

Binary Tree



Doubt Class with Lakshay Bhaiya

Special class

$BT \rightarrow DS$

Root



Siblings

Parent Node

Child

Leaf

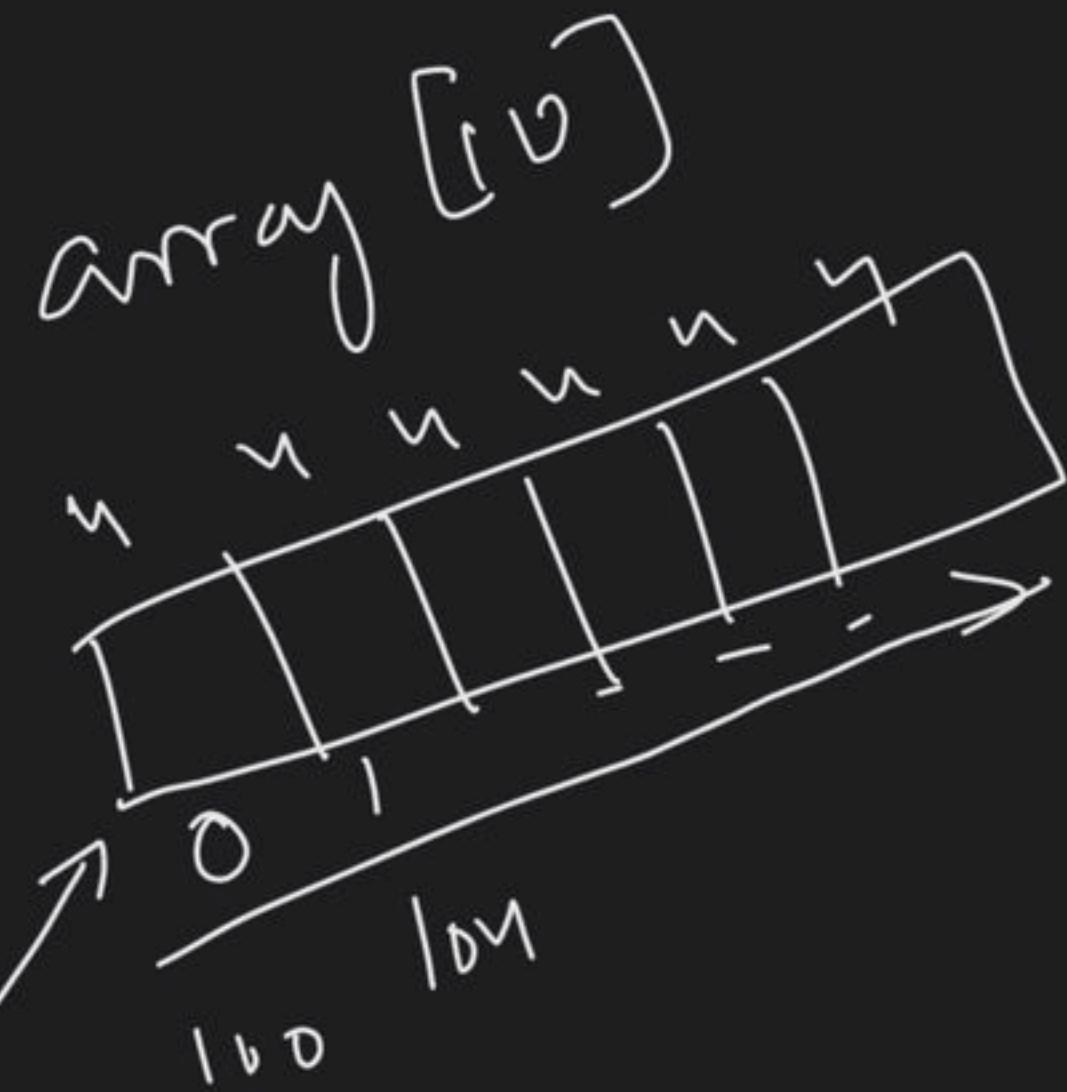
Subtree



Anchor \hookleftarrow Descender

Level

\Rightarrow

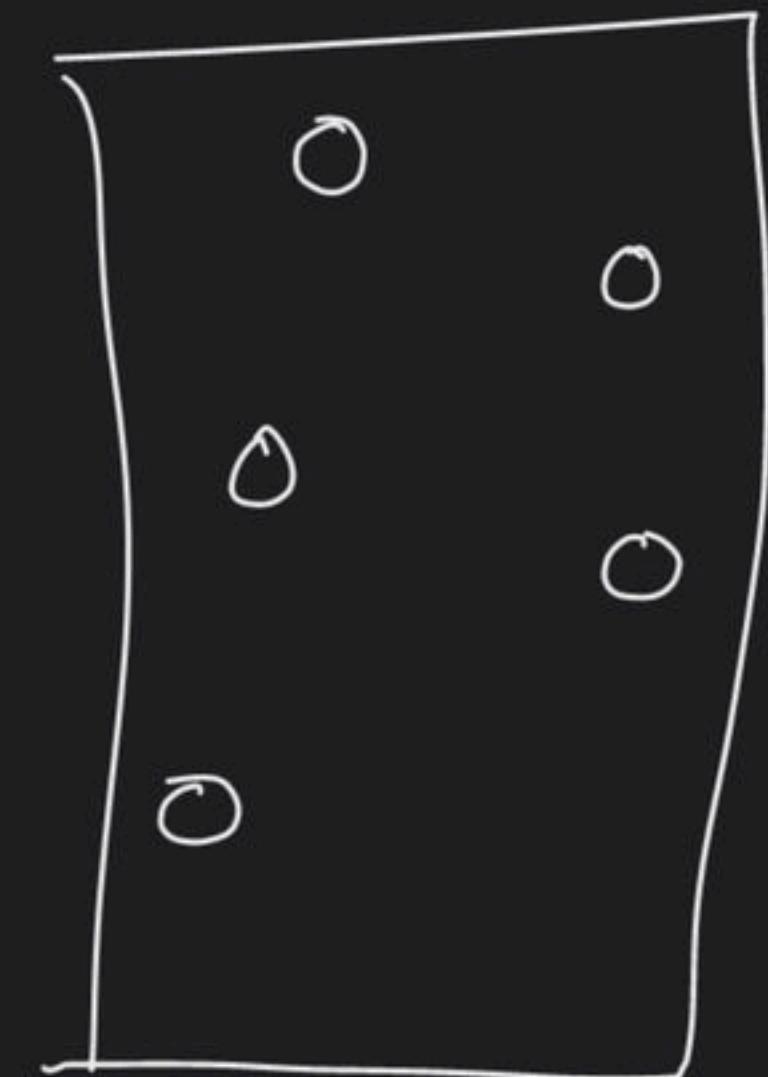


