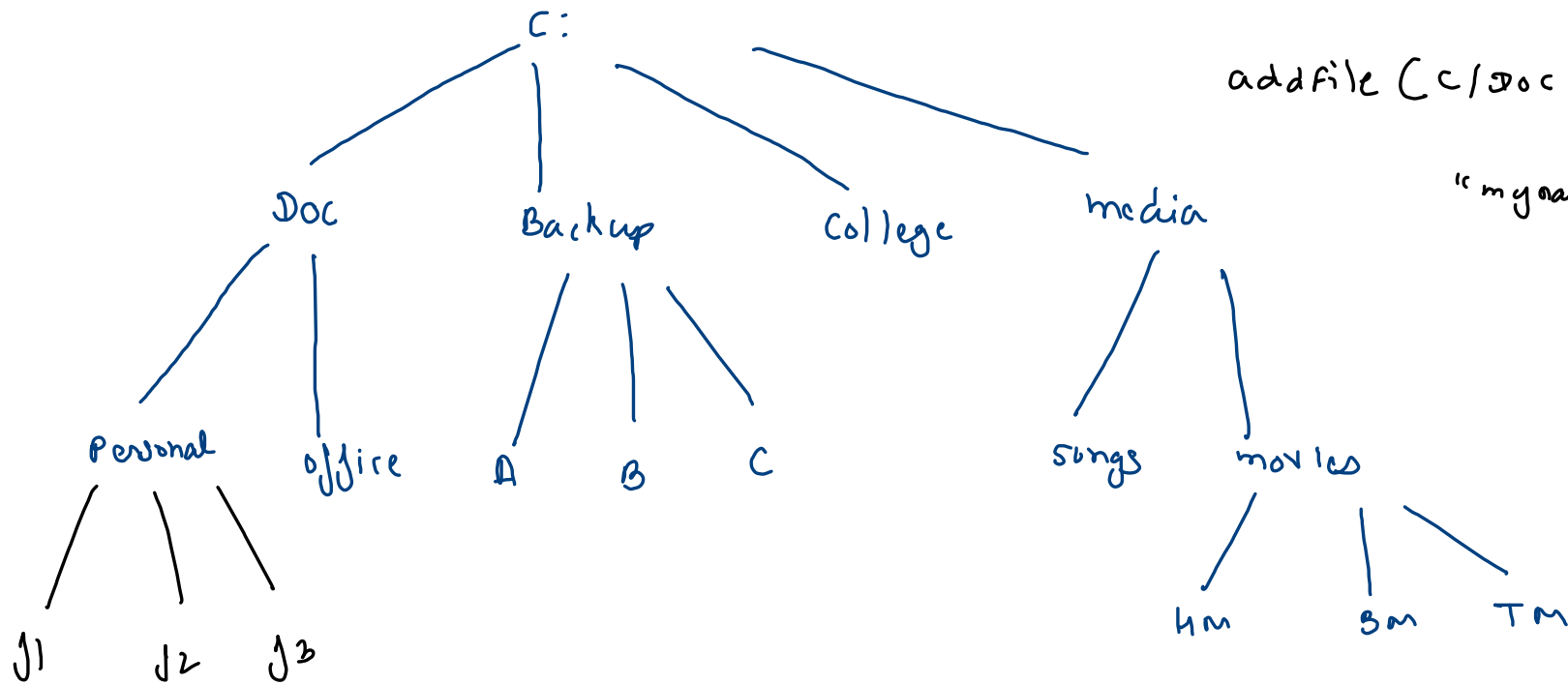


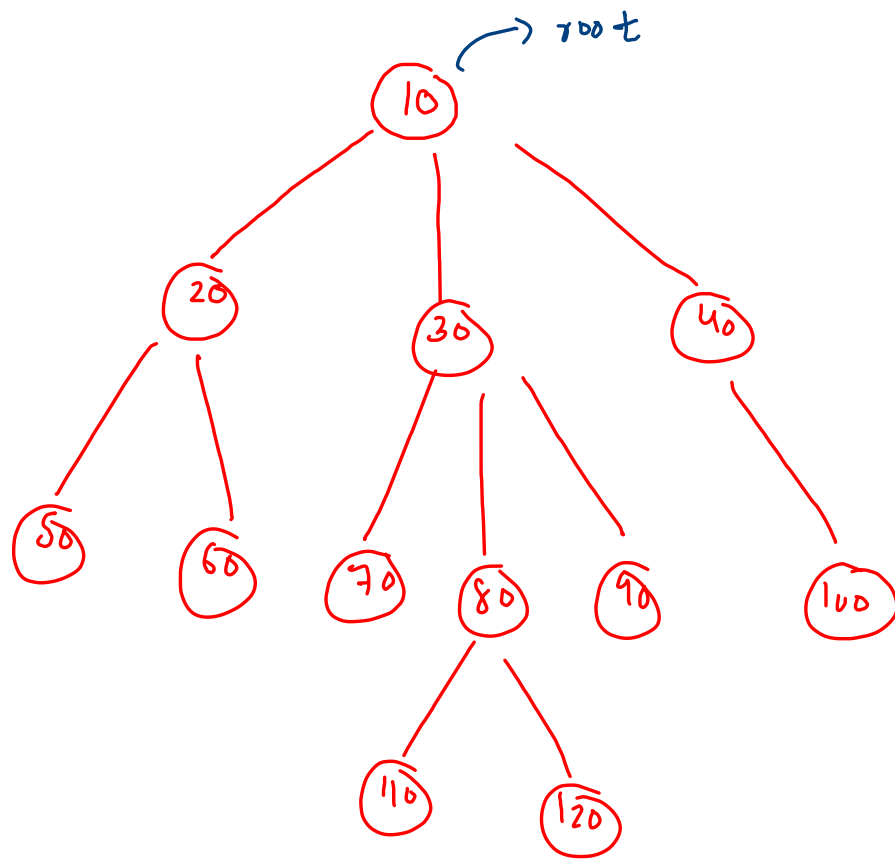
Generic tree

File system

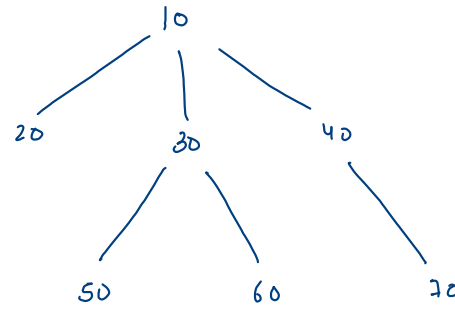


addFile (C/Doc/Personal/J1,

"my name is this");



```
Node {  
    int data;  
  
    ArrayList<Node> children;  
}
```



```

public static class Node {
    int data;
    ArrayList<Node> children new ArrayList();

    Node() {
    }

    Node(int data) {
        this.data = data;
    }
}

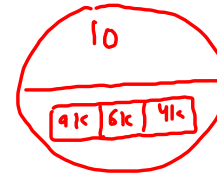
public static void main(String[] args) {
    int[] arr = {"ASSUME INPUT"};

    Node root = construct(arr);

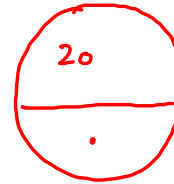
    System.out.println(root.data);
    System.out.println(root.children.size());
}

