

state = ~~0~~ 2    data = -120

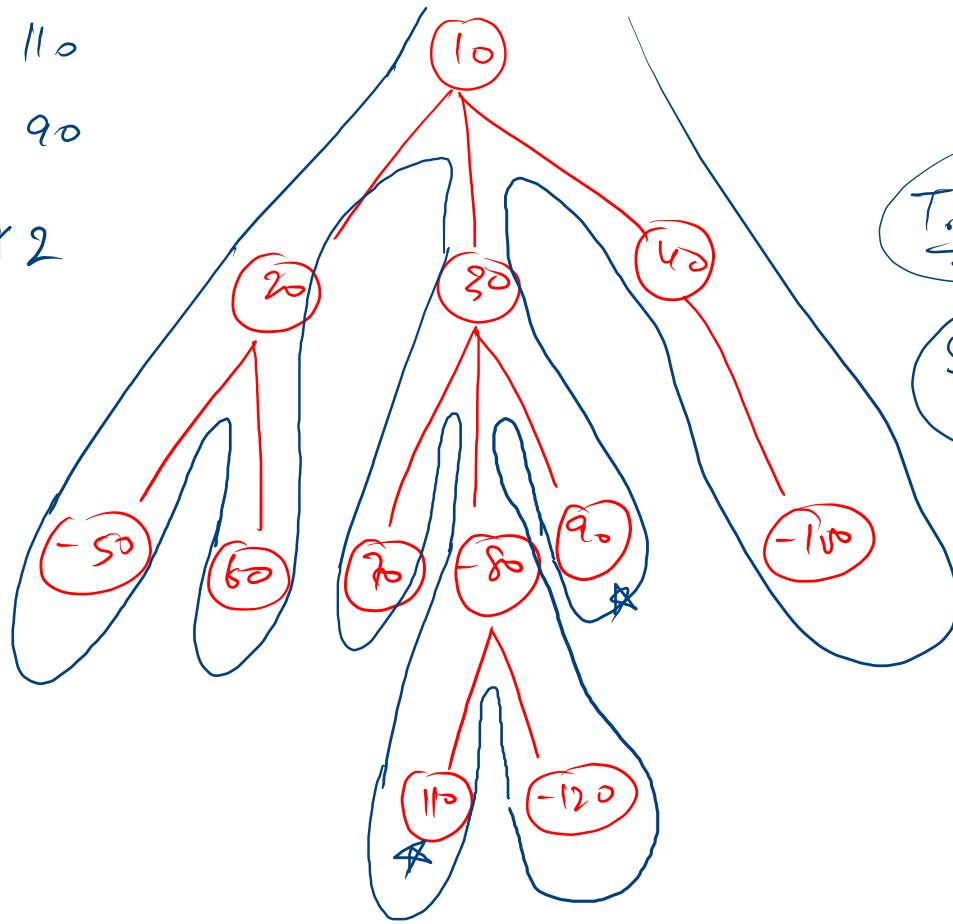
pre = ~~node~~ 10 20 30 40 50 60 70 80 90 100 110

succ = ~~node~~ 90

predecessor = 110

successor = 90

state = 0/2



data = -120

T.C.  $\Rightarrow O(n)$

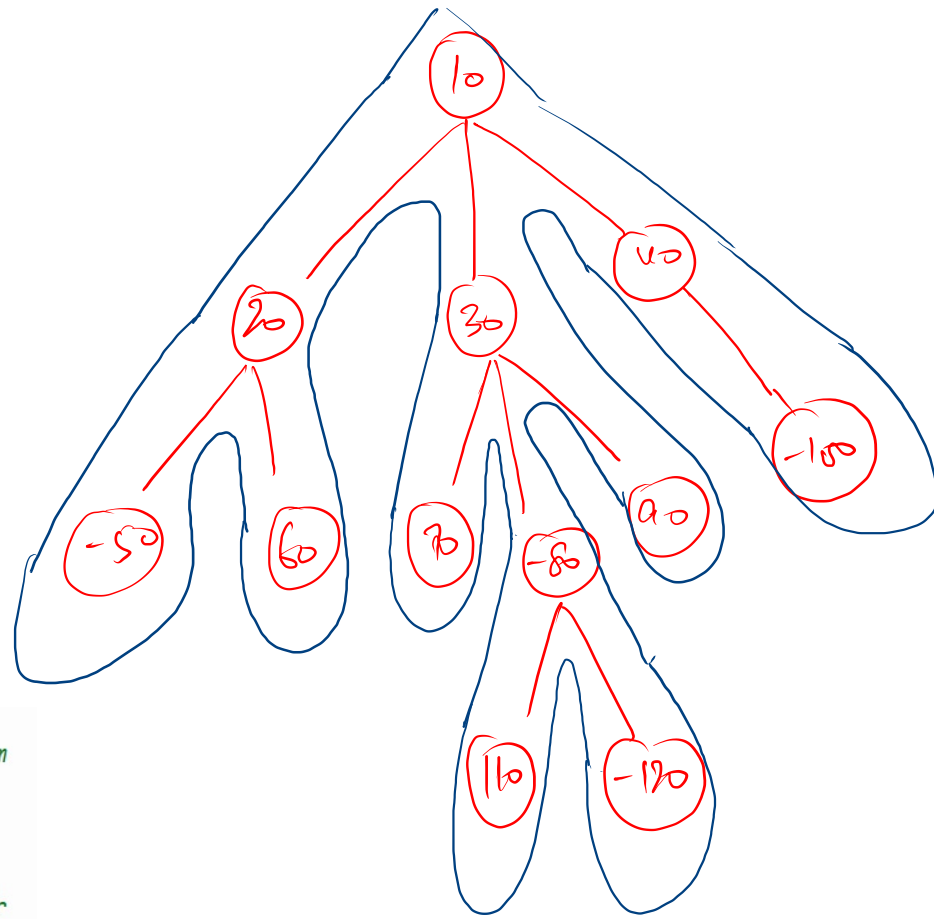
S.C.  $\Rightarrow O(h)$

```
if(state == 0){
    if(data == node.data){
        state = 1;
    }else{
        predecessor = node;
    }
}else if(state == 1){
    successor = node;
    state = 2;
}
```