dynamic ✓

$\hookrightarrow \text{new} \rightarrow \text{heap}$

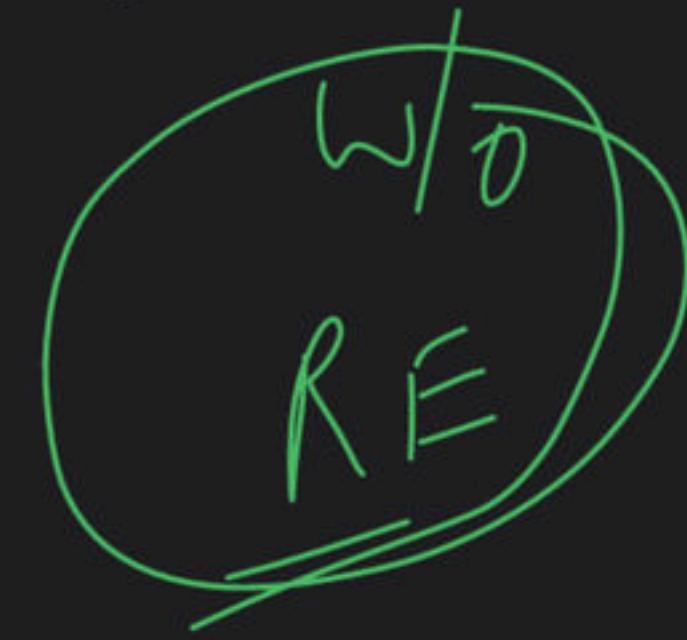
non-contiguous \rightarrow

$\circ \rightarrow \circ$



\Rightarrow 3 traversals

Stack



✓ Preorder \rightarrow NLR

✓ Inorder \rightarrow LNR

