

## Morning

1. Construct Binary Tree using Inorder and level order
2. Construct BST using Inorder }
3. Construct BST using Pre Order
4. Construct BST using Postorder

skip → Construct BST using level order

## Evening (from 7:00 PM)

1. Cameras in Binary Tree
2. House Robber in B-Tree
3. longest zigzag path
4. validate BST
5. Recover BST



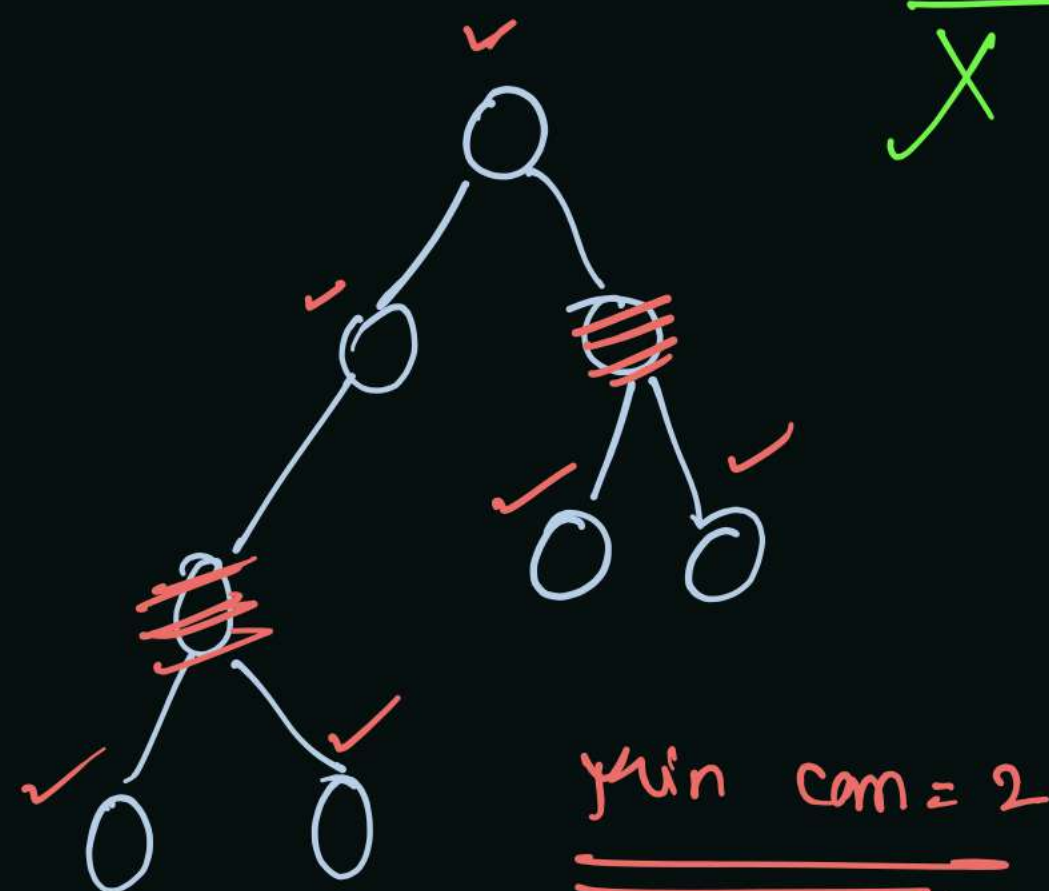
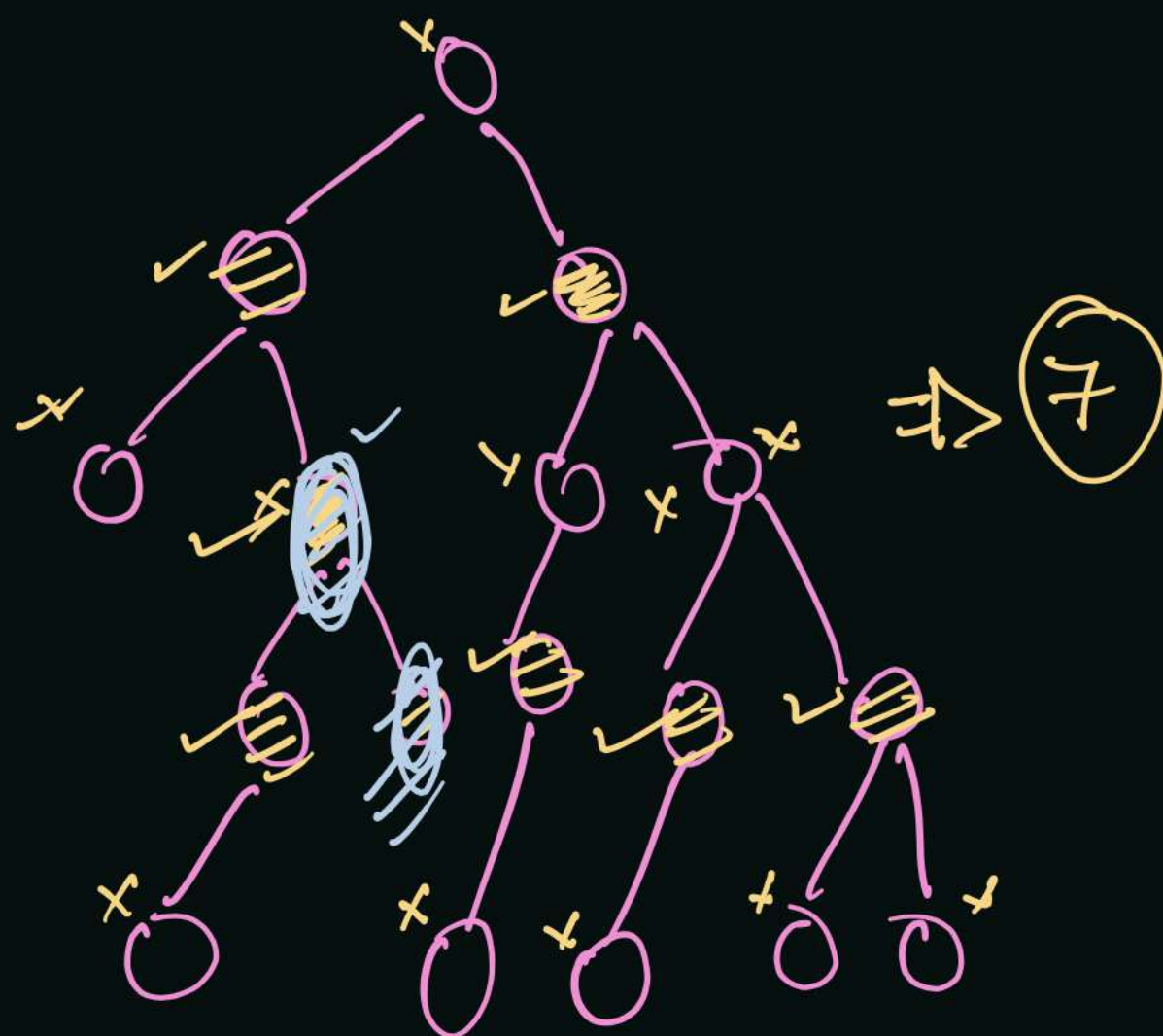
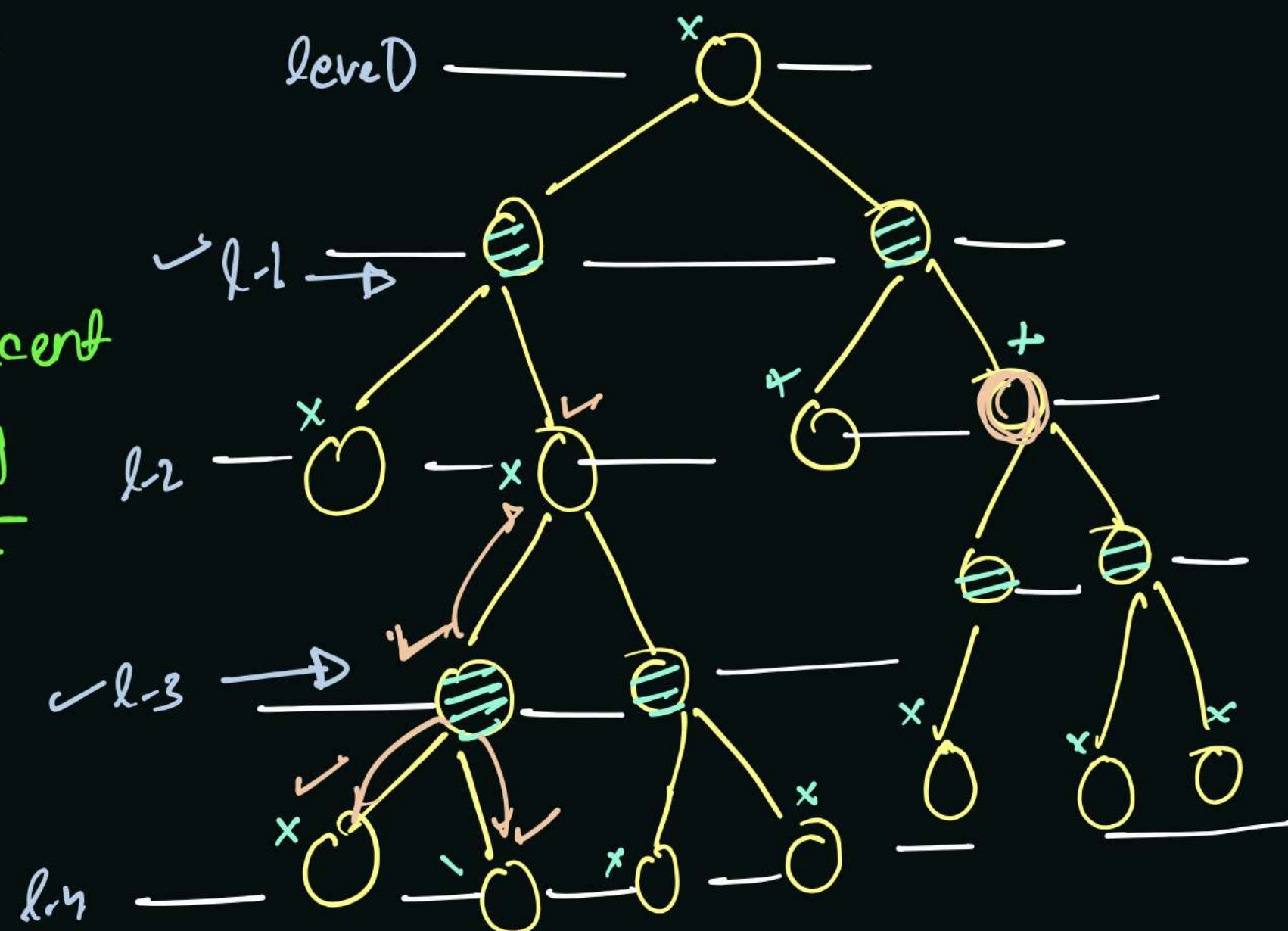
Min number of camera's Required??

