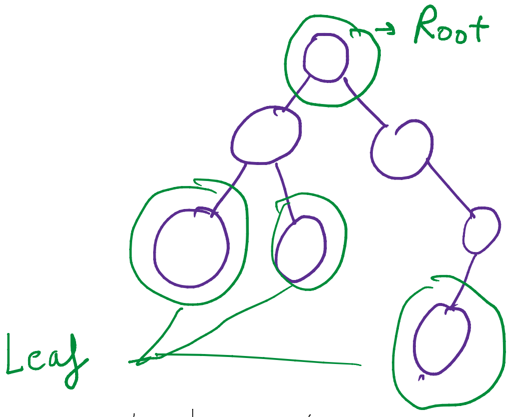
1 November

01 November 2022 21:11

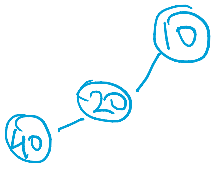
Trees => Binary Tree
1 0

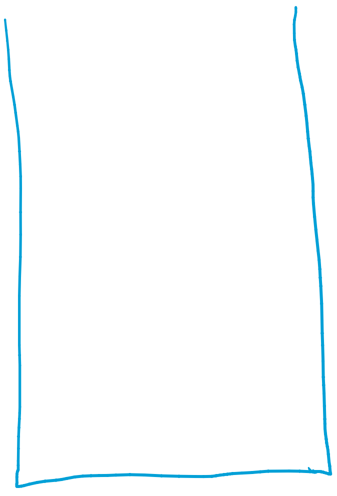
Node{
int data;
Node left;
Node right;
}



[10, 20, 40, -1, -1, 50, 80, -1, -1, -1, 30, 60, -1, 90, -1, -1, 70, -1, -1]

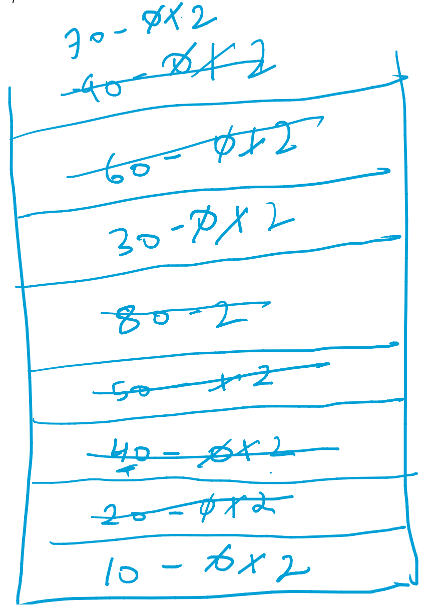
TreeConstructHelp{
Node node;
int state — 0 - left child
1 - Right
2 - Processing done
}



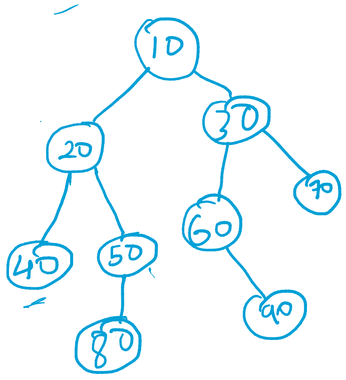


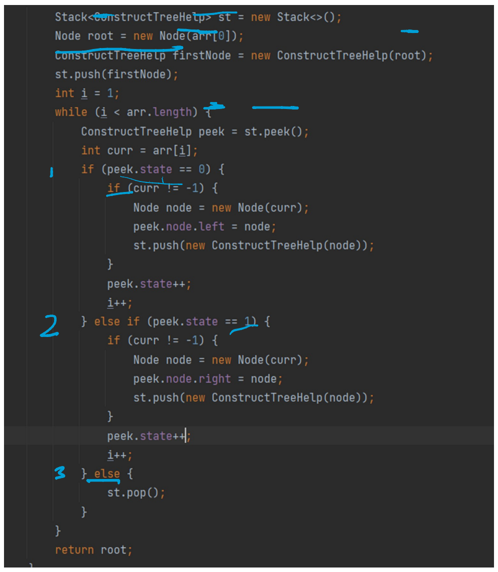
- increase st.peek().state
- if(arr[i] != -1) {
- add the new node
} else add null