✓ Postorder \rightarrow LRN

Levelorder \rightarrow Queue \rightarrow BFS

$\Rightarrow ① \text{ front} = 1$

$② \text{ front} = \emptyset$

$③ F = 2$

$④ F = 3$

$⑤ F = \underline{\text{null}}$

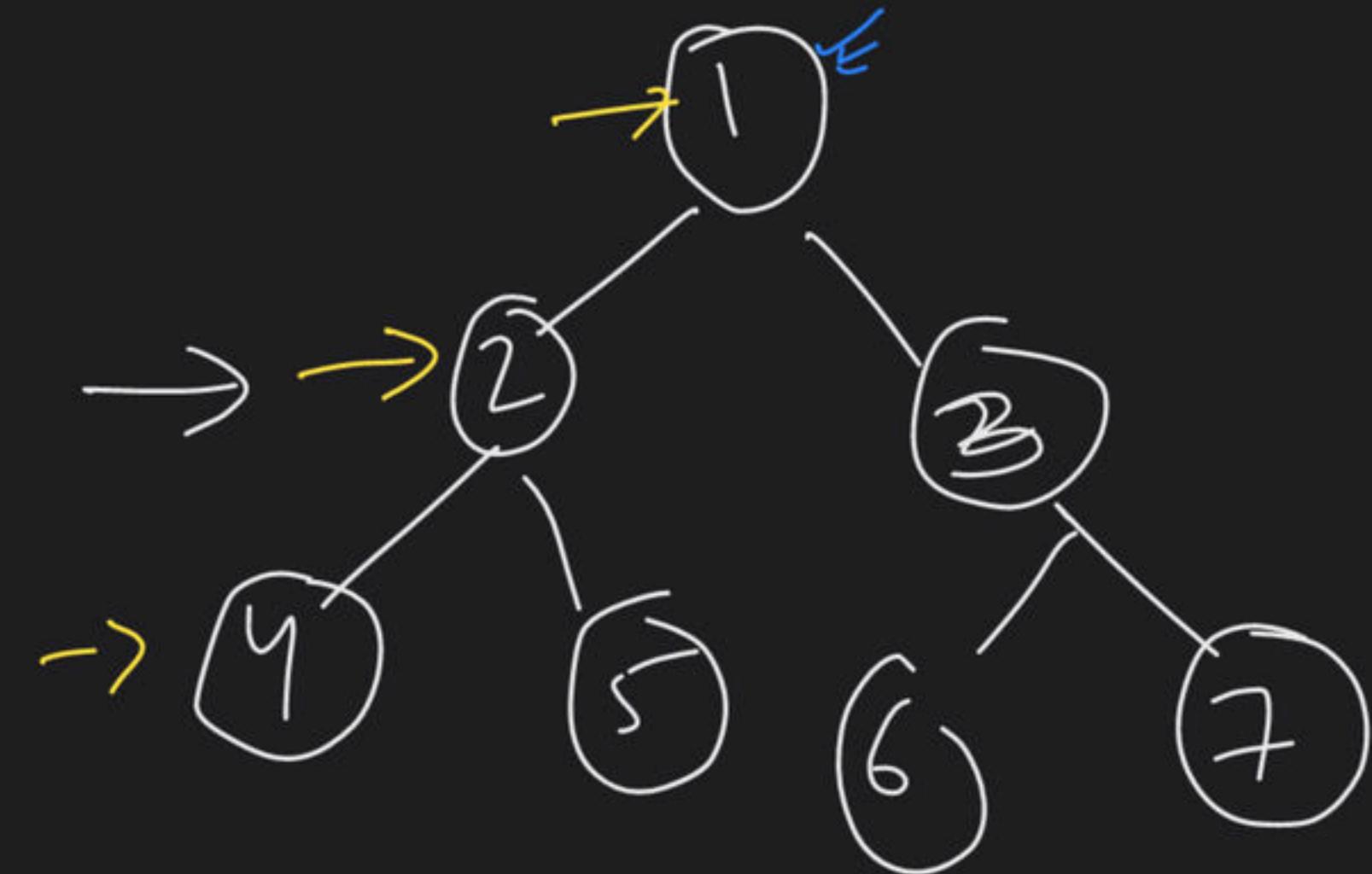
$⑥ F = 4$

$F = 5$

$F = 6$

$F = 7$

