

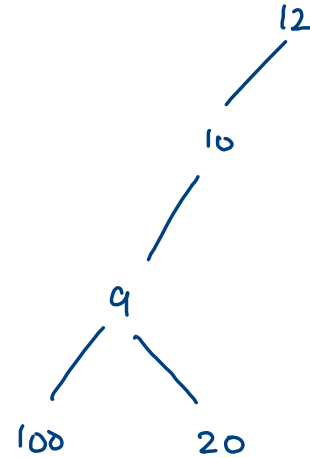
### 337. House Robber III

Medium  6057  93  Add to List  Share

The thief has found himself a new place for his thievery again. There is only one entrance to this area, called `root`.

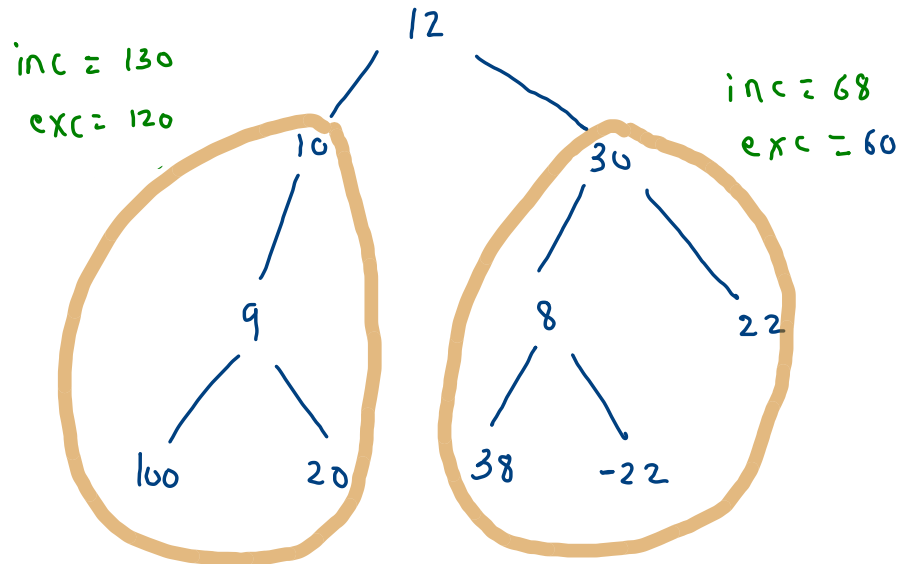
Besides the `root`, each house has one and only one parent house. After a tour, the smart thief realized that all houses in this place form a binary tree. It will automatically contact the police if **two directly-linked houses were broken into on the same night**.

Given the `root` of the binary tree, return *the maximum amount of money the thief can rob **without alerting the police***.



$$\text{ans} = 132 \\ (12 + 100 + 20)$$

eg. (why not alternate level-sum)



pair {  
inc;  
exc;  
}

lp = solve (node.left);  
rp = solve (node.right);

inc = lp.exc + rp.exc + node.data;  
exc = max (lp.inc, lp.exc) +  
max (rp.inc, rp.exc)

inc: 192  
(12, 100, 20, 22, 38)

