

odd ←
 chis add
 reverse

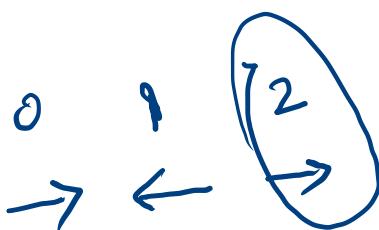
main

100	90	80	.
20	30	40	

helper

40	30	20
80	90	100

flag

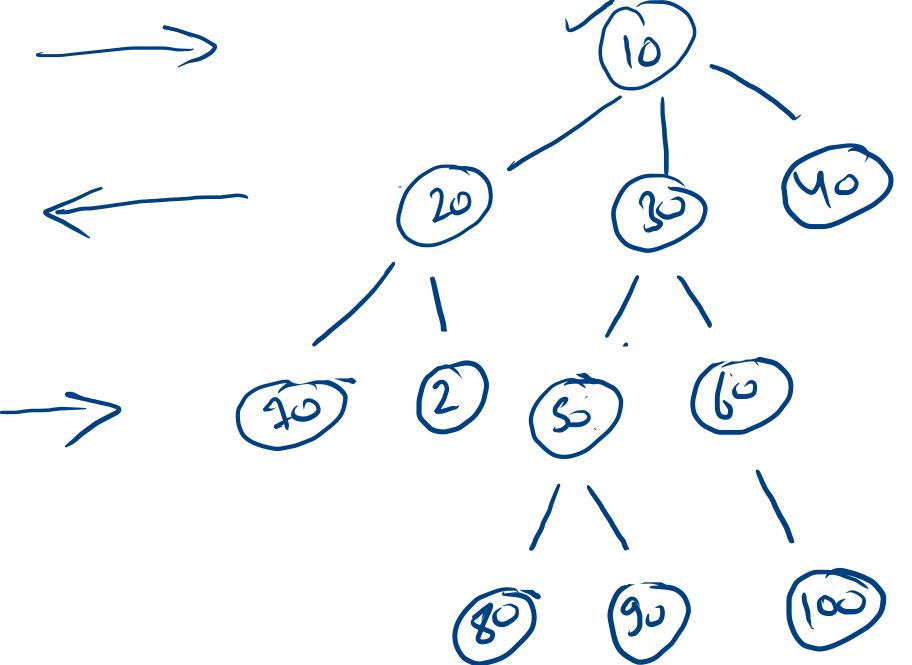


main Q

remove
prev

flag even
add 0 to w
else n-1 to 0

40 30 20
70 2 50 60



class

even

$0 \leq h-1$

odd
 $h-1 \geq 0$

main2

st

class

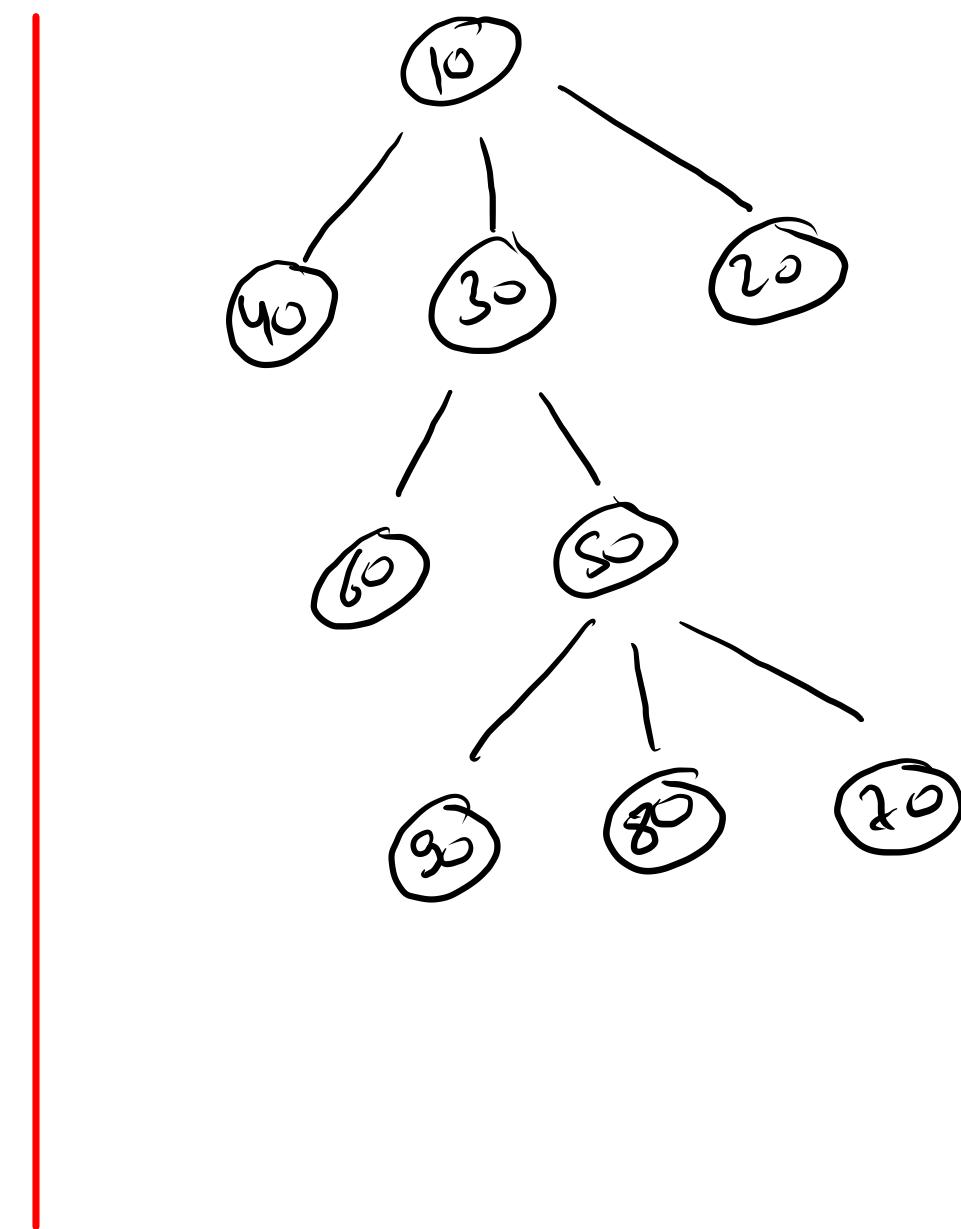
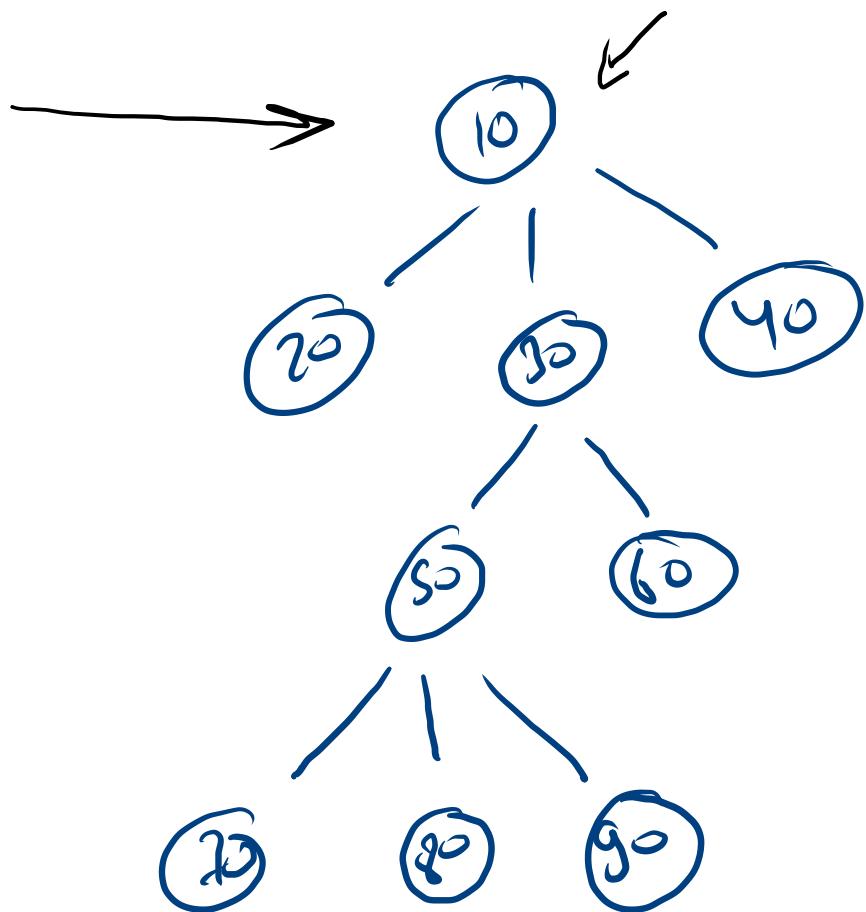
~~20 | 21 5 > 160~~