Criteria  $\rightarrow 20$

$\text{ceil} = \cancel{+\infty} \cancel{110} \cancel{90}$

$\text{floor} = \cancel{-\infty} \cancel{10} \cancel{20} \cancel{60}$



ceil  $\rightarrow$  just larger  
minimum among larger

floor  $\rightarrow$  just smaller  
maximum among smaller

```
if(node.data > criteria){ // Larger
    if(node.data < ceil){ // minimum
        ceil = node.data;
    }
}

if(node.data < criteria){ // Smaller
    if(node.data > floor){ // maximum
        floor = node.data;
    }
}
```

8

Criteria

floor

70

~~110~~ 110

110

~~90~~ 90

90

~~70~~ 70

70

~~60~~ 60

60

40

40

~~30~~ 30

30

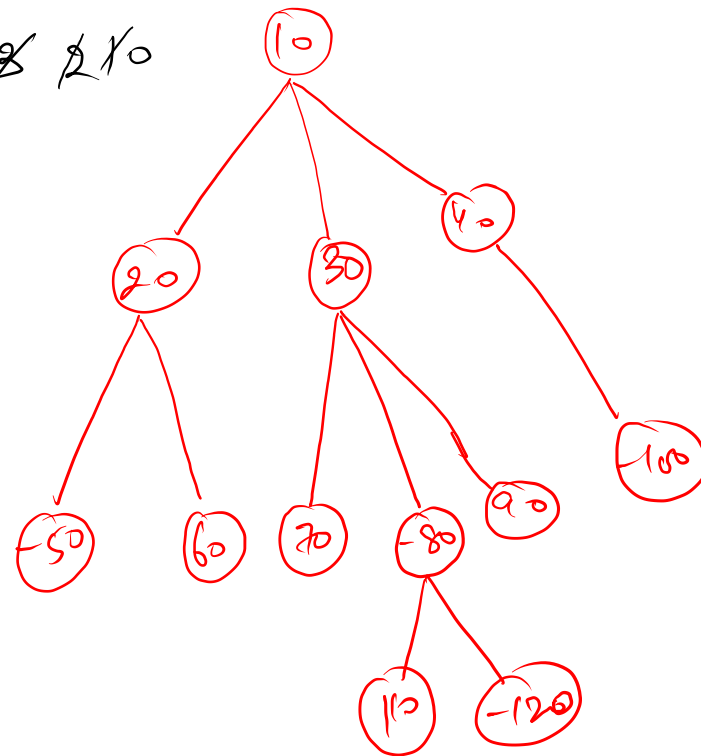
20

20

10

10

$k = 8$  ~~7~~ ~~6~~ ~~5~~ ~~4~~ ~~3~~ ~~2~~ ~~1~~ 0



```

public static int kthLargest(Node root, int k){
    int criteria = Integer.MAX_VALUE;

    while(k != 0){
        floor = Integer.MIN_VALUE;
        ceilAndFloor(root, criteria);
        criteria = floor;
        k--;
    }

    return criteria;
}
    
```