exc: 198  
(10, 100, 20, 30, 38)

```

public Pair helper(TreeNode node) {
    if(node == null) {
        return new Pair(0,0);
    }

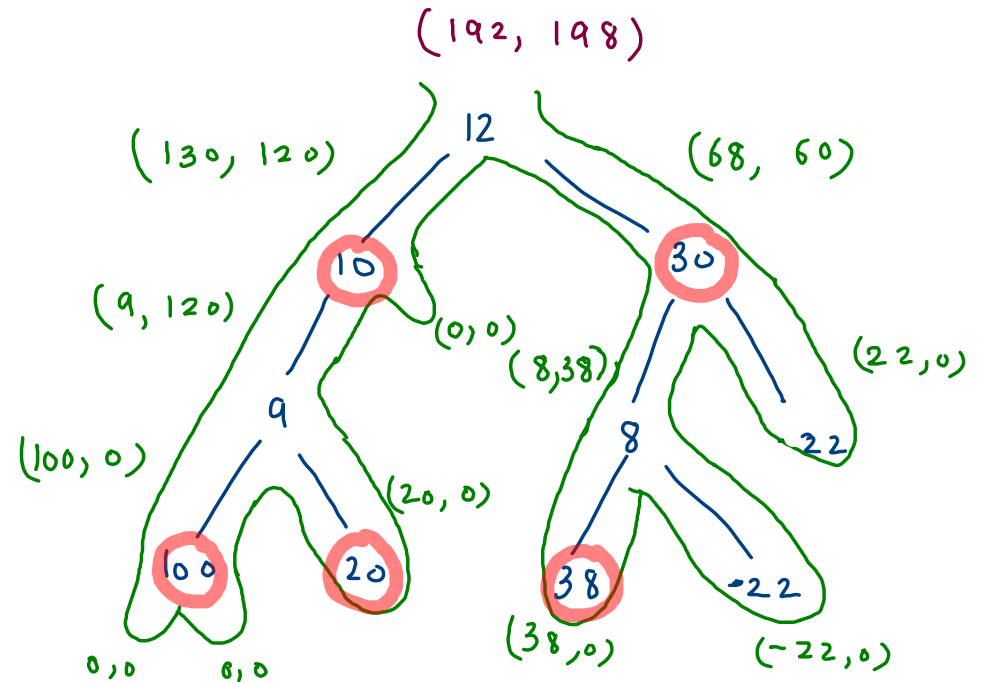
    Pair lp = helper(node.left); //left pair
    Pair rp = helper(node.right); //right pair

    int incMP = lp.excMP + rp.excMP + node.val;
    int excMP = Math.max(lp.incMP,lp.excMP) + Math.max(rp.incMP,rp.excMP);

    return new Pair(incMP,excMP);
}

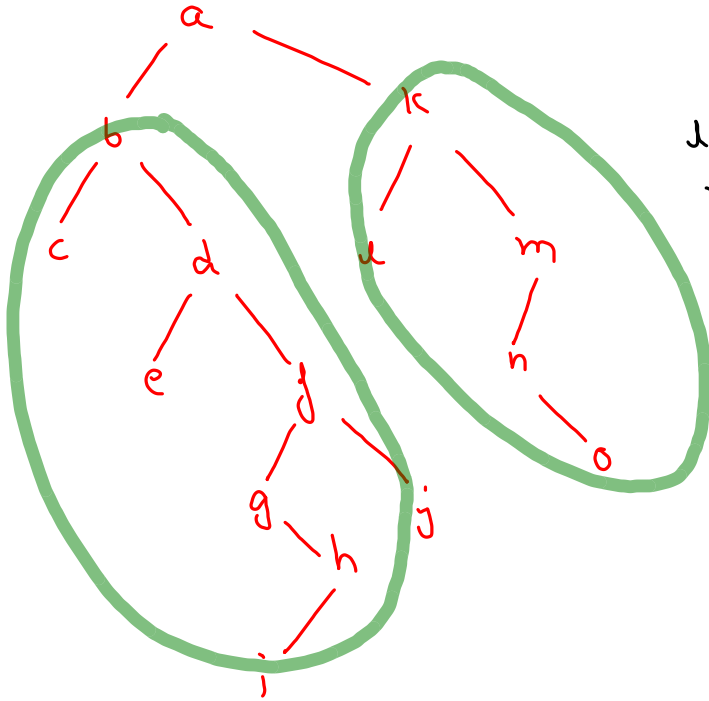
```

$(incMP, excMP)$



## 1372. Longest ZigZag Path in a Binary Tree

$l_{zzp} = 1$   
 $r_{zzp} = 2$   
 $m_{zzp} = 4$



$l_{zzp} = 1$   
 $r_{zzp} = 3$   
 $m_{zzp} = 3$

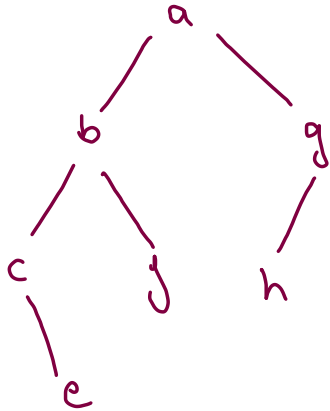
Pair {  
int  $l_{zzp}$ ;  
int  $r_{zzp}$ ;  
int  $m_{zzp}$ ;  
3 } starting from  
node

$lp = \text{solve}(\text{node.left});$   
 $rp = \text{solve}(\text{node.right});$

$l_{zzp} = 1 + lp.r_{zzp};$   
 $r_{zzp} = 1 + rp.l_{zzp};$   
 $m_{zzp} = \max(lp.m_{zzp}, rp.m_{zzp},$   
 $l_{zzp}, r_{zzp});$



## Maximum Path Sum In Between Two Leaves Of Binary Tree



leaf to leaf path

e c b f (e → f)

e c b a g h (e → h)

f b a g h (f → h)

Pair {  
    n2d;  
    d2d;  
}