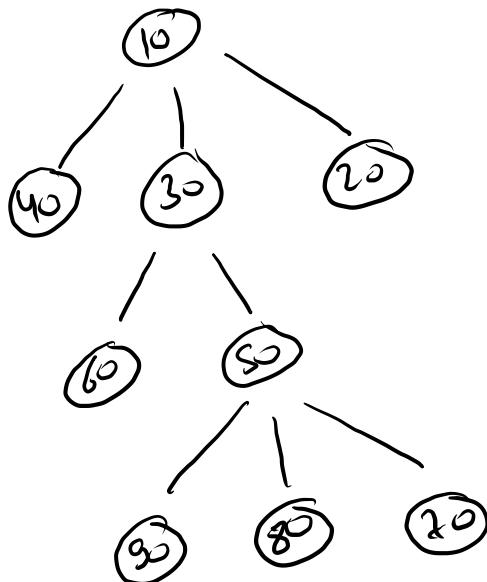
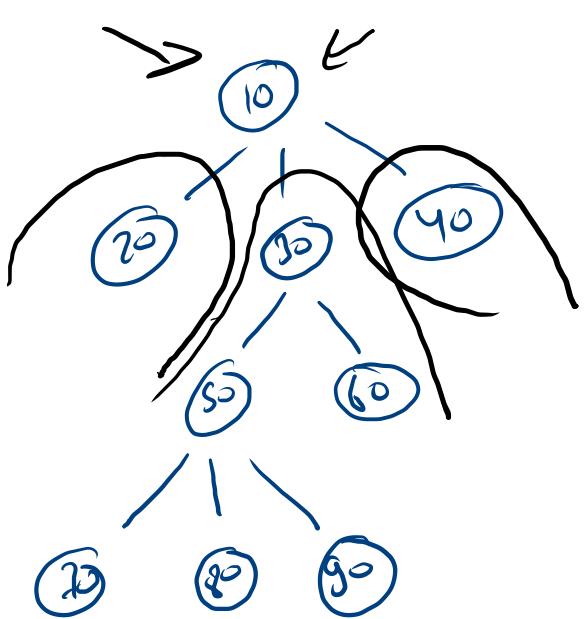
~~20 | 20 | 20 | 60 | 50 | 2 | 20~~

$\emptyset \rightarrow 1$

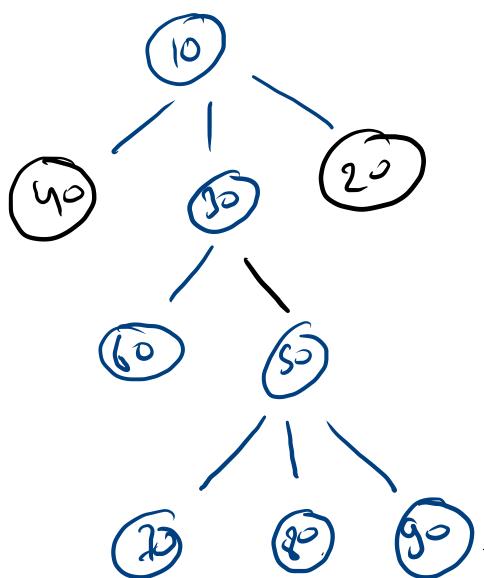
$10 \rightarrow 40 \rightarrow 30 \rightarrow 20$

GT



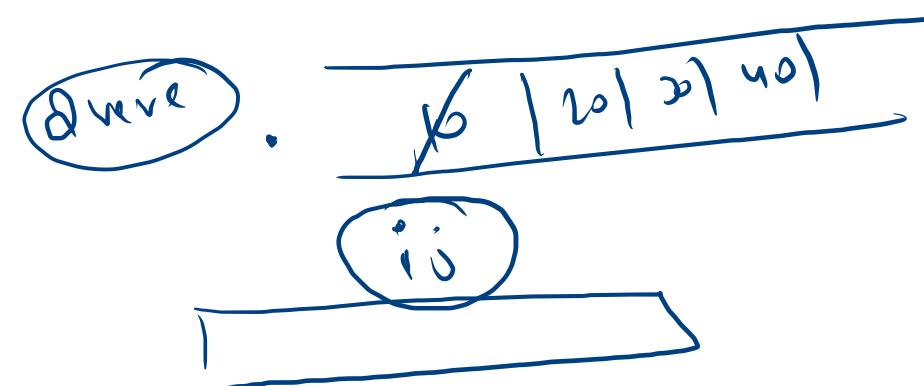
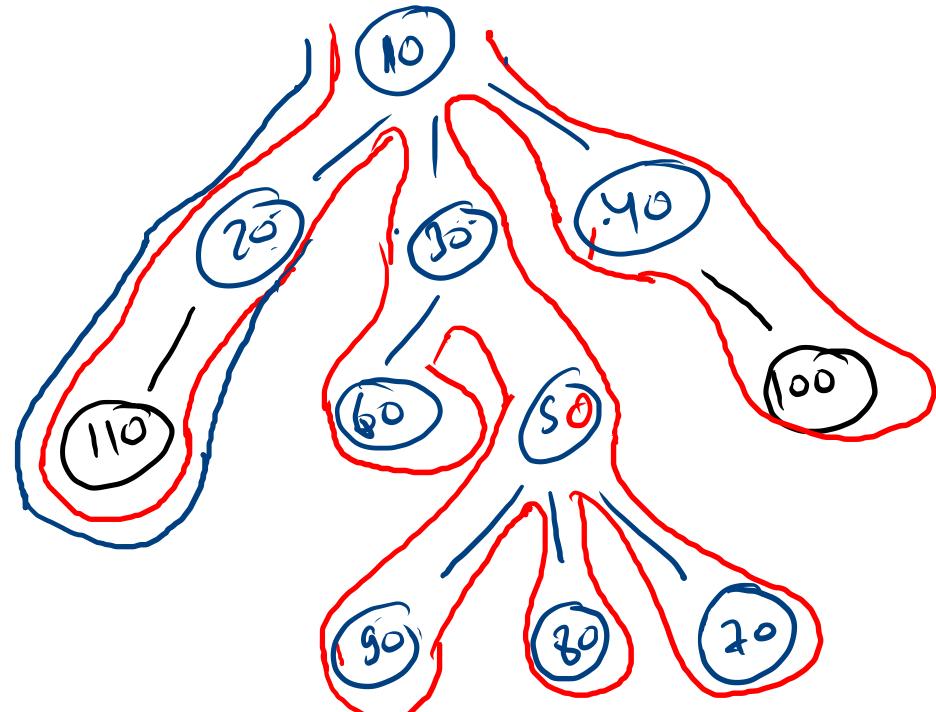
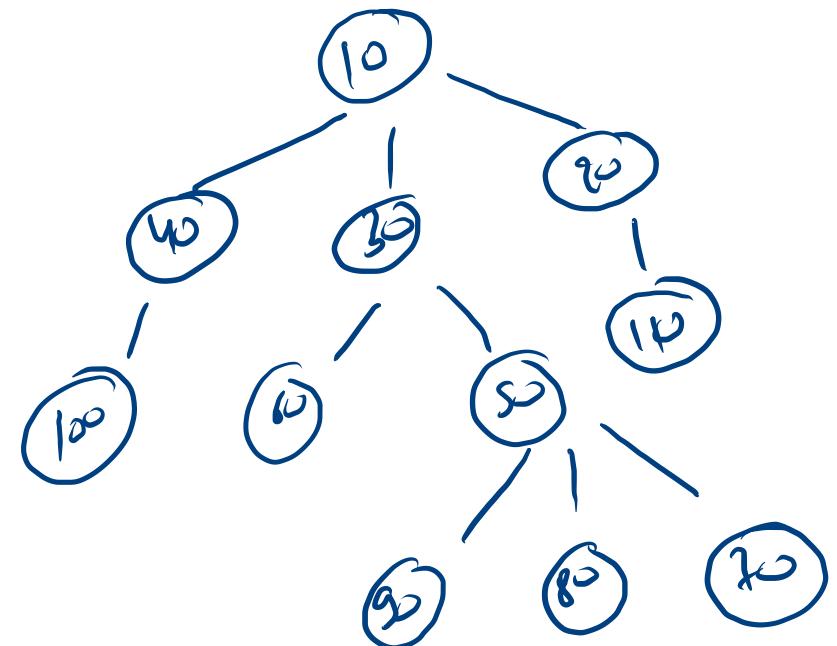