20

10 20 -50 -1 60 -1 -1 30 -70 -1 80 -1 90 -1 -1 40 -100 -1 -1 -1

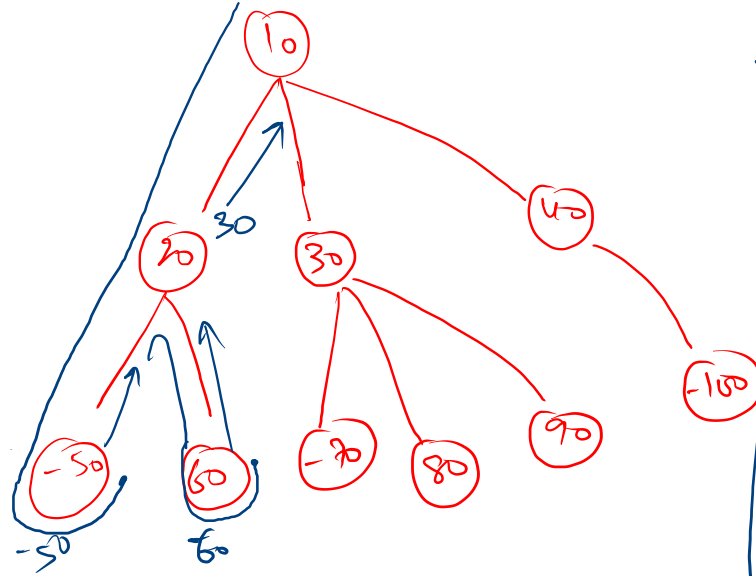
Subtree

part of original Tree

Maximum Subtree Sum

✓ mSum = ~~-100~~ 60

✓ mSumNode = ~~node~~ (60)



static int mSum;  
static Node mSumNode;

```
public static int sum(Node node){  
    int res = 0;  
  
    for(Node child : node.children){  
        res += sum(child);  
    }  
  
    res += node.data;  
  
    if(res > mSum){  
        mSum = res;  
        mSumNode = node;  
    }  
  
    return res;  
}
```

```
mSum = Integer.MIN_VALUE;  
sum(root);  
System.out.println(mSumNode.data+"@"+mSum);
```

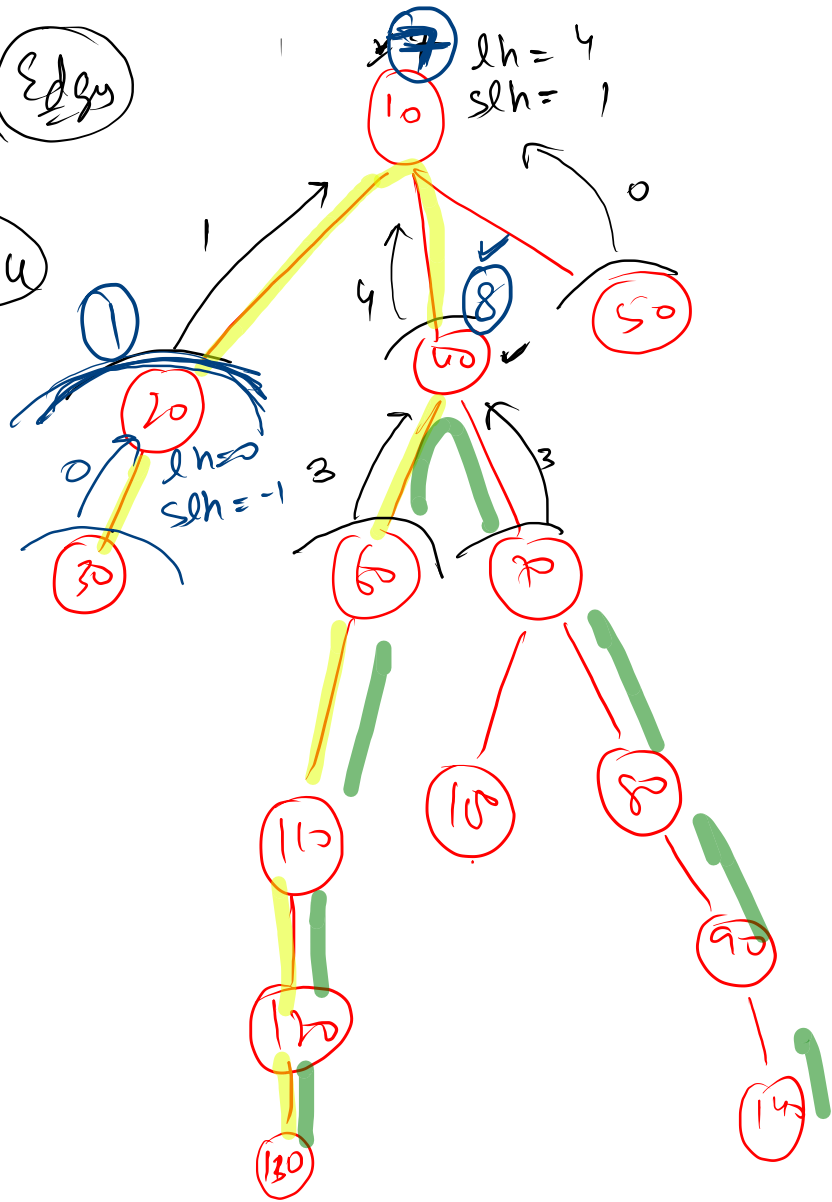
Diameter

Edges

Maximum Distance  
b/w two nodes

H.W.

Concept + Solution



Height

Thurs

$$lh + rh + 2$$

height

Diameter (Node) Max

Ans