1
2 3
4 5 6 7



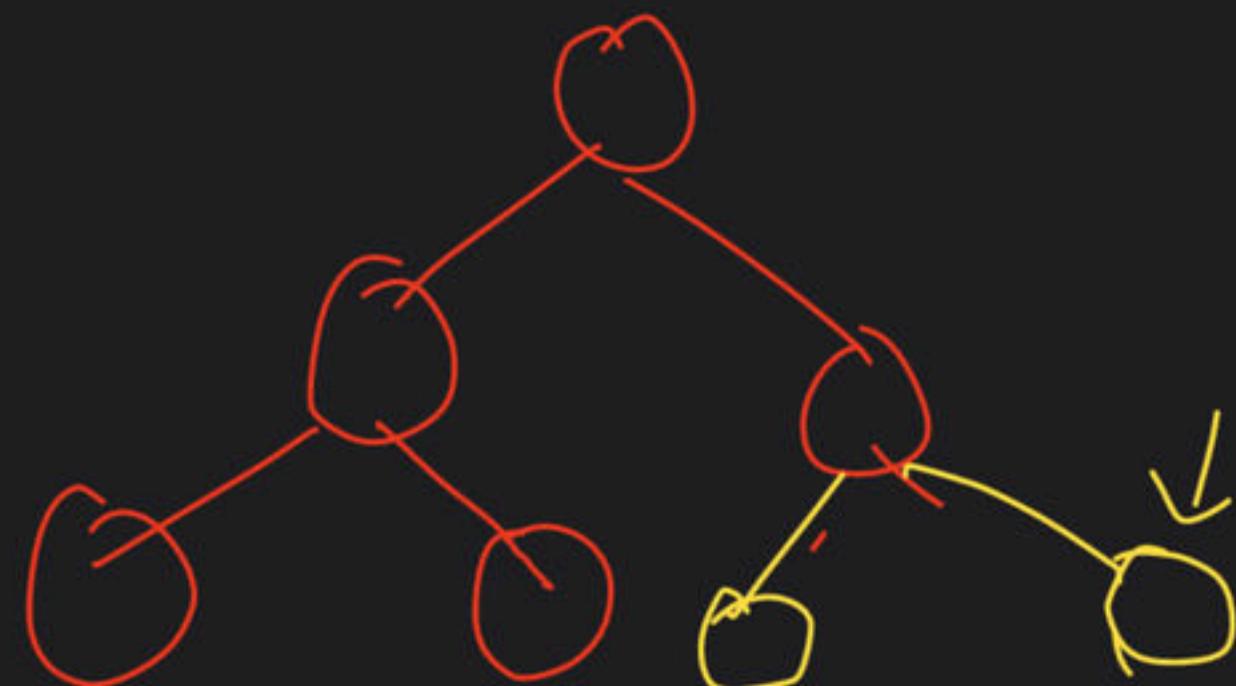
$F = \underline{\text{null}}$

~~start RB tree~~ [4 5 6 7] ~~node~~

Height → ✓ → Fast way??

Diameter → ✓
=

⇒ Complete BT ⇒ all levels, except last level
are filled & all nodes are
as left as possible.