Camera  $\rightarrow$  protect $\rightarrow$  Adjacent NodeMin Camera  $\Rightarrow 6$ 

Adjacent

level

X

Adjacent level  $\rightarrow$  FailCamera Present  $\rightarrow$  Safe

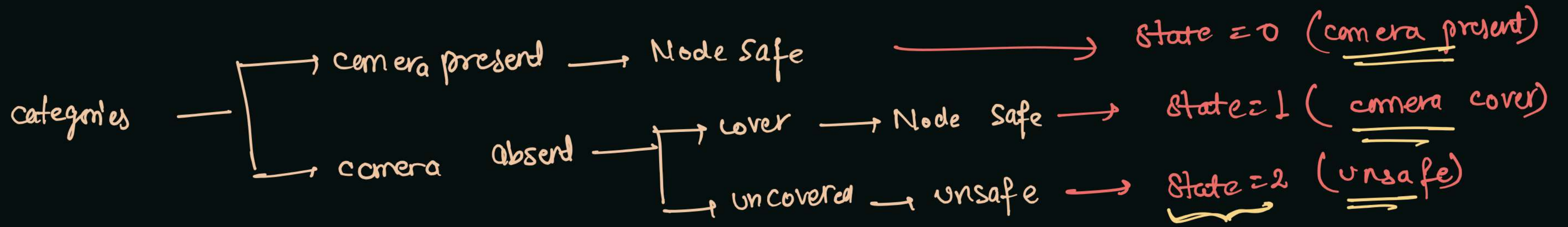
camera absent

Cover  $\rightarrow$  Safe

uncover

 $\rightarrow$  unsafe





Post Order → Result of lower level

if (node == null) return 1;

left → lres → lres

right → rres → rres

lres      rres      node  
 [ 1      1 ] → return 2 || return 2

2	2	→ camera++; return 0
0	2	→ camera++; return 0      return 0 camera++;
0	0	→ return 1
0	1	→ return 1