```

Root = 21k



21k



91k



61k



41k



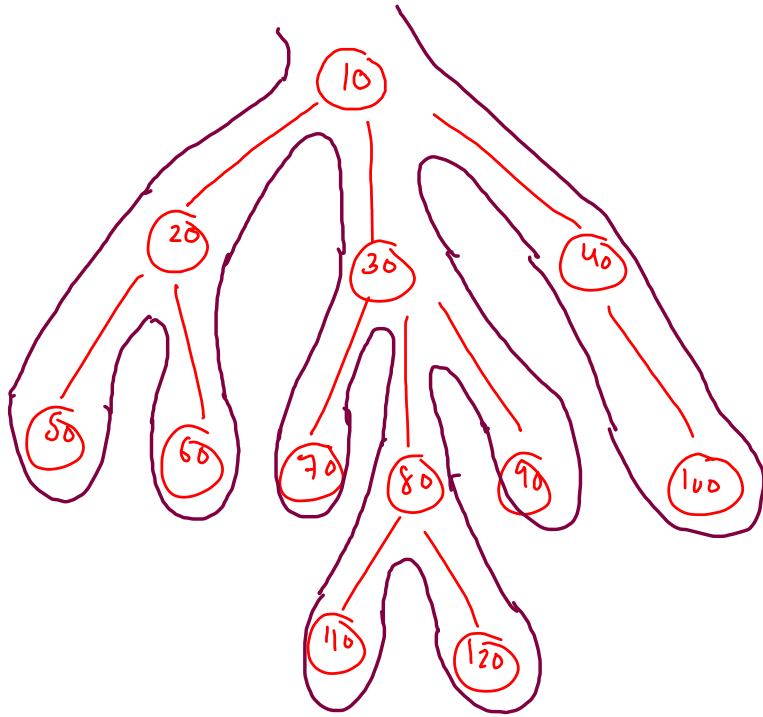
111k



161k



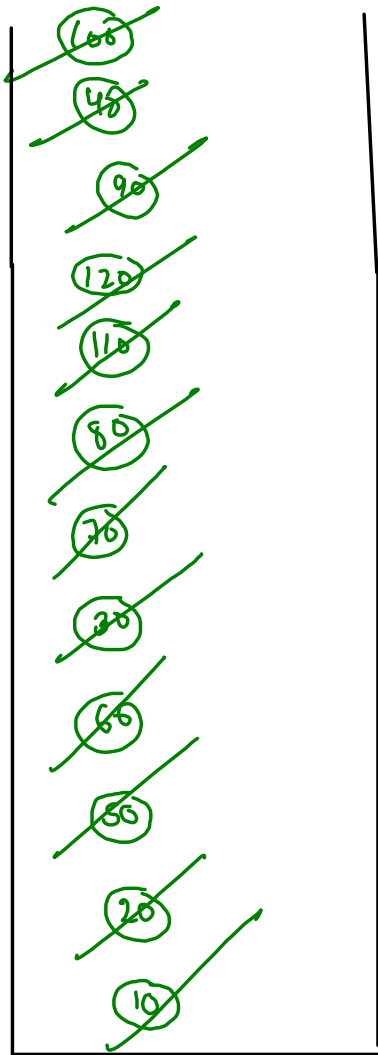
181k



node pre area  $\rightarrow$  node, data

node post area  $\rightarrow$  -

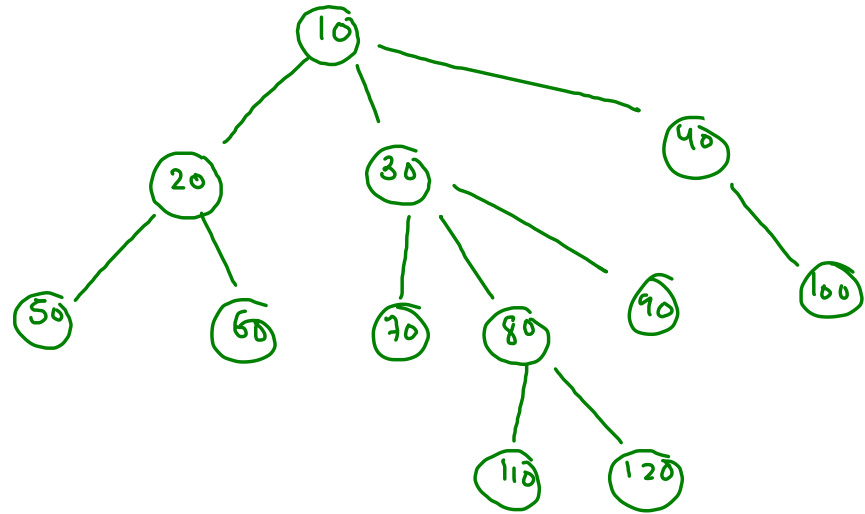
arr: 10 20 50 - 60 - - 30 70  
 - 80 110 - 120 - - 90 -  
 - 40 100 - - -