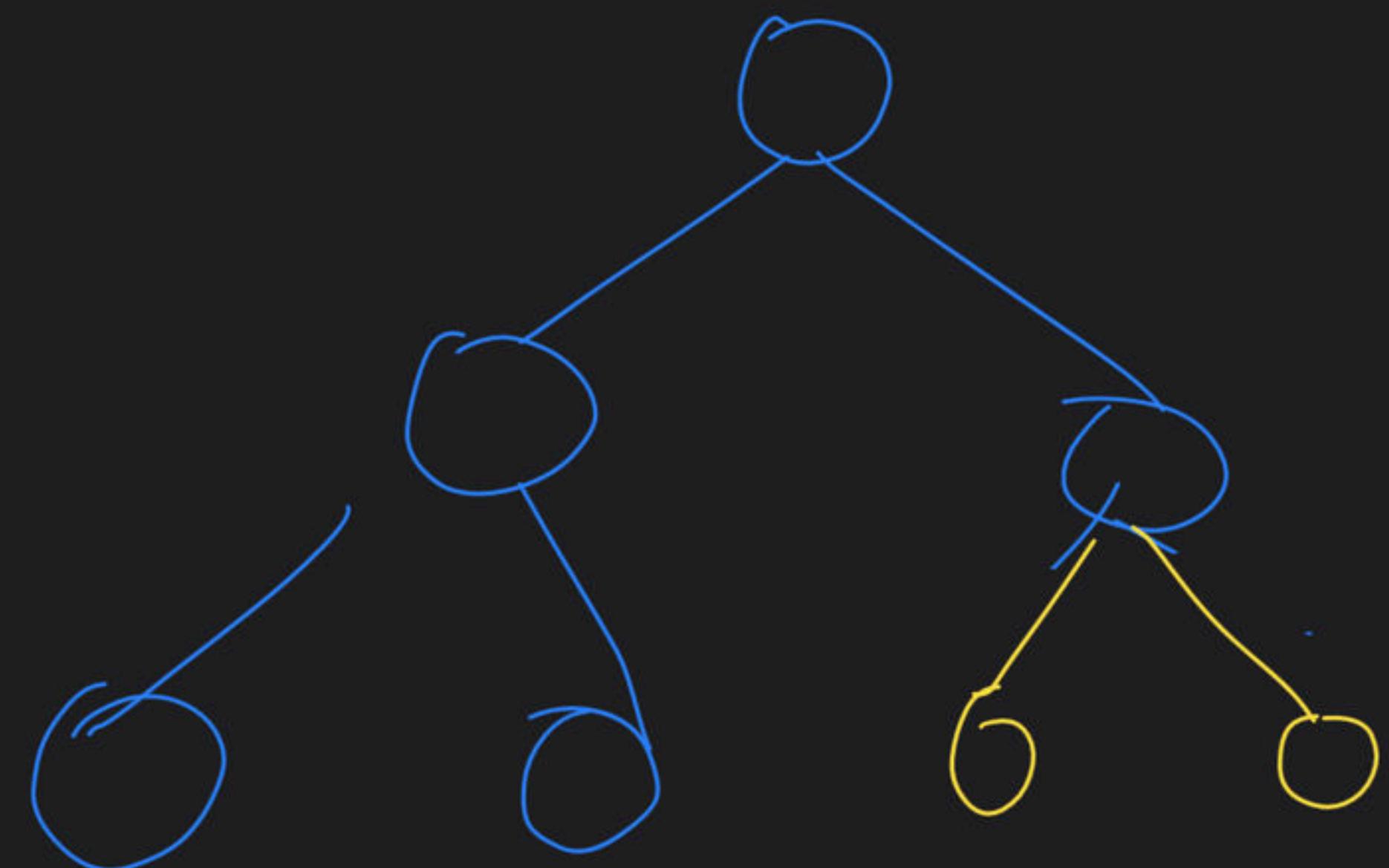


Perfect BT

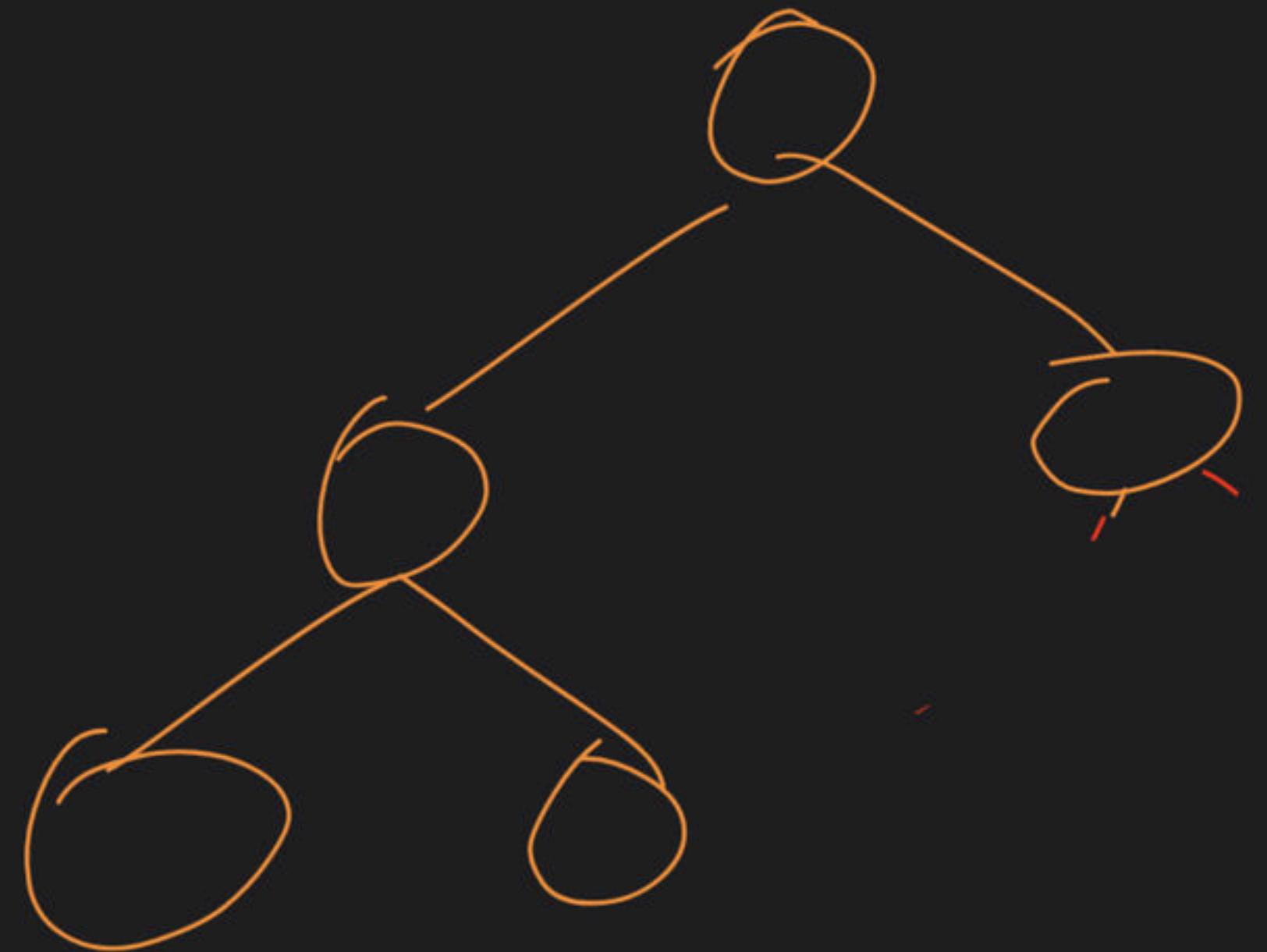