→ return 1





71

$$\text{profit} = 0$$

C. Directly Gist connected

→ We can skip more than one house -

## Format of answer

 $a, b$ 

# Robbery

# Robbery

Option available at

## Mode

# Robbery

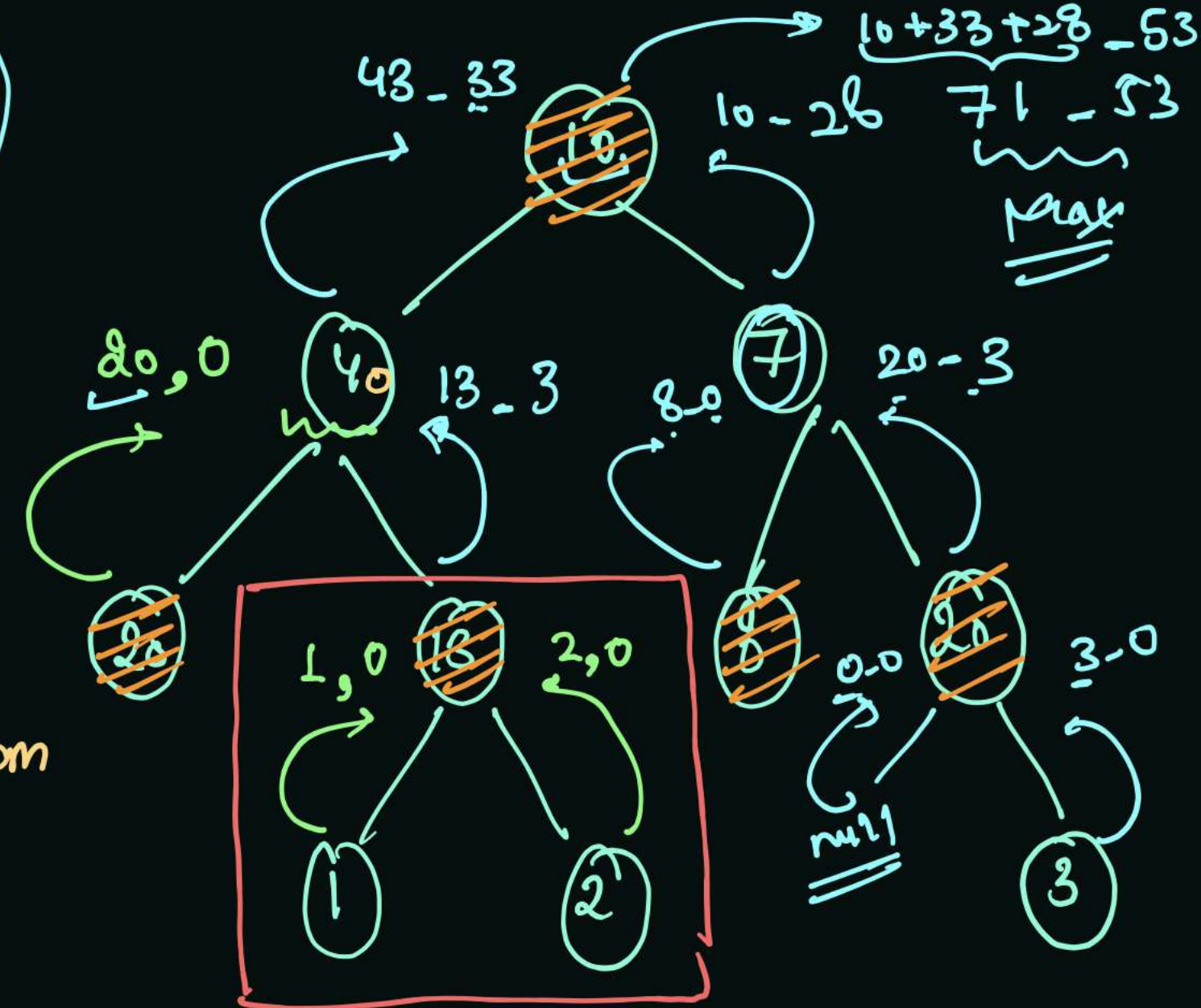
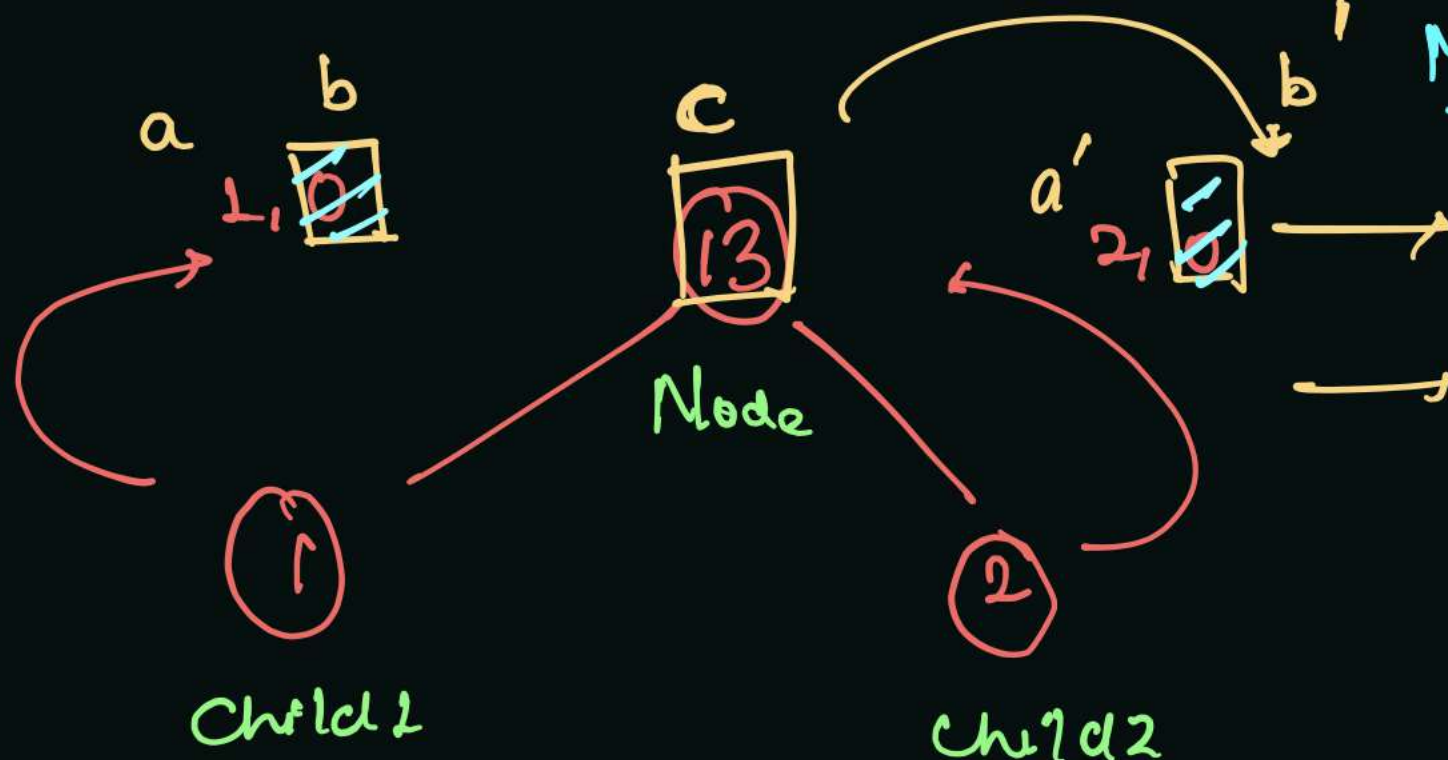
$$c + b + b' = 13 + 0 + 0 = 13$$

