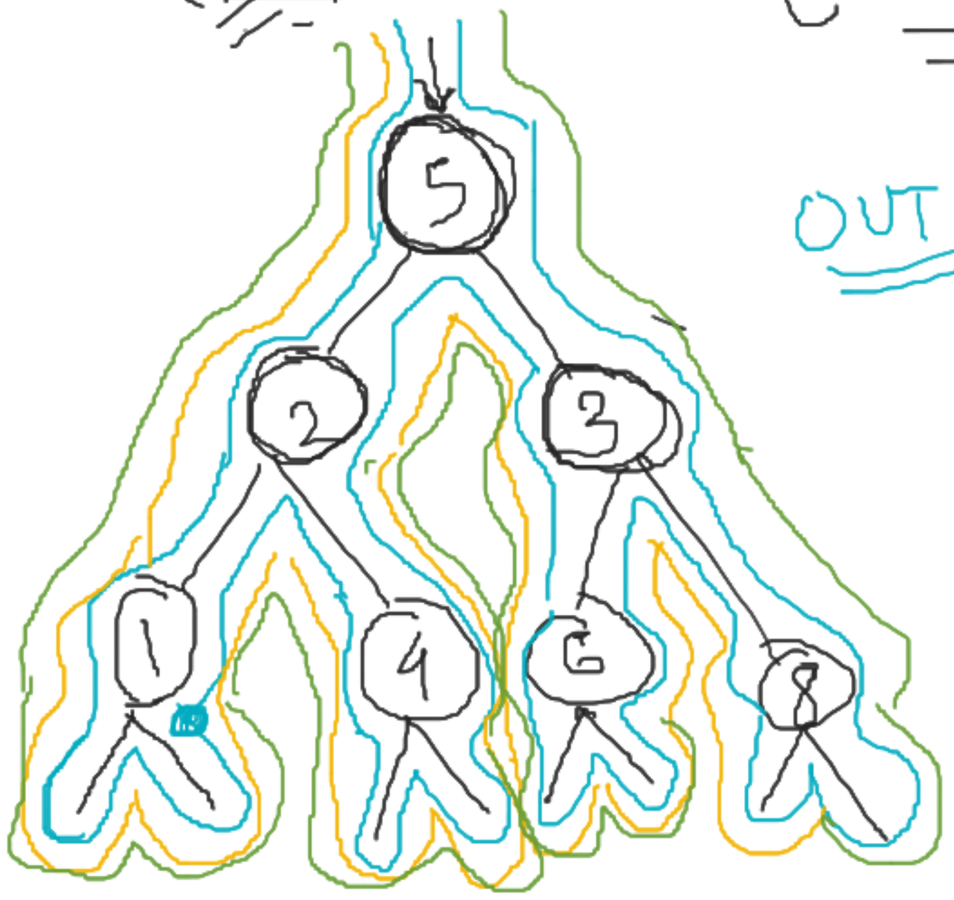
1. Depth First Search
    1.1 Preorder
    1.2 Inorder
    1.3 Postorder

2.Breadth First Search



1. Preorder { Root Left Right }

1426835



output: 5214368

2. Inorder { Left Root Right }

1 2 5 6 3 8

1 2 4 5 3 6 8

Output : 1245638

3 PostOrder :
[Left Right Root]

1426835