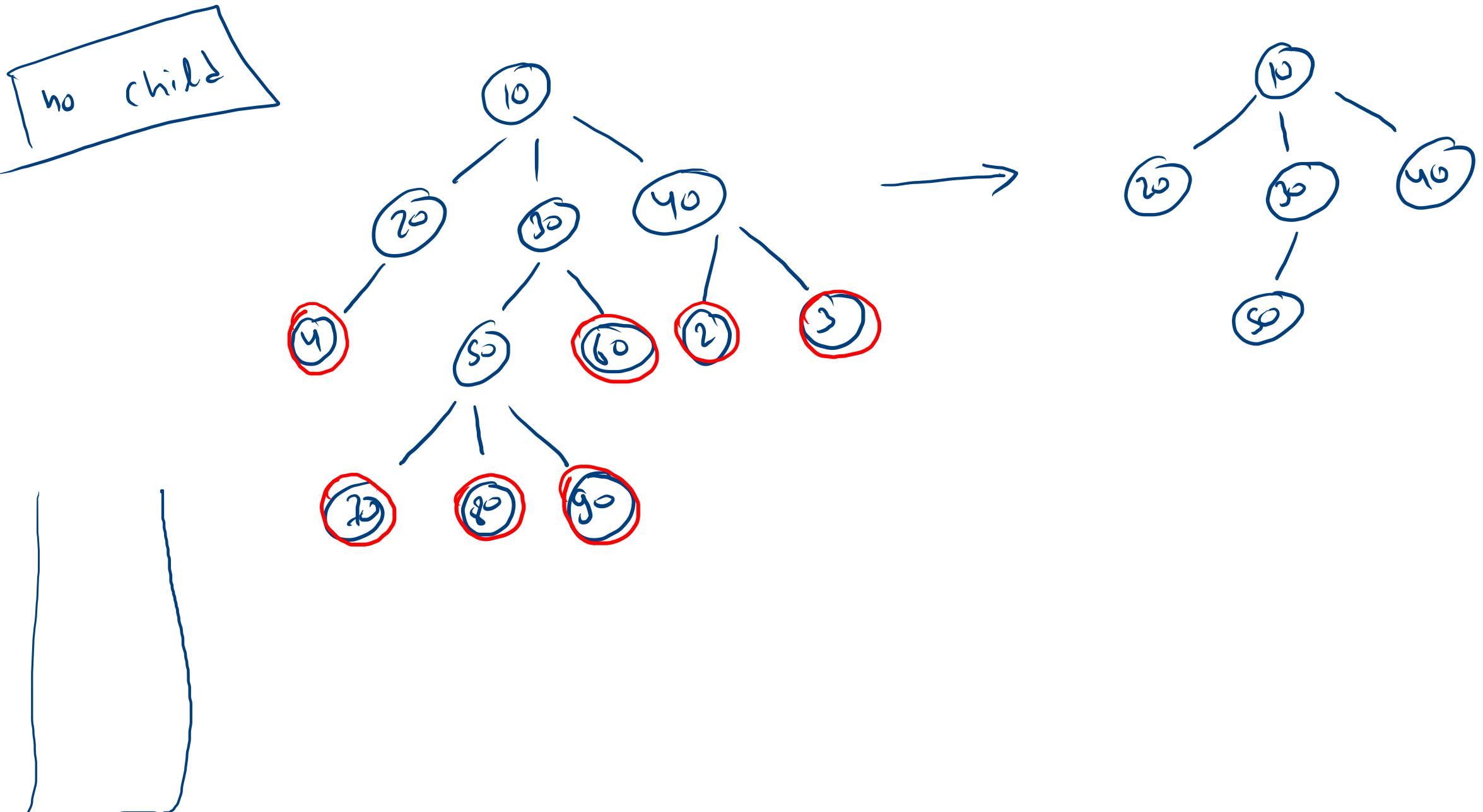
all nodes
children revert

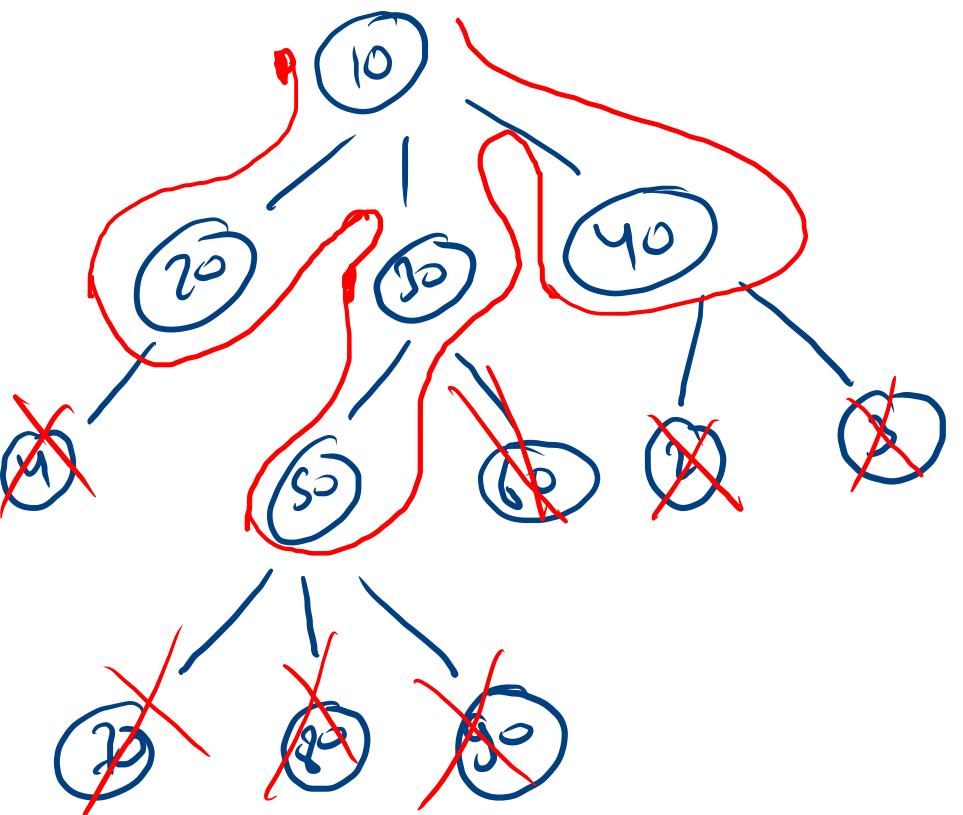


```
public static void mirror(Node node){
```

```
    for(Node child: node.children){  
        mirror(child);  
    }  
}  
Collections.reverse(node.children);  
Collections.sort((~));
```