## Robbery x

$$a + a' = 3$$

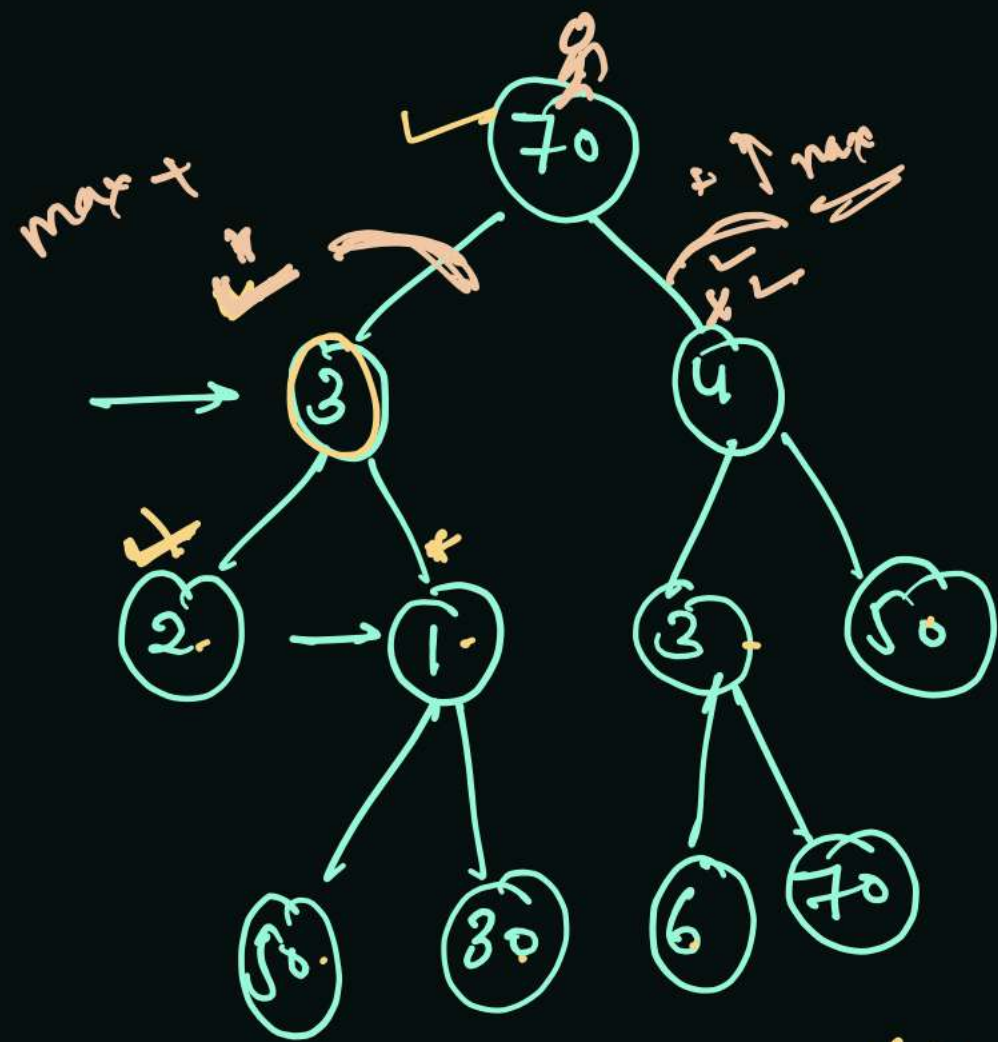
Max {  $a+b$ ,  $a+b$ ,  $b+b$  }

# Robbery

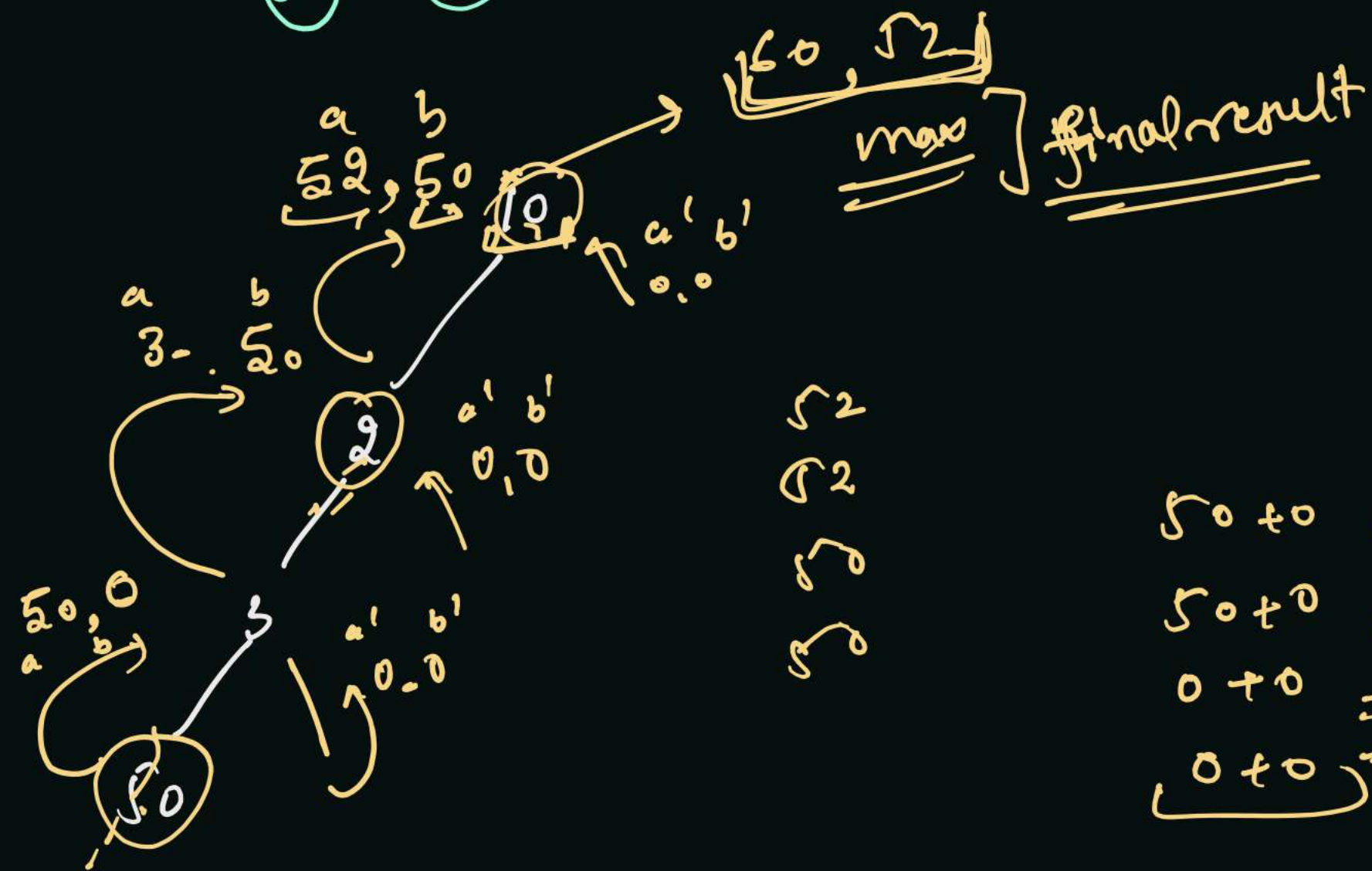
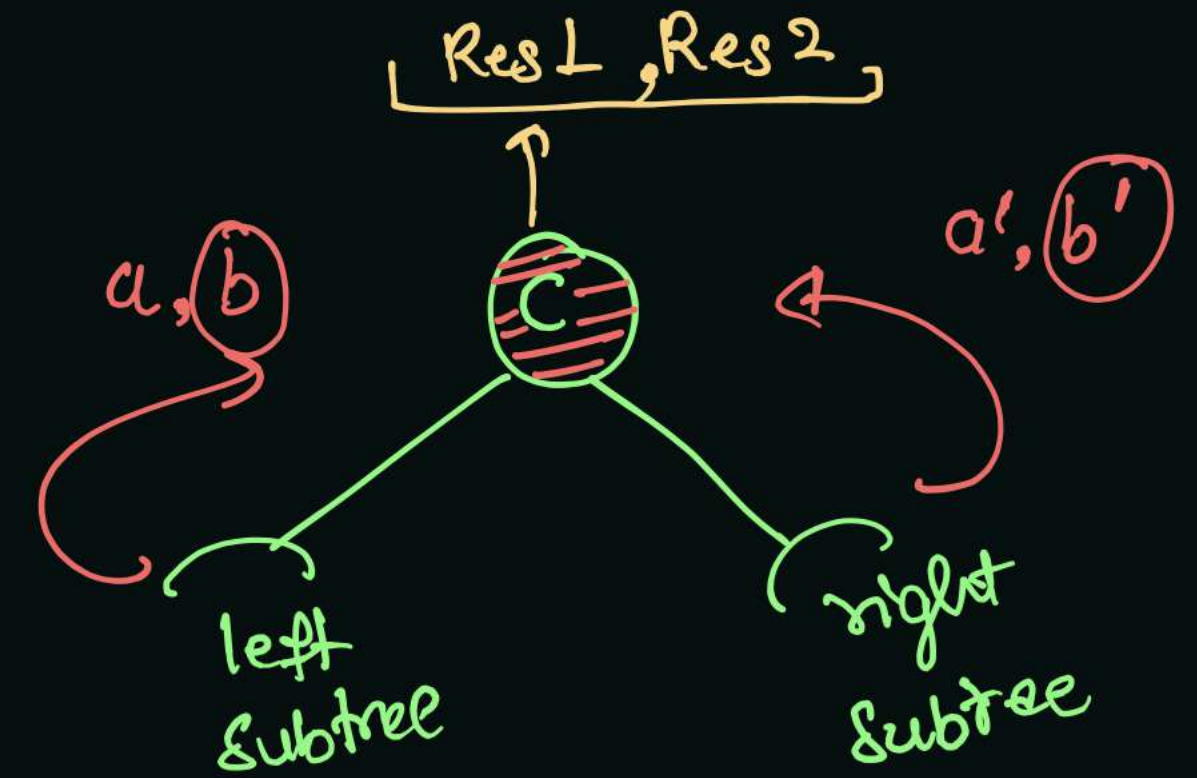