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
10	20	40	-1	-1	50	80	-1	-1	-1	30	60	-1	90	-1	-1	70	-1	-1



i: 17 curr: 70





```
Stack<ConstructTreeHelp> st = new Stack<>();
Node root = new Node(arr[0]);
ConstructTreeHelp firstNode = new ConstructTreeHelp(root);
st.push(firstNode);
int i = 1;
while (i < arr.length) {
    ConstructTreeHelp peek = st.peek();
    int curr = arr[i];
1   if (peek.state == 0) {
        if (curr != -1) {
            Node node = new Node(curr);
            peek.node.left = node;
            st.push(new ConstructTreeHelp(node));
        }
        peek.state++;
        i++;
2   } else if (peek.state == 1) {
        if (curr != -1) {
            Node node = new Node(curr);
            peek.node.right = node;
            st.push(new ConstructTreeHelp(node));
        }
        peek.state++;
        i++;
3   } else {
        st.pop();
    }
}
return root;
```

$$\boxed{10 - \cancel{6} \times 2}$$

```
        } else {
            st.pop();
        }
    }
    return root;
```

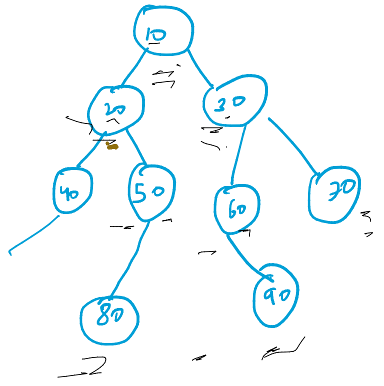
Display of BT

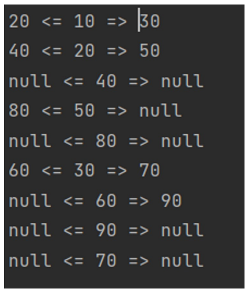
Pre-order

CLR => 10, 20, 40, 50, 80, 30, 60, 90, 70

```
displayPre(root){
    if(root==null) return;
    Sout(root);

    displayPre(root.left)
    displayPre(root.Right);
}
```



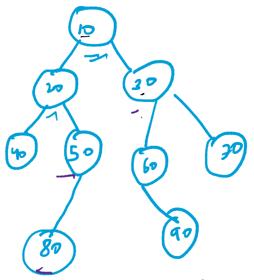


```
20 <= 10 => 30
40 <= 20 => 50
null <= 40 => null
80 <= 50 => null
null <= 80 => null
60 <= 30 => 70
null <= 60 => 90
null <= 90 => null
null <= 70 => null
```

20 <= 10 => 30

In-Order

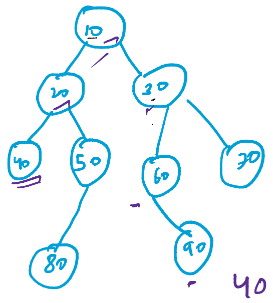
LCR



40, 20, 80, 50, 10, 60, 90, 30, 70

```
displayIn(root){
    if(root==null) return;
    displayIn(root.left)
    Sout(root);
    displayIn(root.Right);
}
```

Post-Order LRC

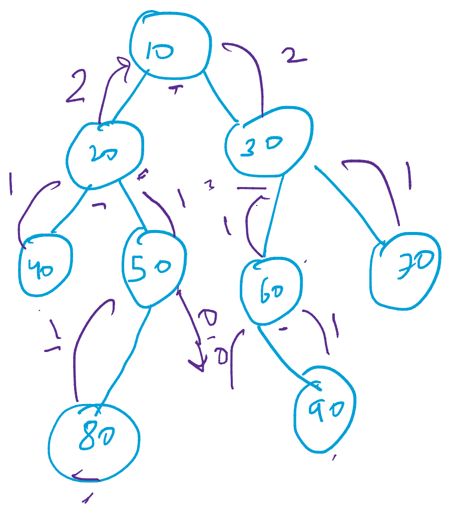


40, 80, 50, 20, 90, 60, 70, 30, 10

```
displayPost(root){
    if(root==null) return;

    displayPost(root.left)
    displayPost(root.Right);
    Sout(root)
}
```

4


10
20
30
40
50
60
70
80
90
2
2
1
1
0
0