arr: 10 20 50 -1 60 -1 -1 30 70

-1 80 110 -1 120 -1 -1 90 -1

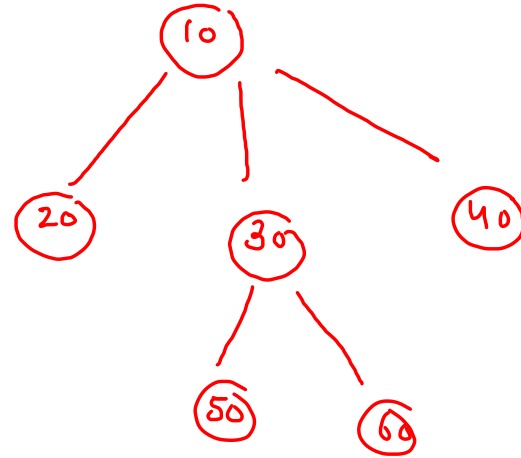
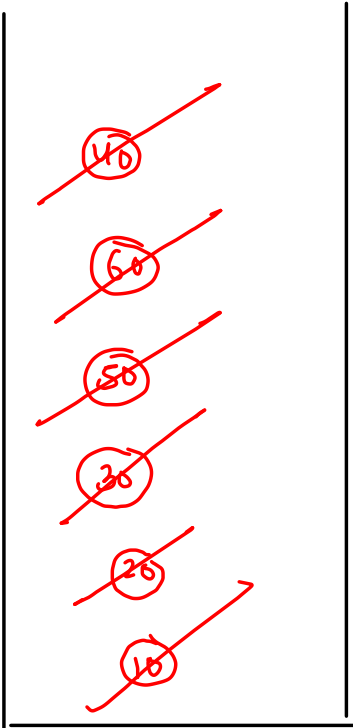
-1 40 100 -1 -1 -1 .  
i



root = 10

arr : 10 20 -1 30 50 -1 60 -1 -1 40 -1 -1

?



root = 10

10 20 50 -1 60 -1 -1 30 70 -1 80 110 -1 120 -1 -1 90 -1

-1 40 100 -1 -1 -1

i

```
public static Node construct(int[] arr) {
    Stack<Node> st = new Stack<>();

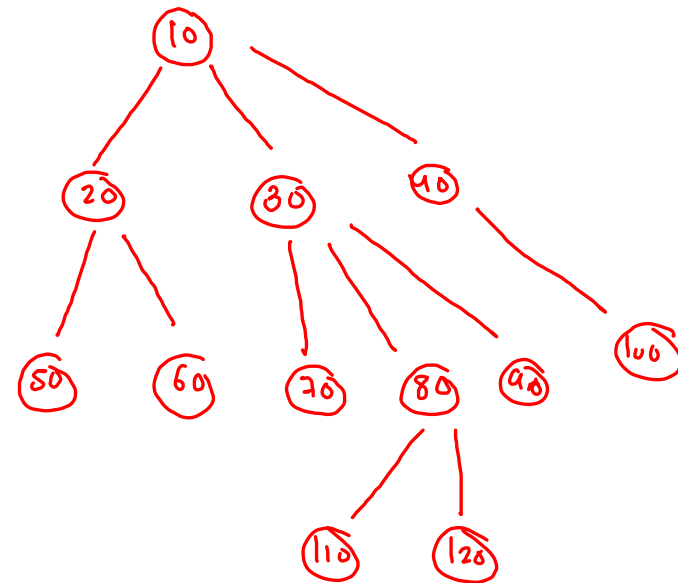
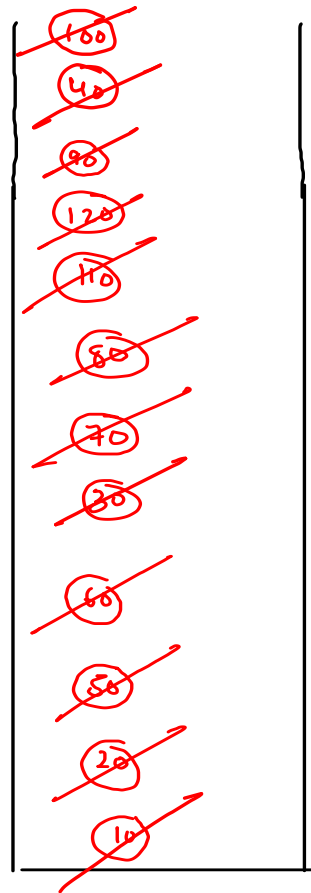
    Node root = null;

    for(int i=0; i < arr.length; i++) {
        if(arr[i] == -1) {
            st.pop();
        }
        else {
            Node nn = new Node(arr[i]);

            if(st.size() == 0) {
                //nn is the root node
                root = nn;
            }
            else {
                Node par = st.peek();
                par.children.add(nn);
            }

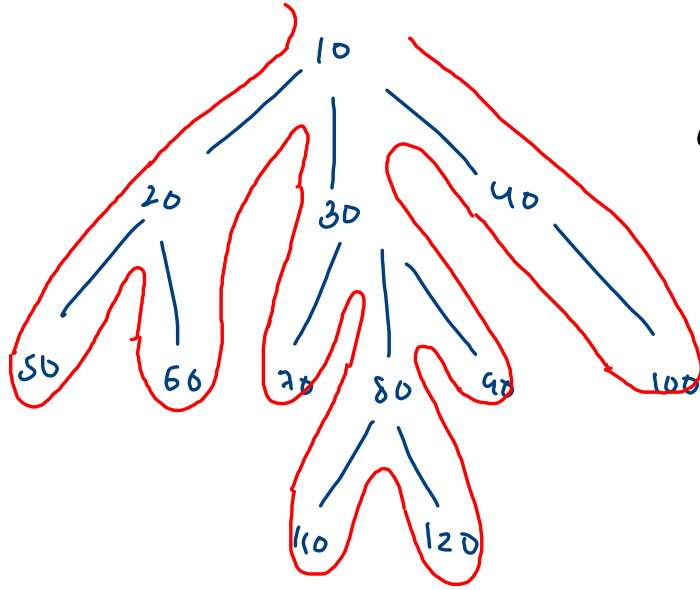
            st.push(nn);
        }
    }

    return root;
}
```