- a  $\rightarrow$  profit with Robbery in ~~node~~ child1
- a'  $\rightarrow$  Profit with Robber in child2
- b  $\rightarrow$  Profit without Robbery in child1
- b'  $\rightarrow$  profit without Robber in child2
- c  $\rightarrow$  Can make profit with Robbery  
in current Node  $\rightarrow$





$a \rightarrow$  Robbed in child 1  
 $a' \rightarrow$  " " child 2  
 $b \rightarrow$  No Rob in child 1  
 $b' \rightarrow$  " " child 2  
 $c \rightarrow$  profit at node current

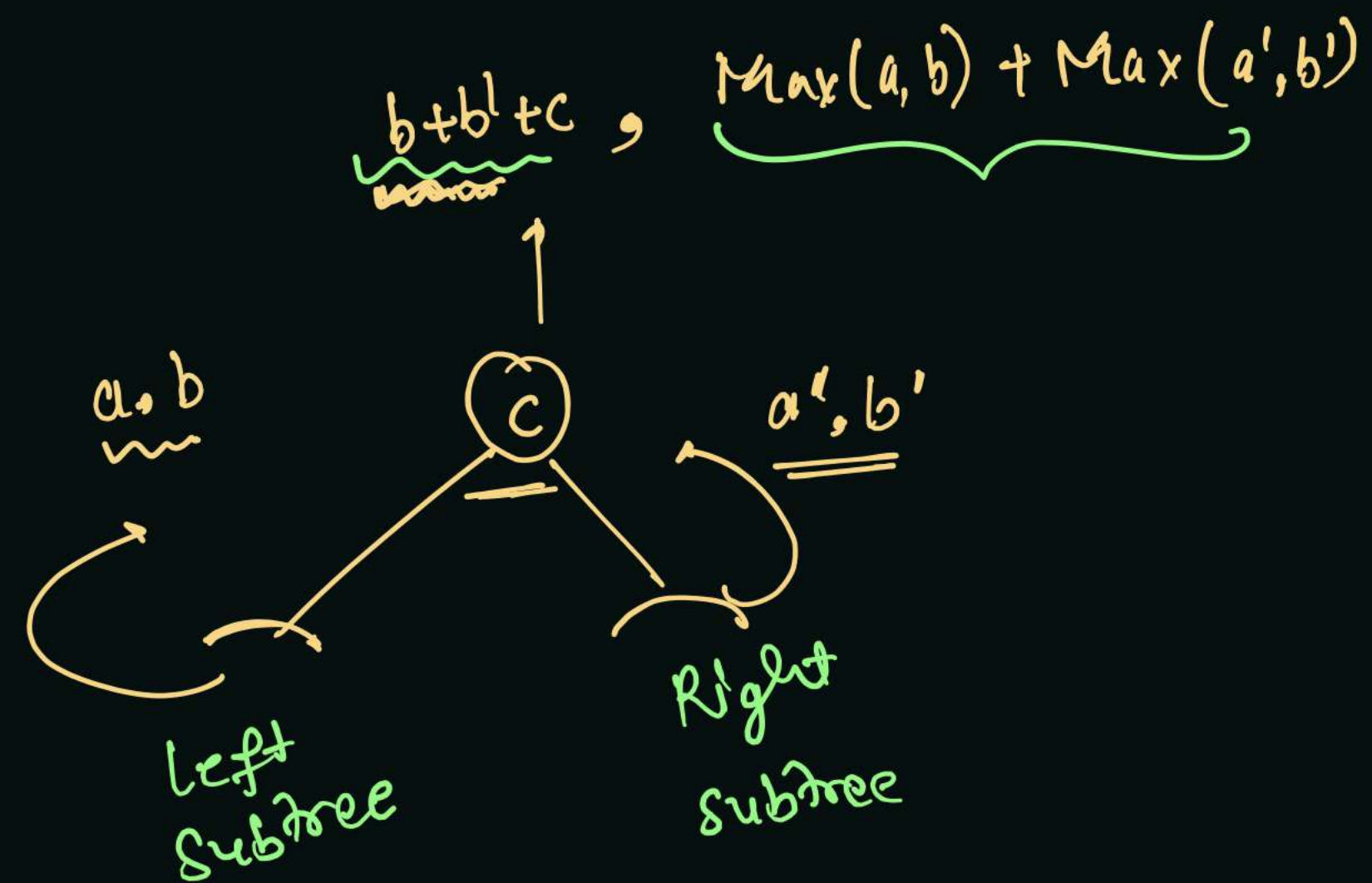


$c \rightarrow$  Robbery  $\checkmark$   $\{ \underline{b + b'} + c \}$  Res 1  
 $c \rightarrow$  Robbery  $\times$

$50 + 0 =$   
 $50 + 0 =$   
 $0 + 0 =$   
 $0 + 0 =$

$3 + 0 = 3$   
 $3 + 0 = 3$   
 $50 + 0 = 50$   
 $50 + 0 = 50$

$\left\{ \begin{array}{l} \underline{a + a'} \\ \underline{a + b'} \\ \underline{a' + b} \\ \underline{b + b'} \end{array} \right\} \underline{\underline{\text{Max}}} \Rightarrow \underline{\underline{\text{Res 2}}}$