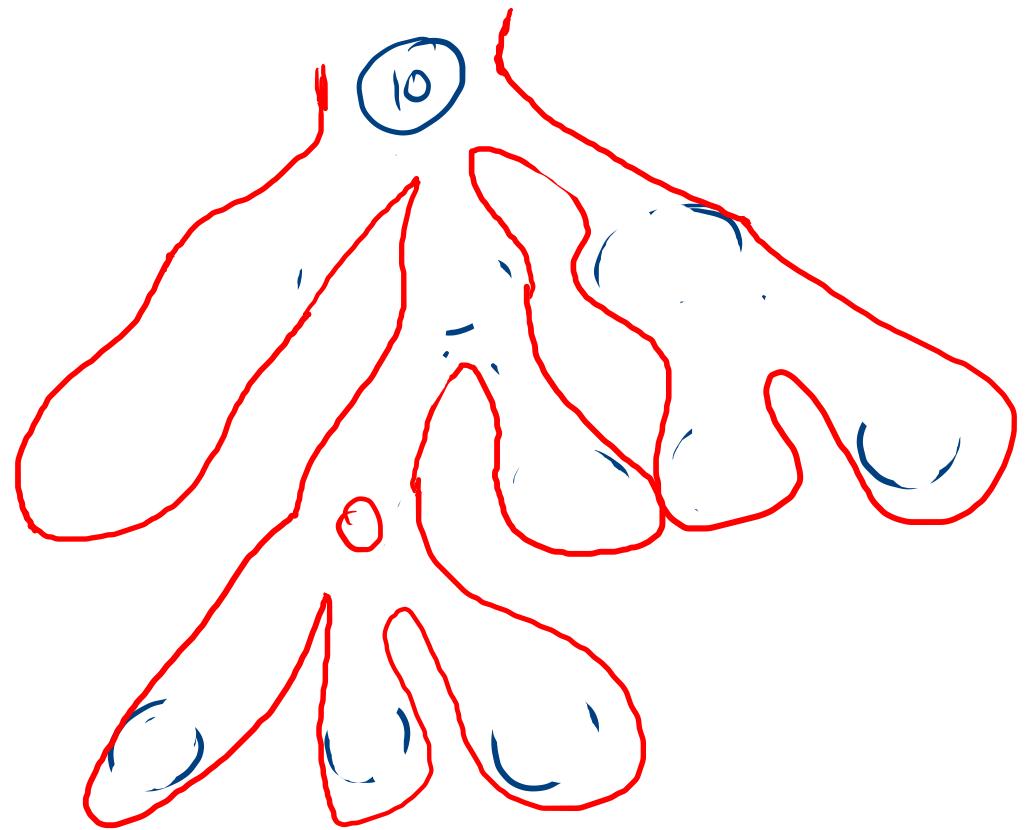


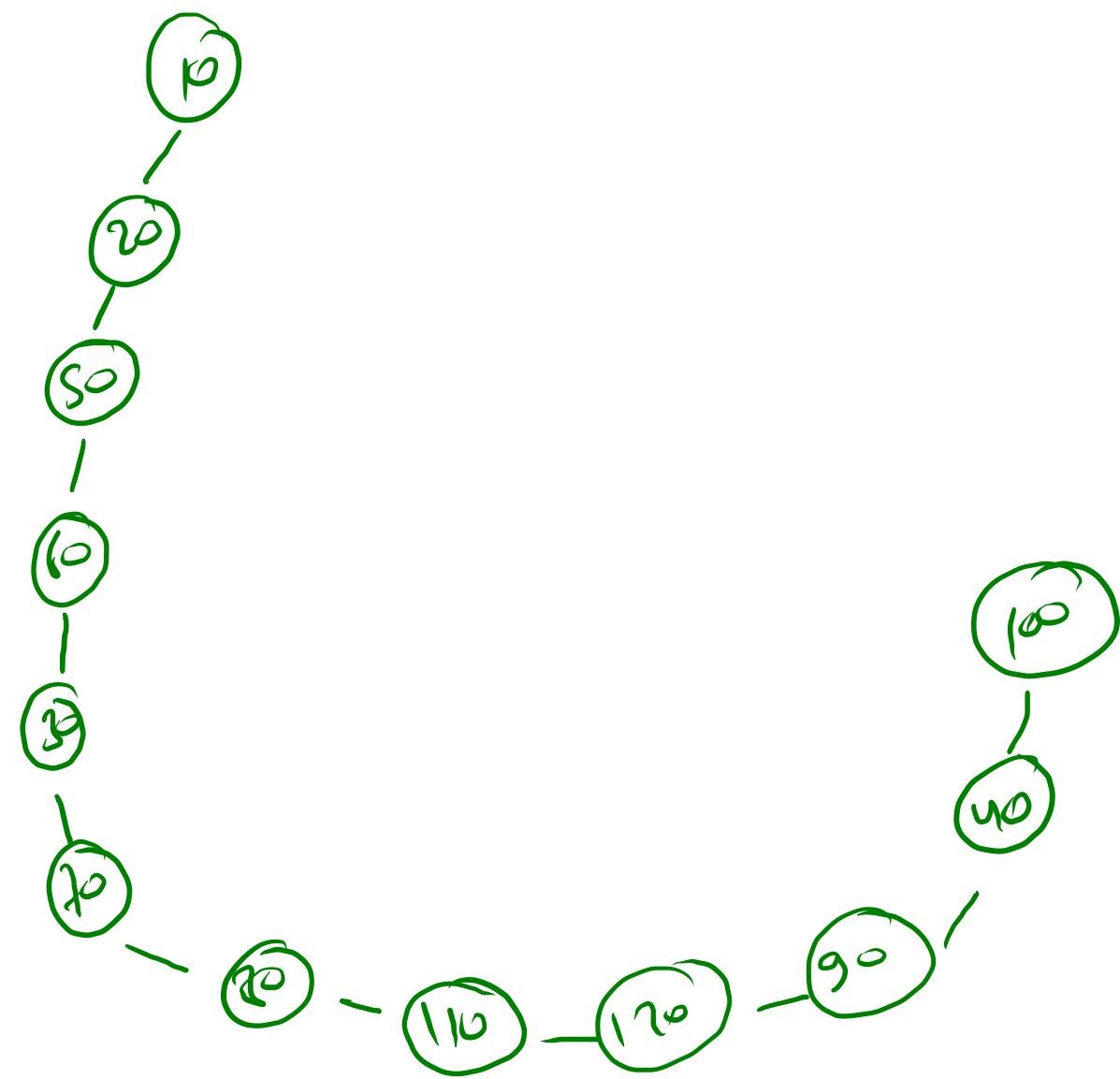
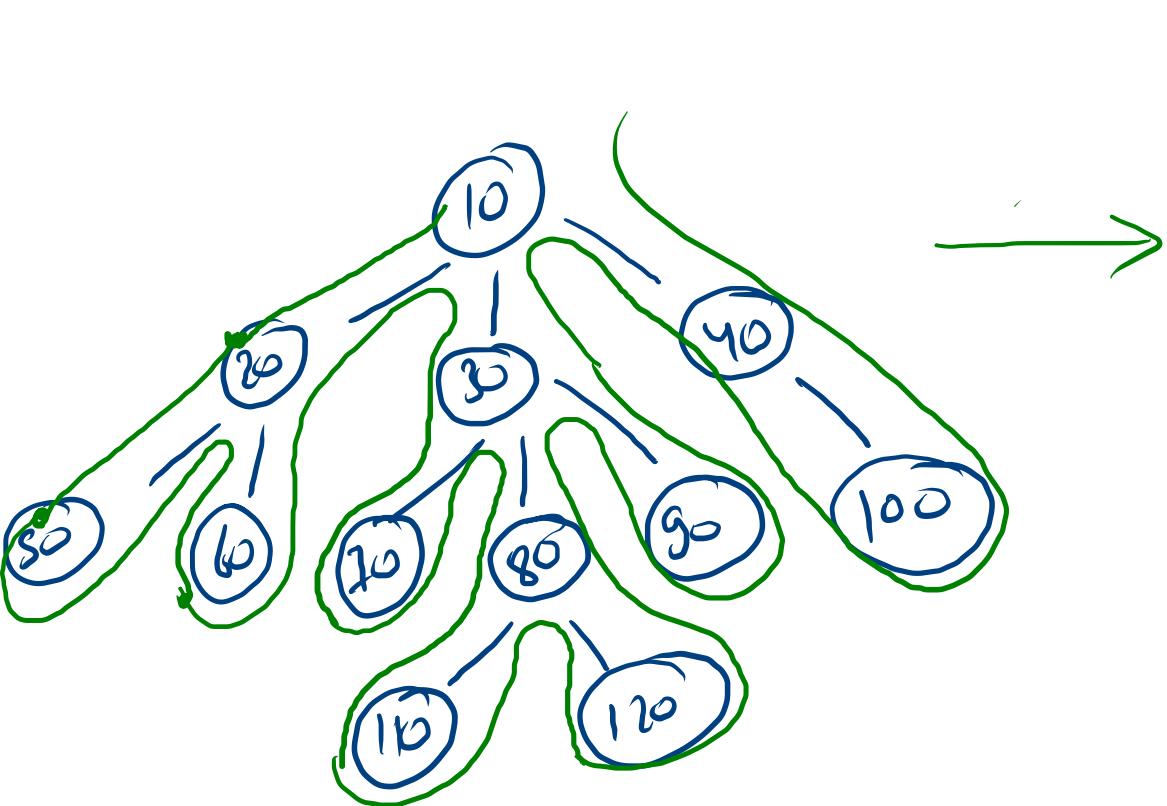


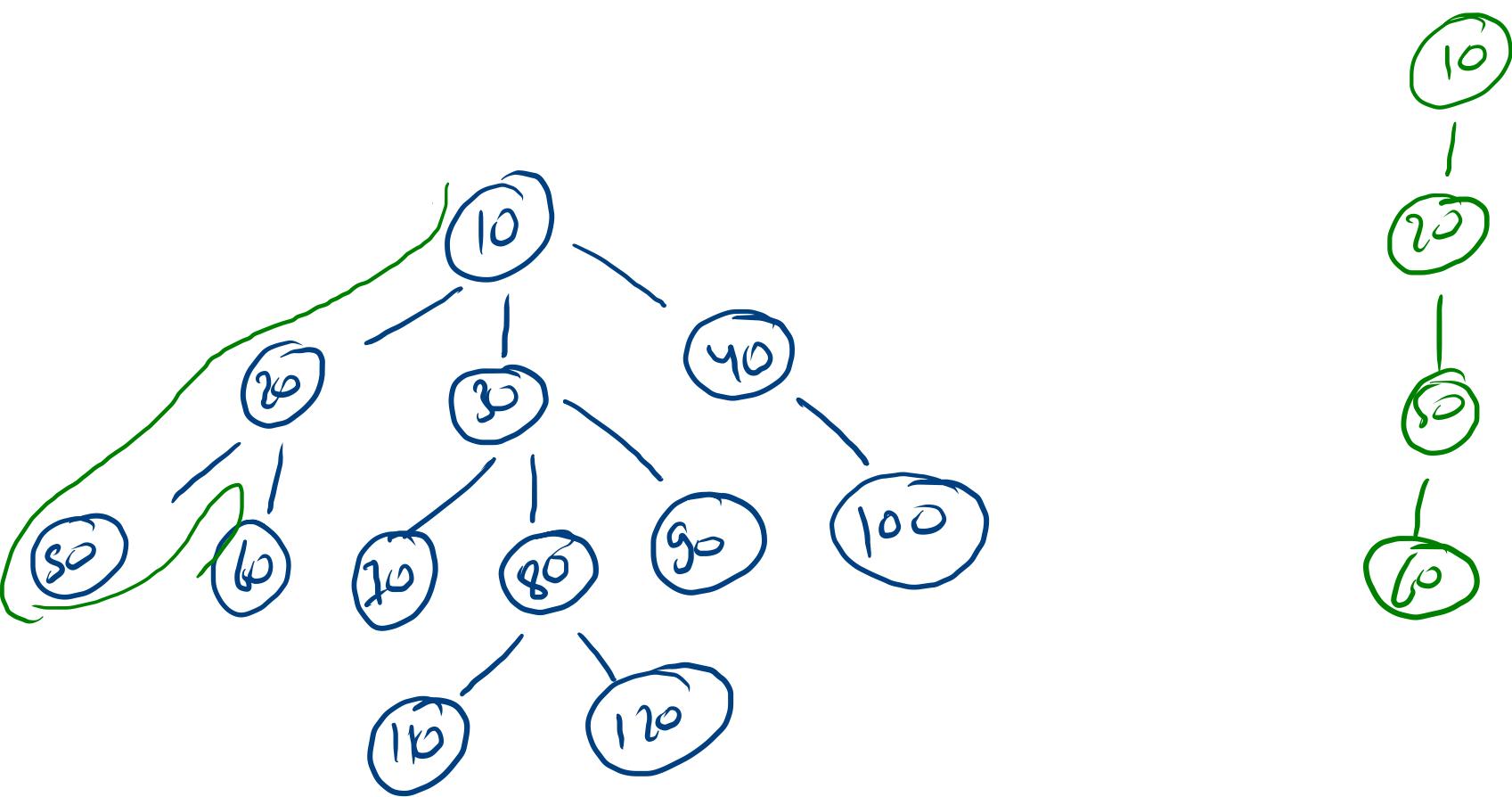
remove all child leaves

cull all children

```
for(Node child: node.children){  
    removeLeaves(child);  
}  
  
for(int i=0;i<node.children.size();i++){  
    Node child = node.children.get(i);  
    if(child.children.size() == 0){  
        node.children.remove(i);  
        i--;  
    }  
}
```