~~root = null~~

display()



display(10)

10 → 20, 30, 40

20 → 50, 60

50 →

60 →

30 → 70, 80, 90

70 →

80 → 110, 120

110 →

120 →

90 →

40 → 100

100 →

display(10)

→ 10 → 20, 30, 40

display(20),

display(30),

display(40).



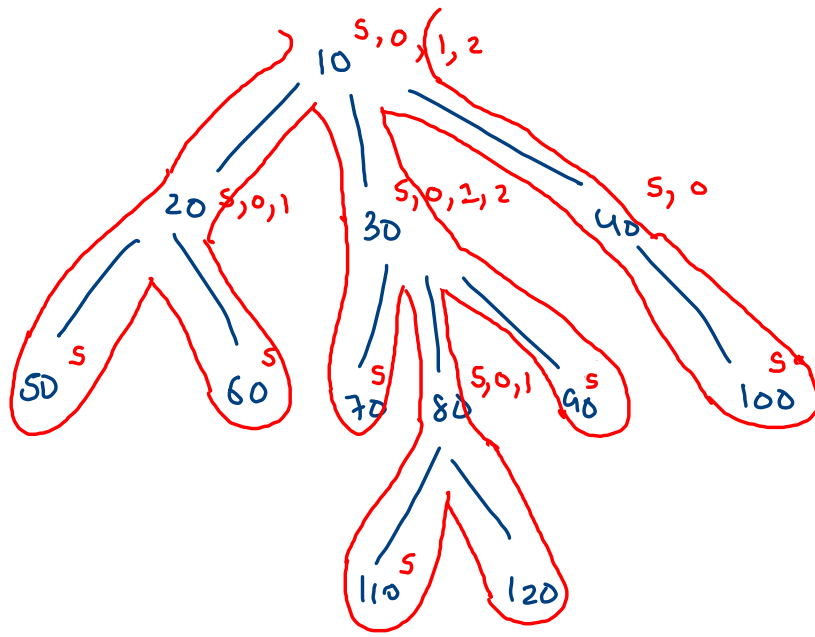
```

public static void display(Node node) {
    //self task
    System.out.print(node.data + " -> ");
    for(int i = 0; i < node.children.size(); i++) {
        Node child = node.children.get(i);
        System.out.print(child.data + ", ");
    }
    System.out.println(".");

    for(int i = 0; i < node.children.size(); i++) {
        Node child = node.children.get(i);
        display(child);
    }
}

```

S -> self task



10 -> 20, 30, 40, .

20 -> 50, 60, .

50 -> .

60 -> .

30 -> 70, 80, 90, .

70 -> .

80 -> 110, 120, .

110 -> .

120 -> .

90 -> .

40 -> 100, .

100 -> .