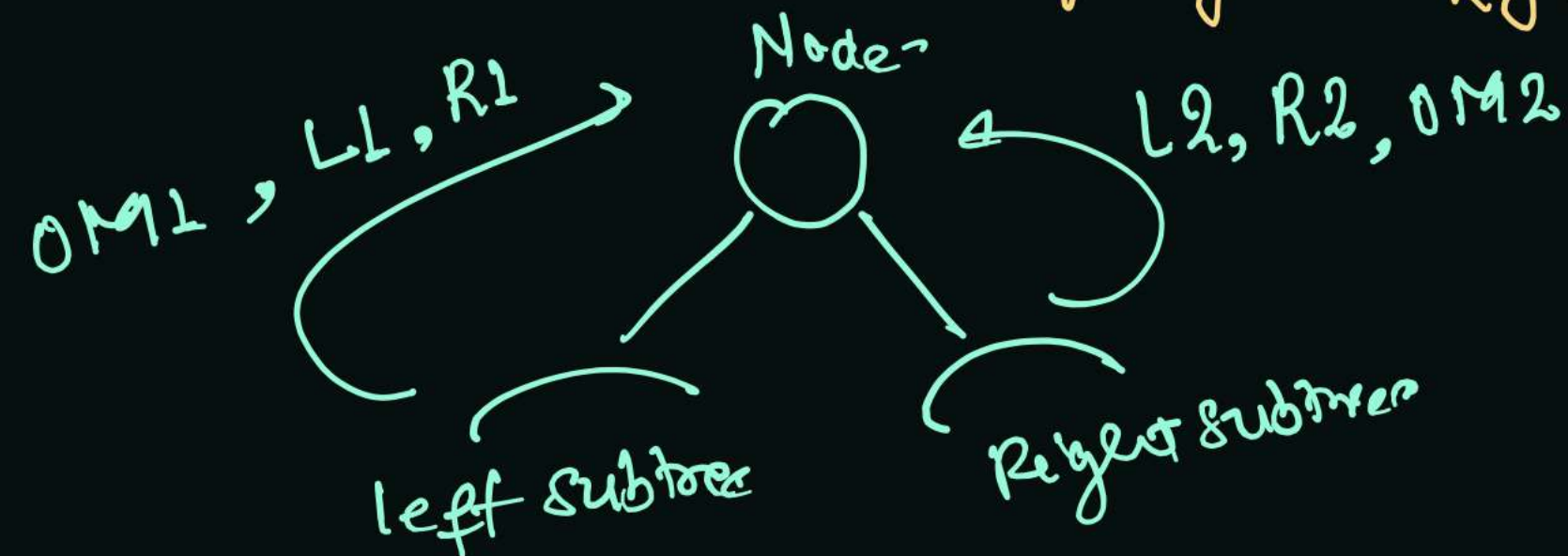




Saturday, 14 August 2021 7:07 PM

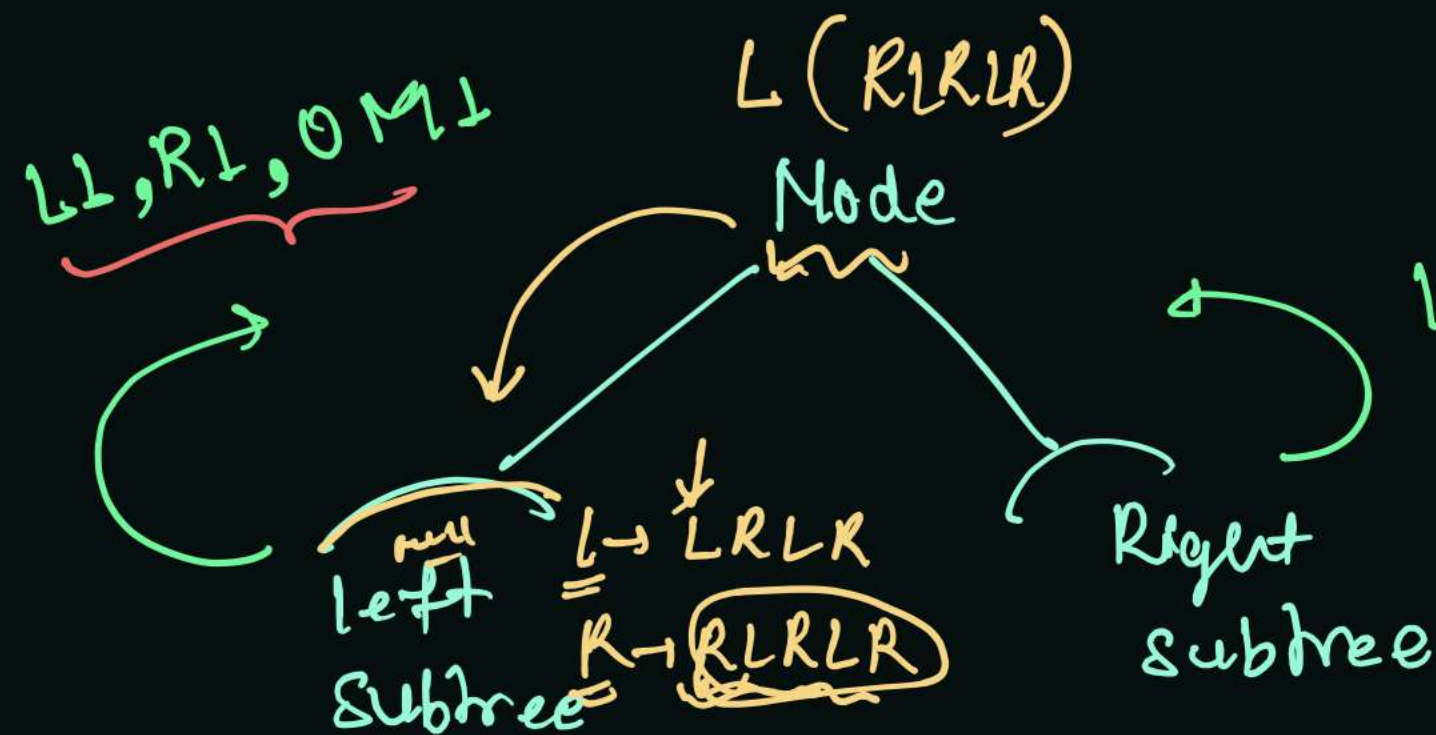
Saturday, 14 August 2021

7:07 PM





Base case  
root == null  
return -1



possibility of  
 answer →

$L1 \rightarrow$  Longest ZigZag of left subtree start at L

$R1 \rightarrow$  " " " " start " R

$OM1 \rightarrow$  Longest " " left subtree

$L2 \rightarrow$  Longest zigzag of right subtree start at L

$R2 \rightarrow$  " " " " " start " R

$OM2 \rightarrow$  " " " " right subtree.