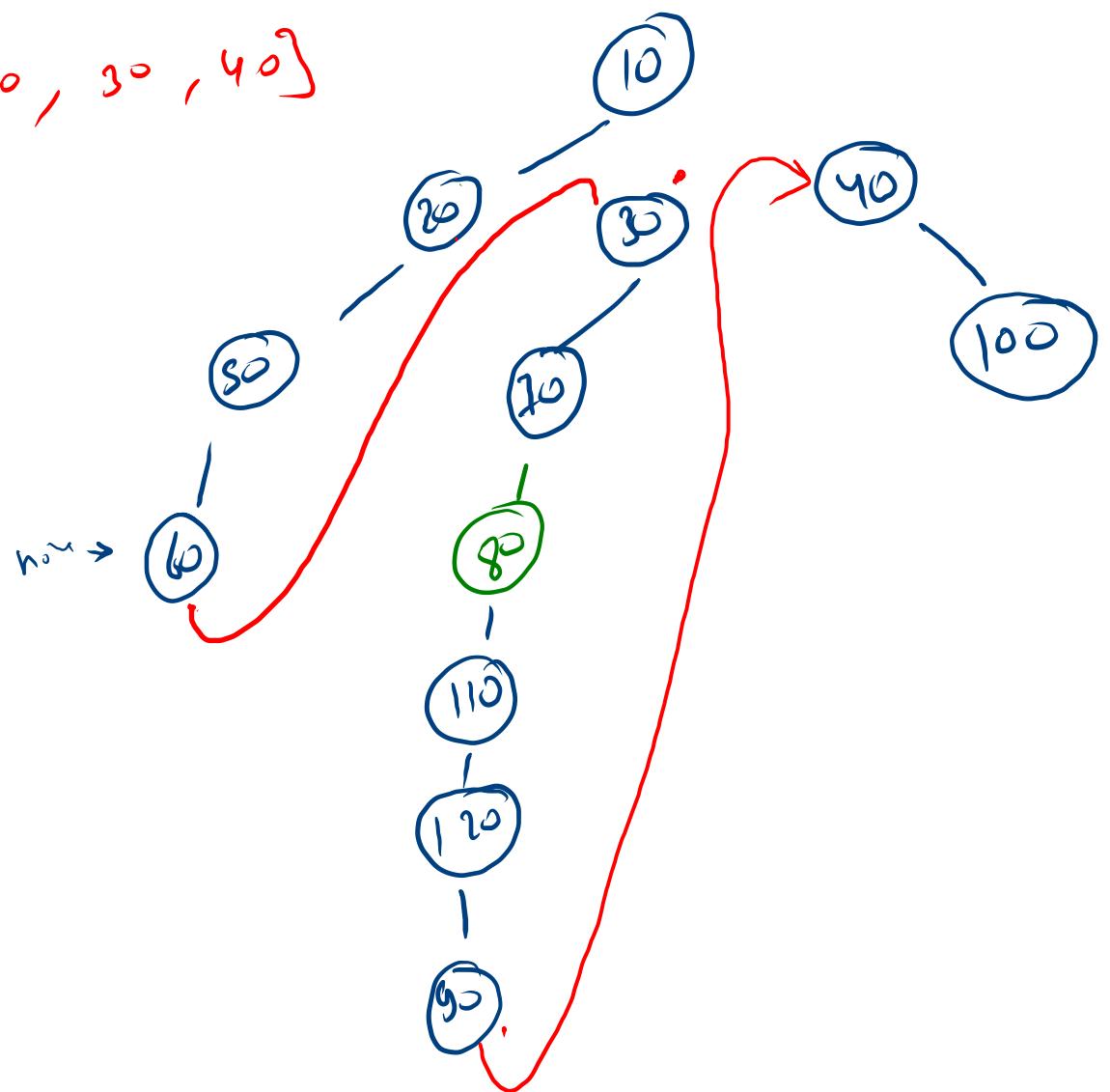






$\{20, 30, 40\}$

$n-1$
 $n-2$



second son = $(n-2)^{\text{th}}$

20

last = $(n-1)^{\text{th}}$

30

last.children.last(last);

