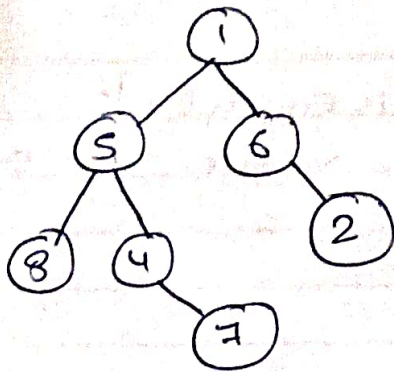


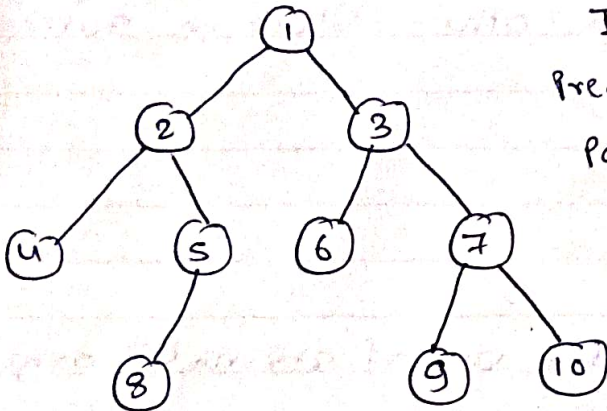
Traversal of Tree



Inorder [8, 5, 4, 7, 1, 6, 2]

Preorder [1, 5, 8, 4, 7, 6, 2]

PostOrder [8, 7, 4, 5, 2, 6, 1]



Inorder [4, 2, 8, 5, 1, 6, 3, 9, 7, 10]

Preorder [1, 2, 4, 5, 8, 3, 6, 7, 9, 10]

Post Order [4, 8, 5, 2, 6, 9, 10, 7, 3, 1]

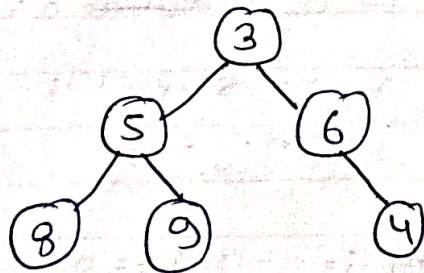
Inorder :- left \rightarrow Root \rightarrow right.

Preorder :- root \rightarrow left \rightarrow right

Post Order :- left \rightarrow right \rightarrow root.

Now, we will gonna dry run one example using recursion.

consider the Tree :-



Inorder [8, 5, 9, 3, 6, 4]

We will only do dry run for Inorder Traversal.

Tracing each step of recursion
for clear understanding.

Function Inorder (root, arraylist arr)

{

if (root == null)

{

return

}

// root = 3

Inorder (root.left, arr); // 5, arr.

arr.add (root.val)

Inorder (root.right, arr);

}

⇓

Inorder (5, arr)

{

(8, arr) →
arr.add (root.val)

~~// 8 is null so~~
~~in arr add~~

Inorder (root.right, val);

}

Inorder (8, arr)

{

(8.left, arr)

// 8.left == null
so, return.

arr.add (8) // It will add 8 in

Inorder (8.right, arr) our empty arraylist

Inorder (8.right, arr)

{

(8.right == null)

{ return;

}

Inorder(s, arr)

{

Inorder(s, arr); // It is done
arr.add(root.val) list has one
// It will add 5 also. element 8

Inorder(s.left, arr)

arr = [8, 5]

Inorder(s.left, arr)

{

// because the recursion will go one step
return back with new

arr.add(9);

// Recursion will go right, but it is now.

}

arr = [8, 5, 9,

Inorder(3, arr)

{

Inorder(3.left, arr); // It is done.

arr.add(3); [8, 5, 9, 3]

Inorder(3.right, arr);

}

Inorder(6, arr)

{

(6.left == null) so, it will return.

arr.add(6); [8, 5, 9, 3, 6].

~~arr.add~~

Inorder(6.right, arr)

}

Inorder(4, arr)

{

its left is null.

arr.add(4); // [8, 5, 9, 3, 6, 4]

its right is also null.

}

Now, this function will be removed from stack,

Each function will be removed step by step.

Inorder(3, arr)

{

Inorder(3.left, arr) // done

arr.add(3); // done.

Inorder(3.right, arr) // done

So, it will return to main fun^c

where arr has all the nodes

[8, 5, 9, 3, 6, 4]

}