Start at L →

Increment of length

$$\text{length} = R1 + 1$$

Start at R →

Increment of length

$$\text{length} = L2 + 1$$

omax

omax

$$\left[ \begin{array}{l} \underline{OM1} \text{ vs } \underline{OM2} \text{ vs} \\ \underline{R1+1} \text{ vs } \underline{L2+1} \end{array} \right]$$





# Validate BST

Saturday, 14 August 2021

7:08 PM

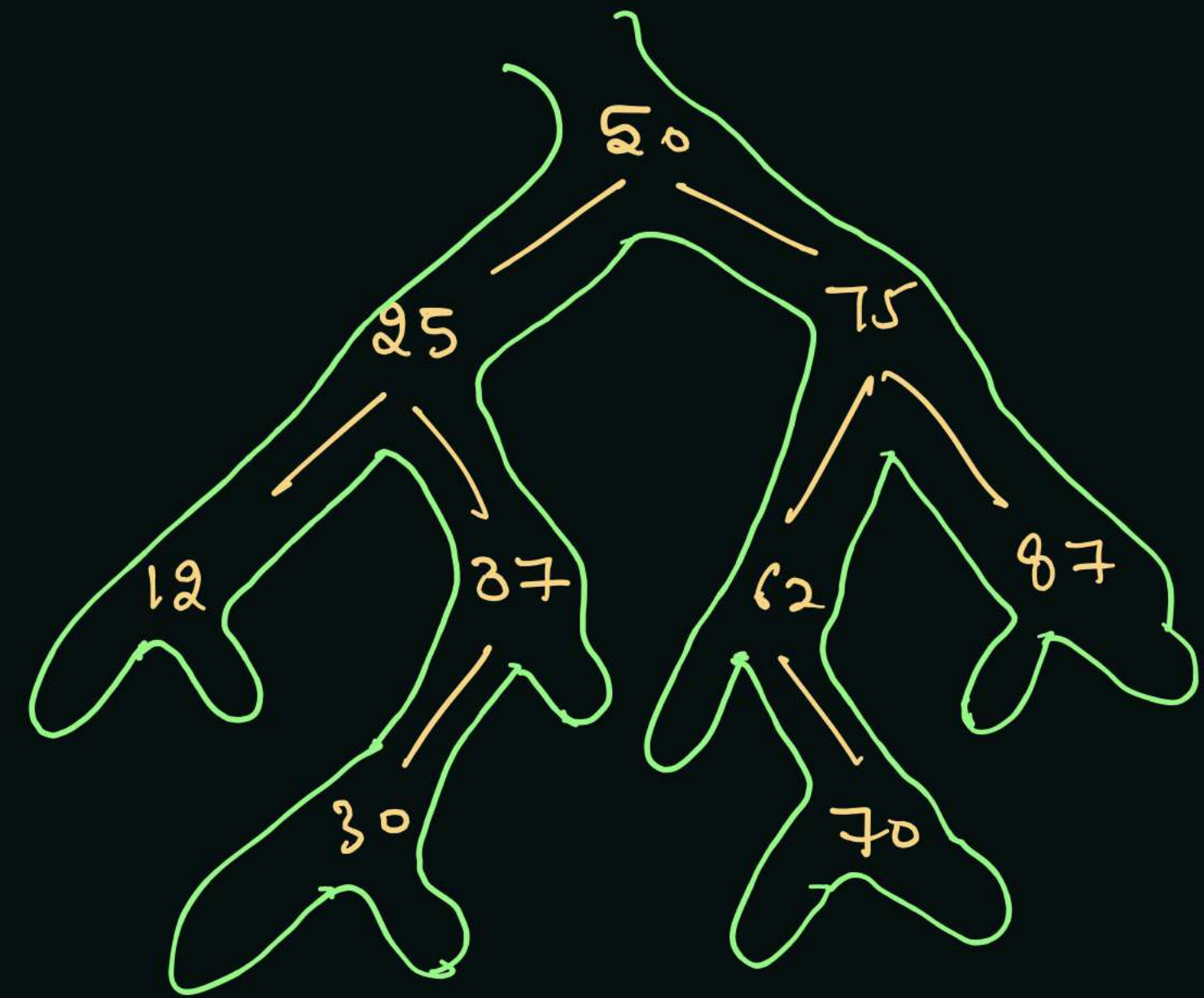
17. W.

prev = ~~12~~ ~~25~~ ~~30~~ ~~37~~ ~~50~~ ~~62~~ ~~70~~ ~~75~~ 87

curr = 25 ~~30~~ ~~37~~ ~~50~~ ~~62~~ ~~70~~ ~~75~~ 87

```
{ if( prev.val > curr.val ) {  
    return False;  
}
```

Base Idea = ?



→ # In Order of BST is always sorted.