getLeave

```

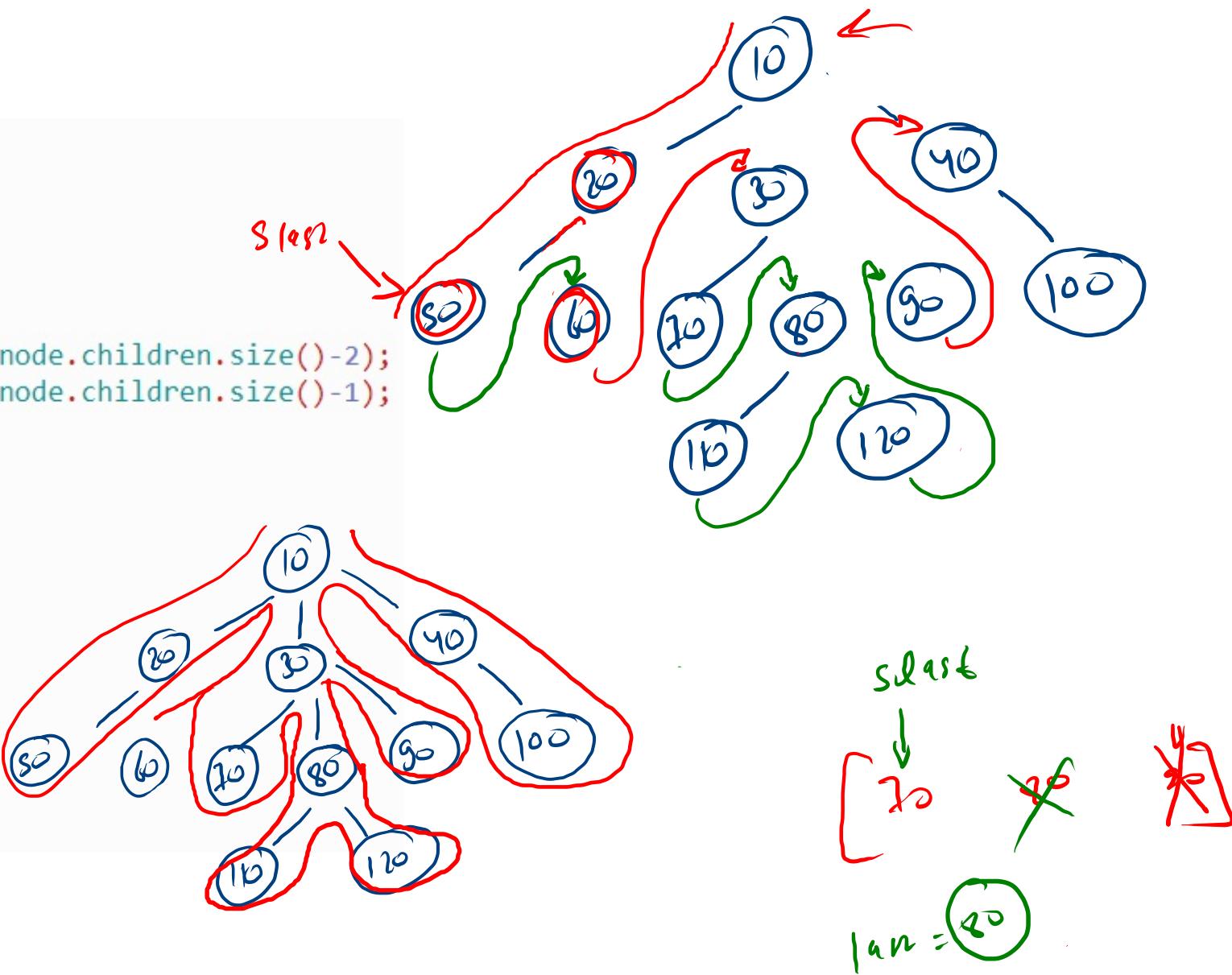
public static void linearize(Node node){
    for(Node child: node.children){
        linearize(child);
    }

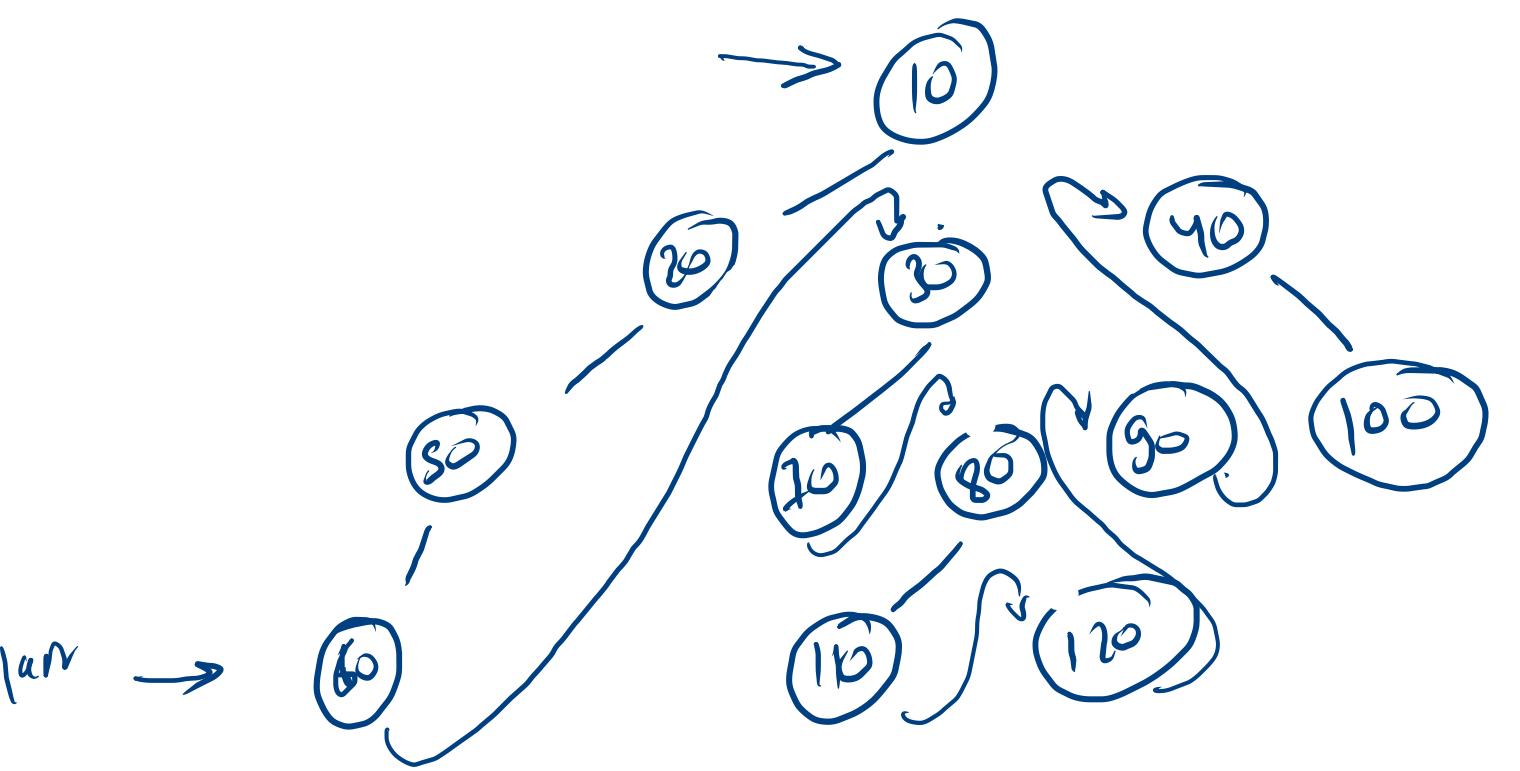
    while(node.children.size() > 1){
        Node secLast = node.children.get(node.children.size()-2);
        Node last = node.children.remove(node.children.size()-1);

        Node leave = getLast(secLast);
        leave.children.add(last);
    }
}

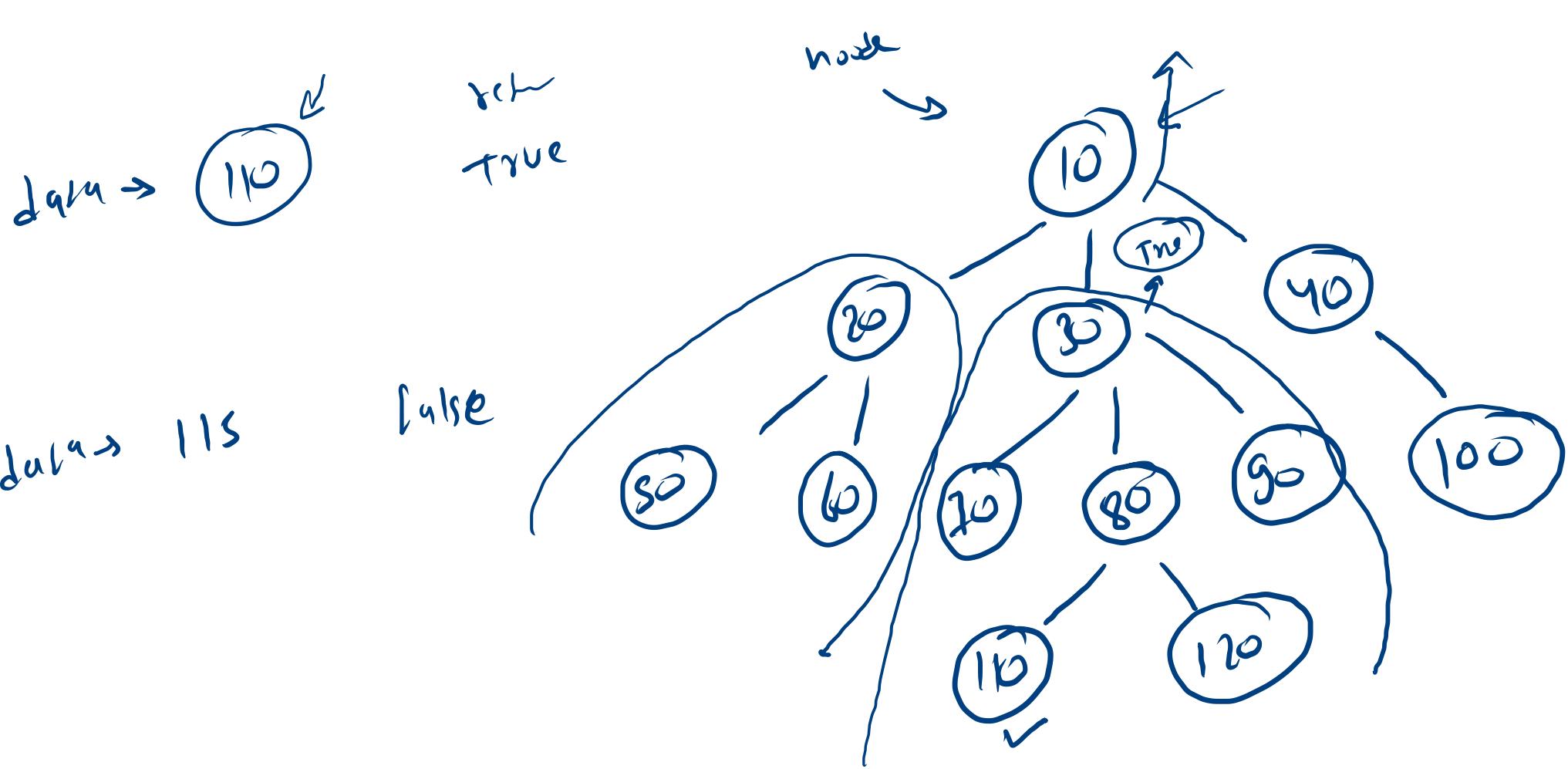
public static Node getLast(Node node){
    while(node.children.size() != 0){
        node = node.children.get(0);
    }
    return node;
}

```

 $\Theta(n)$
 $\Theta(n)$




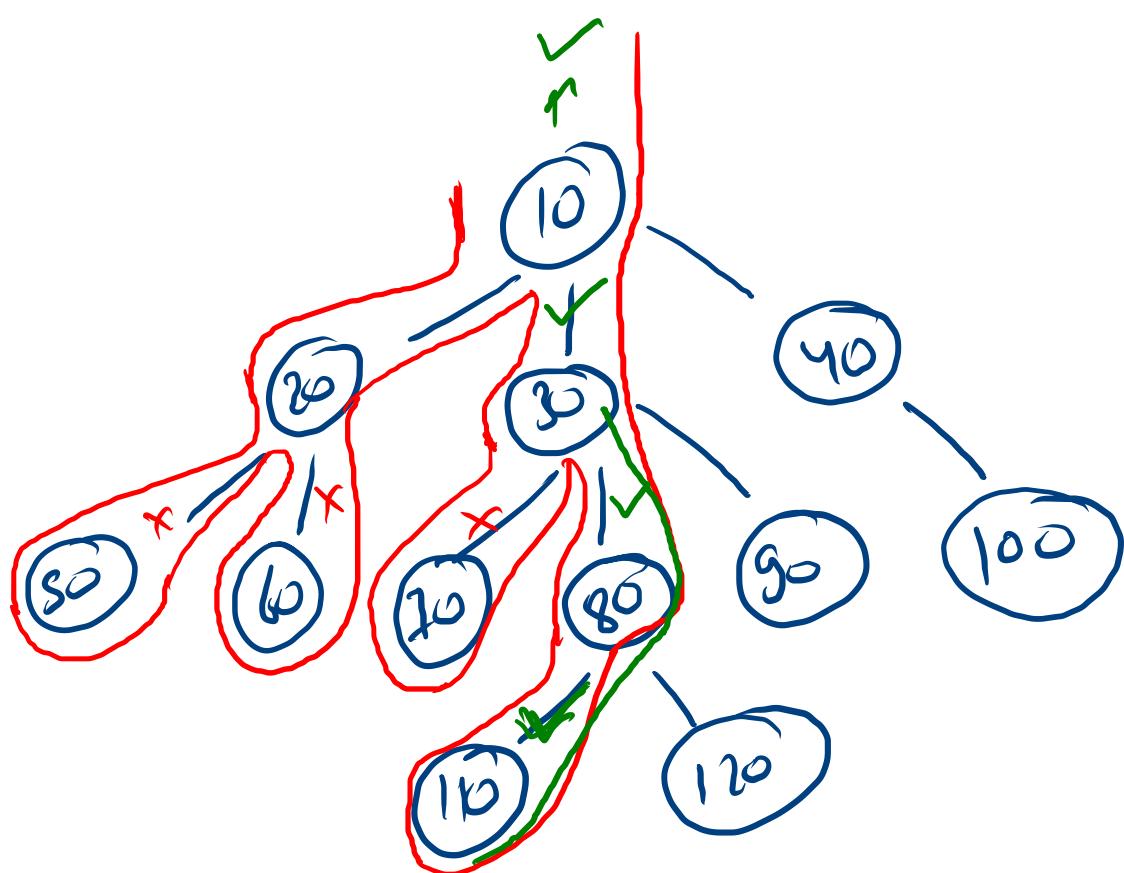
license return last

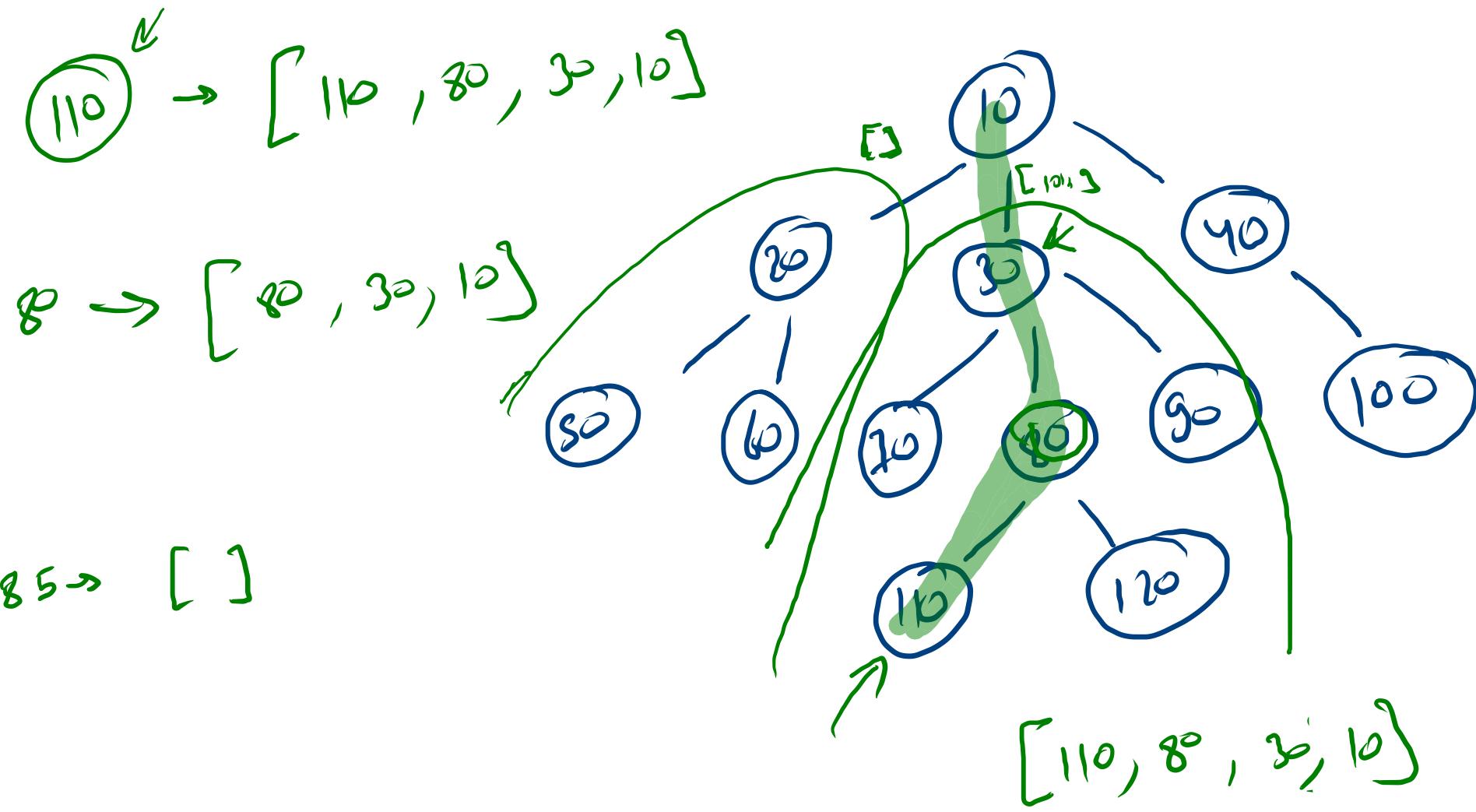


$$\Delta t^u = 110$$

```
public static boolean find(Node node, int data) {  
    if(data == node.data){  
        return true;  
    }  
  
    for(Node child: node.children){  
        boolean result = find(child, data);  
        if(result == true){  
            return true;  
        }  
    }  
    return false;  
}
```