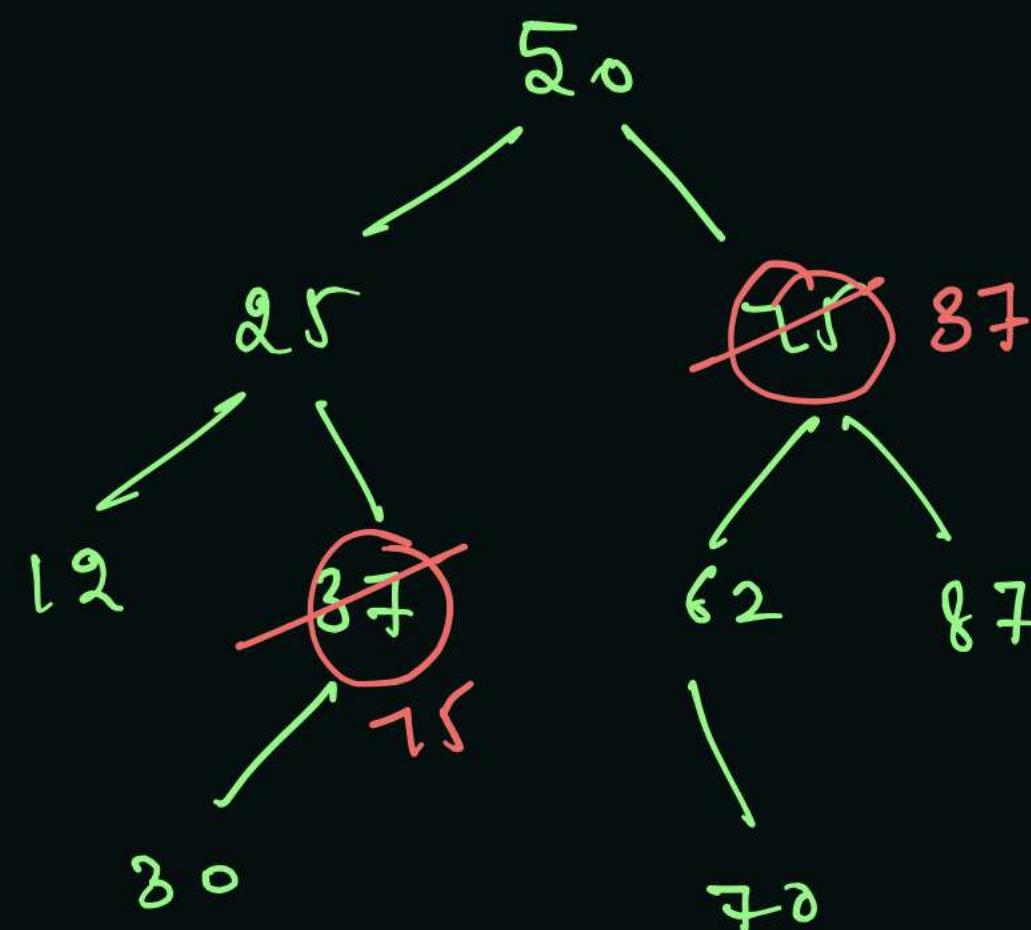
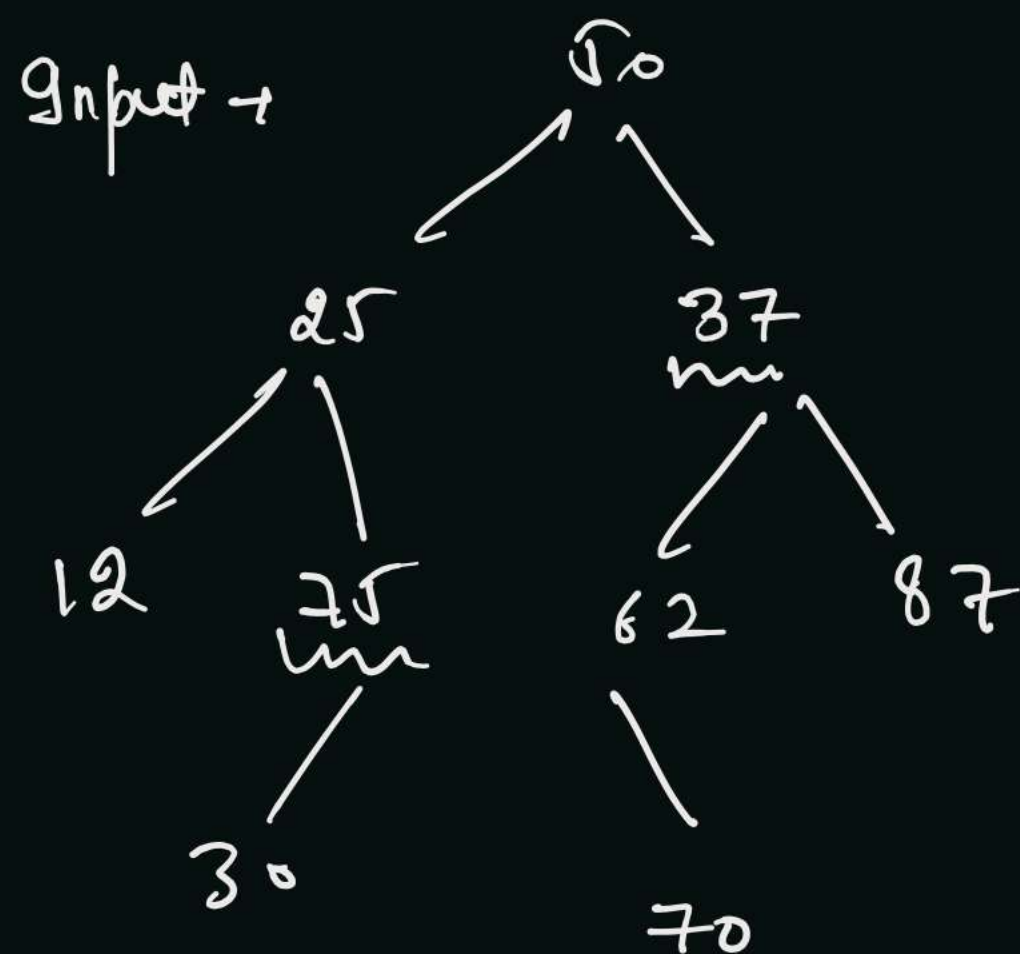
what not to do?

- (1) inorder get and check if it is sorted or not
- (2) Min - Max Strategy.

Right way

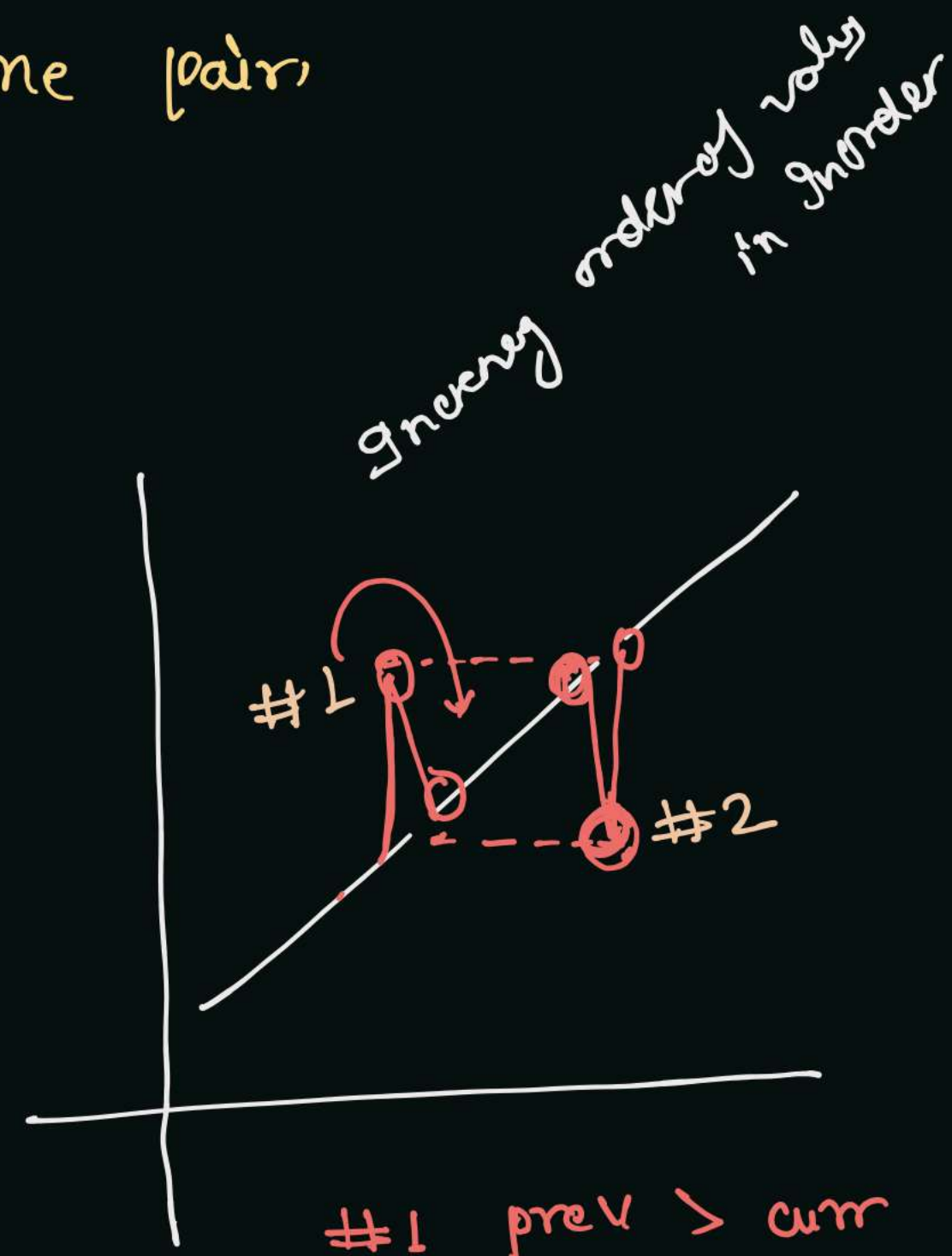


All nodes are perfectly placed except one pair,  
find that pair and fix it.



# Edge case → Just Adjacent  
values are swapped

A.W.



#1 prev > curr

#2 prev > curr