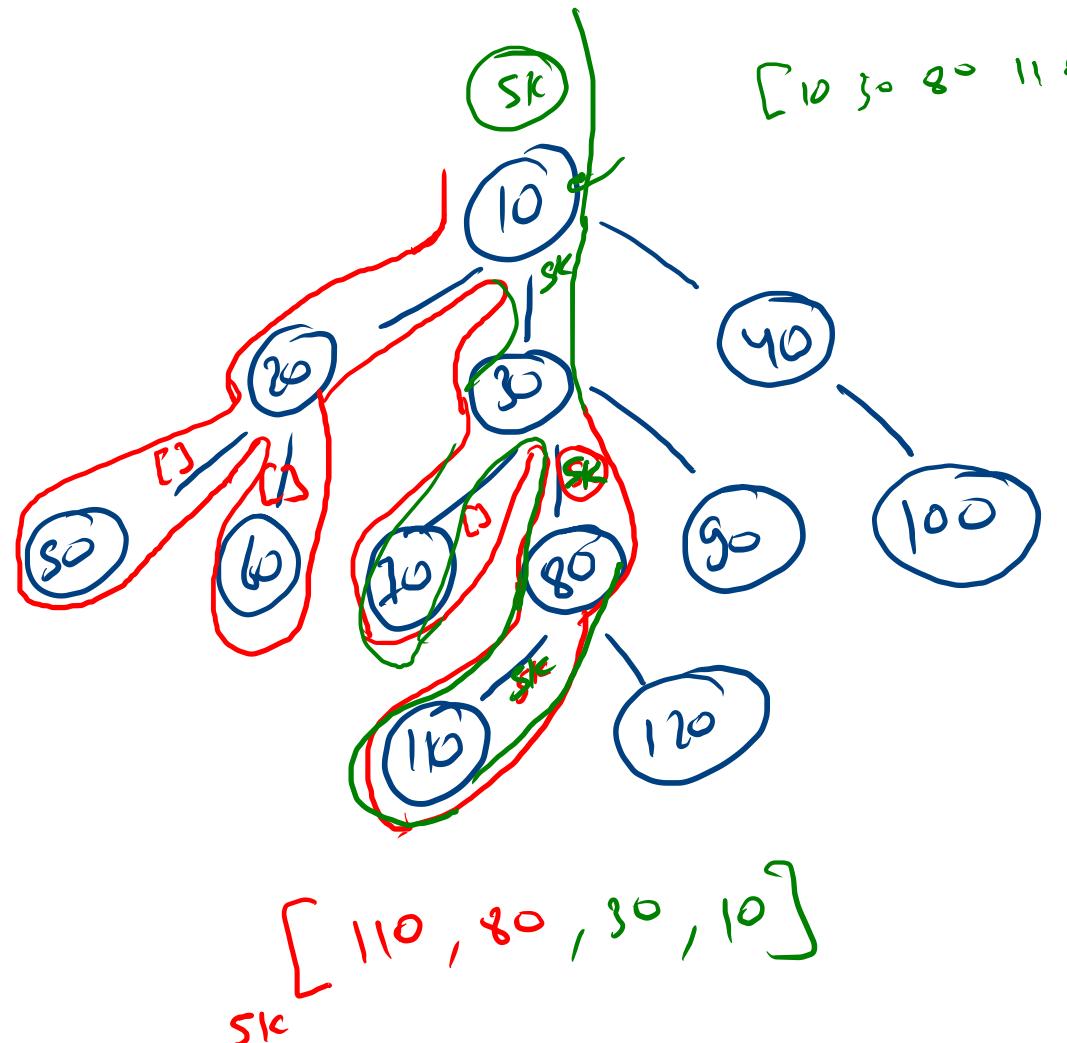
83

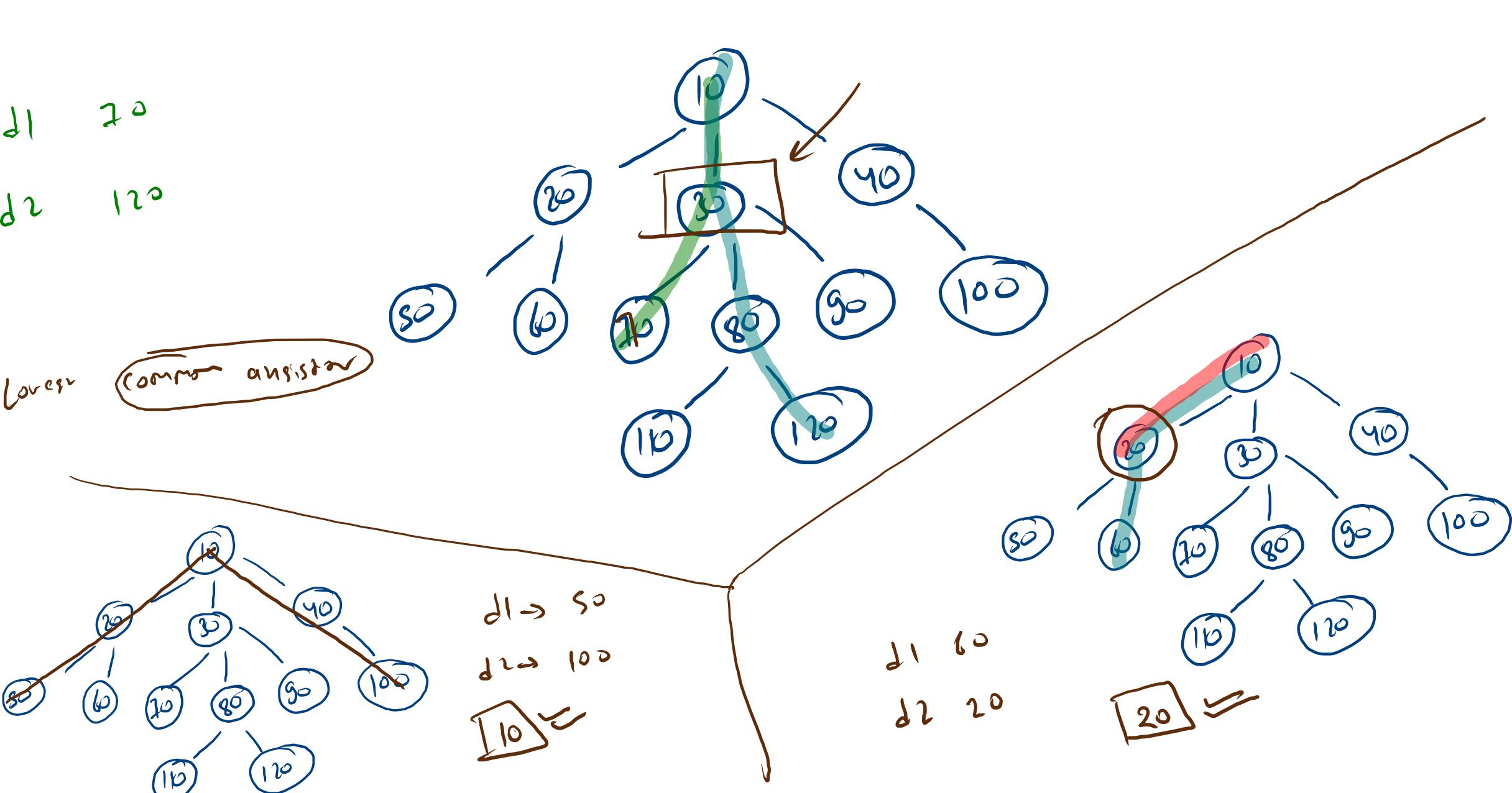


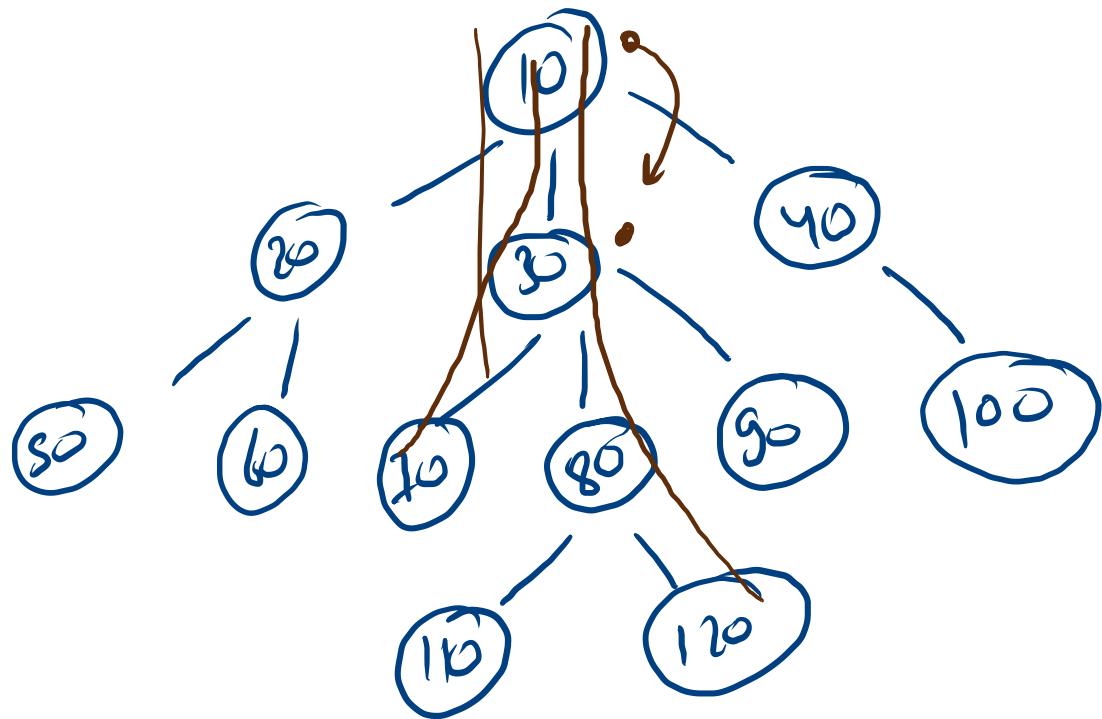


110 $\theta(n)$

```
public static ArrayList<Integer> nodeToRootPath(Node node, int data){  
    if(data == node.data){   sk  
        ArrayList<Integer> list = new ArrayList<Integer>();  
        list.add(data);  
        return list;  
    }  
  
    for(Node child: node.children){  
        ArrayList<Integer> list = nodeToRootPath(child, data);  
        if(list.size() > 0 ){  
            list.add(node.data);  
            return list;  
        }  
    }  
  
    ArrayList<Integer> list = new ArrayList<Integer>();  
    return list;  
}
```



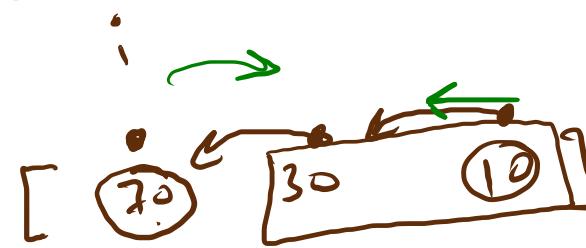




hole \neq root

$d_1 \quad 20$

$d_2 \quad 120$



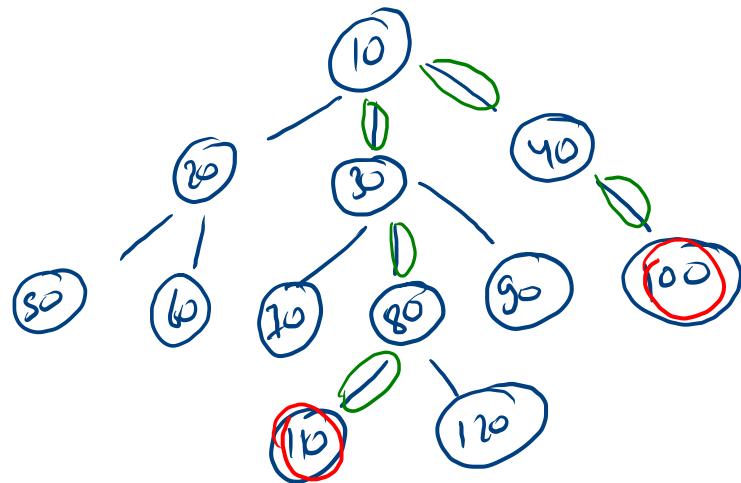
let more same

$i+1$

~~N.V.~~

Distance Between Two Nodes In A Generic Tree

$$d_1 = 10^0$$
$$d_2 = 11^0$$
$$5$$



$$d_1 \rightarrow 2^0$$
$$d_2 \rightarrow 12^0$$
$$\textcircled{3} \Leftarrow$$