~~all~~ nodes

exactly 2 child

→ except leaf nodes
& leaf are at
same level



full BT \rightarrow ~~All~~ Nodes has either 0 or
2 child nodes

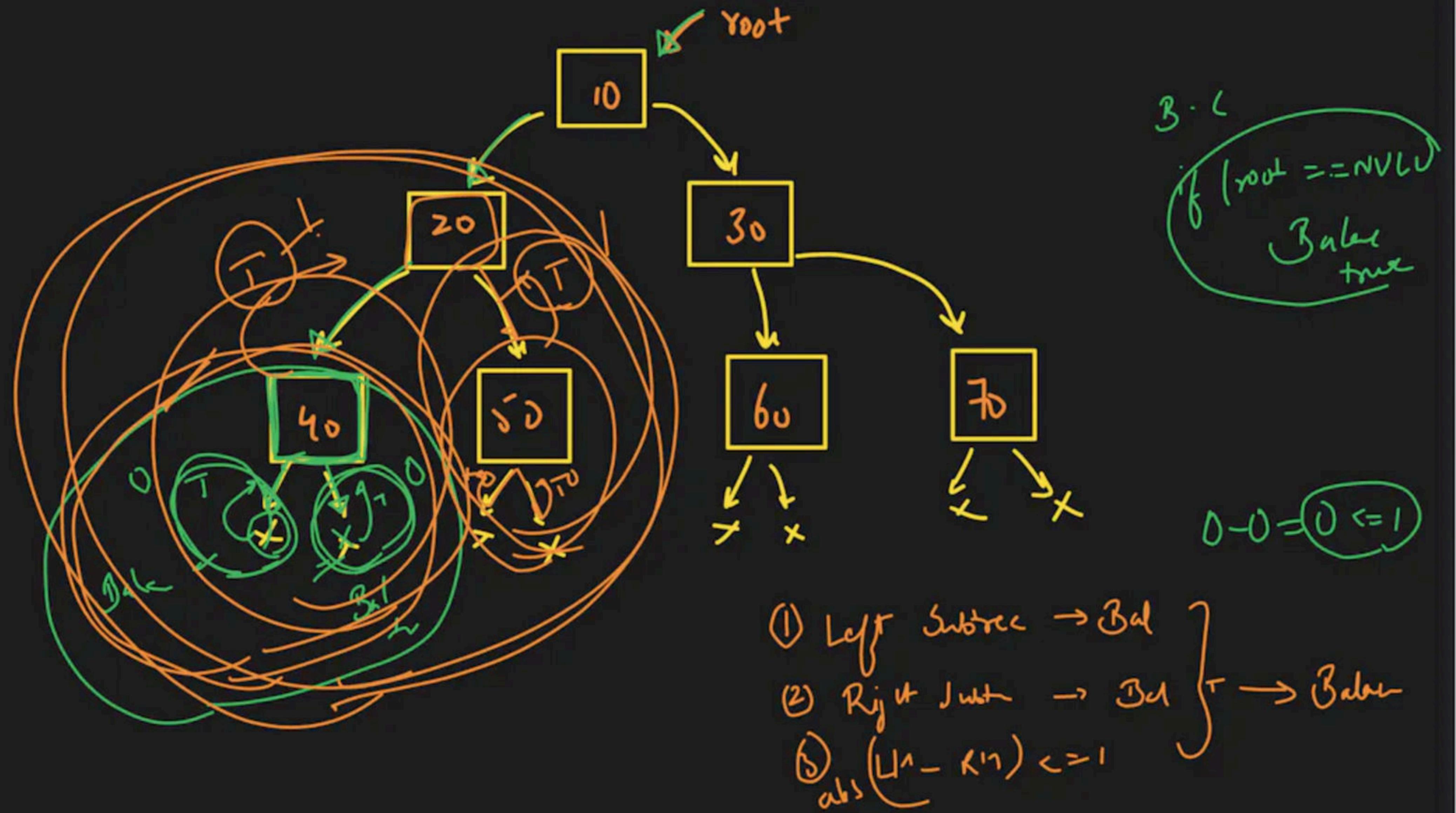


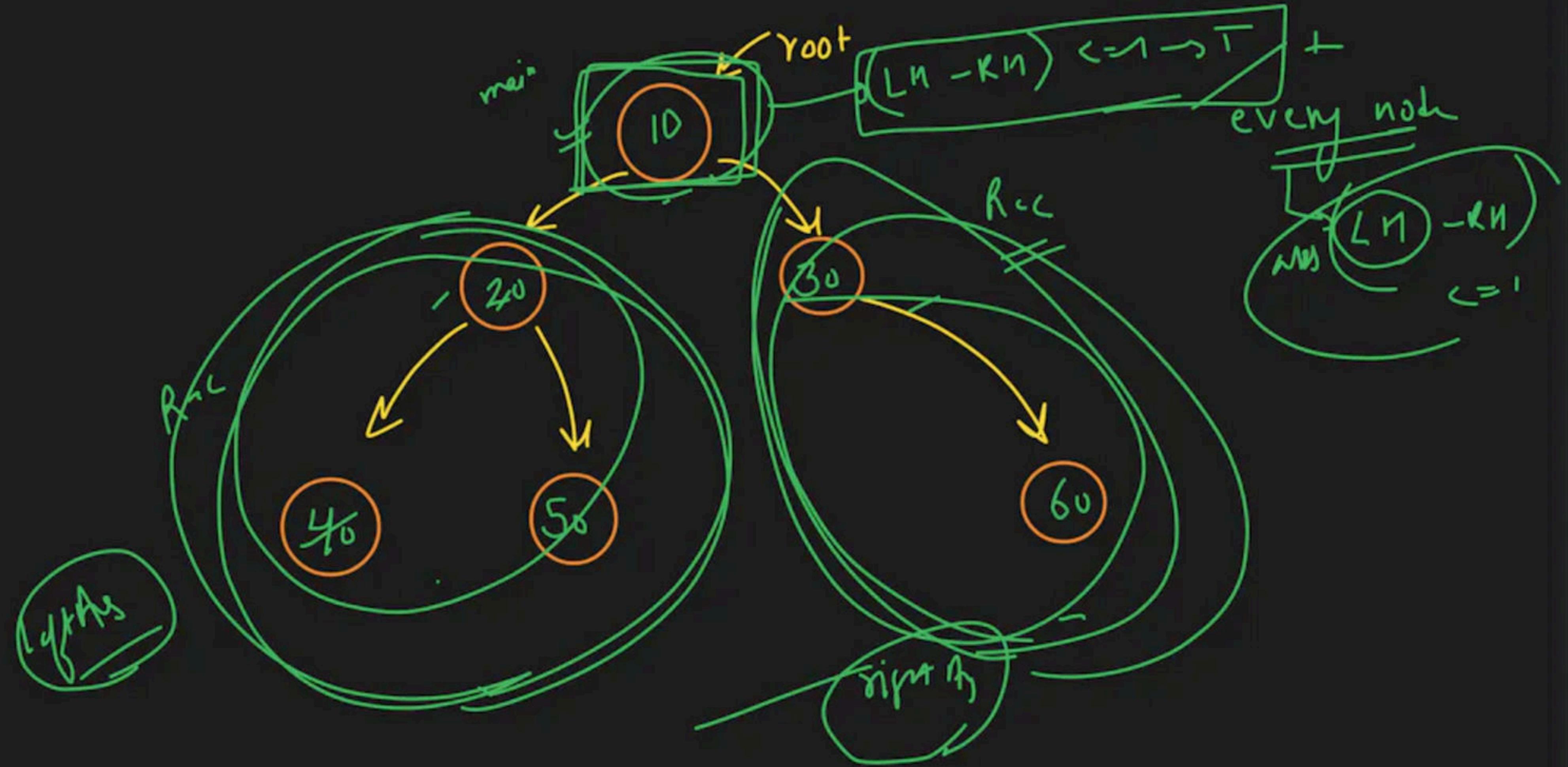


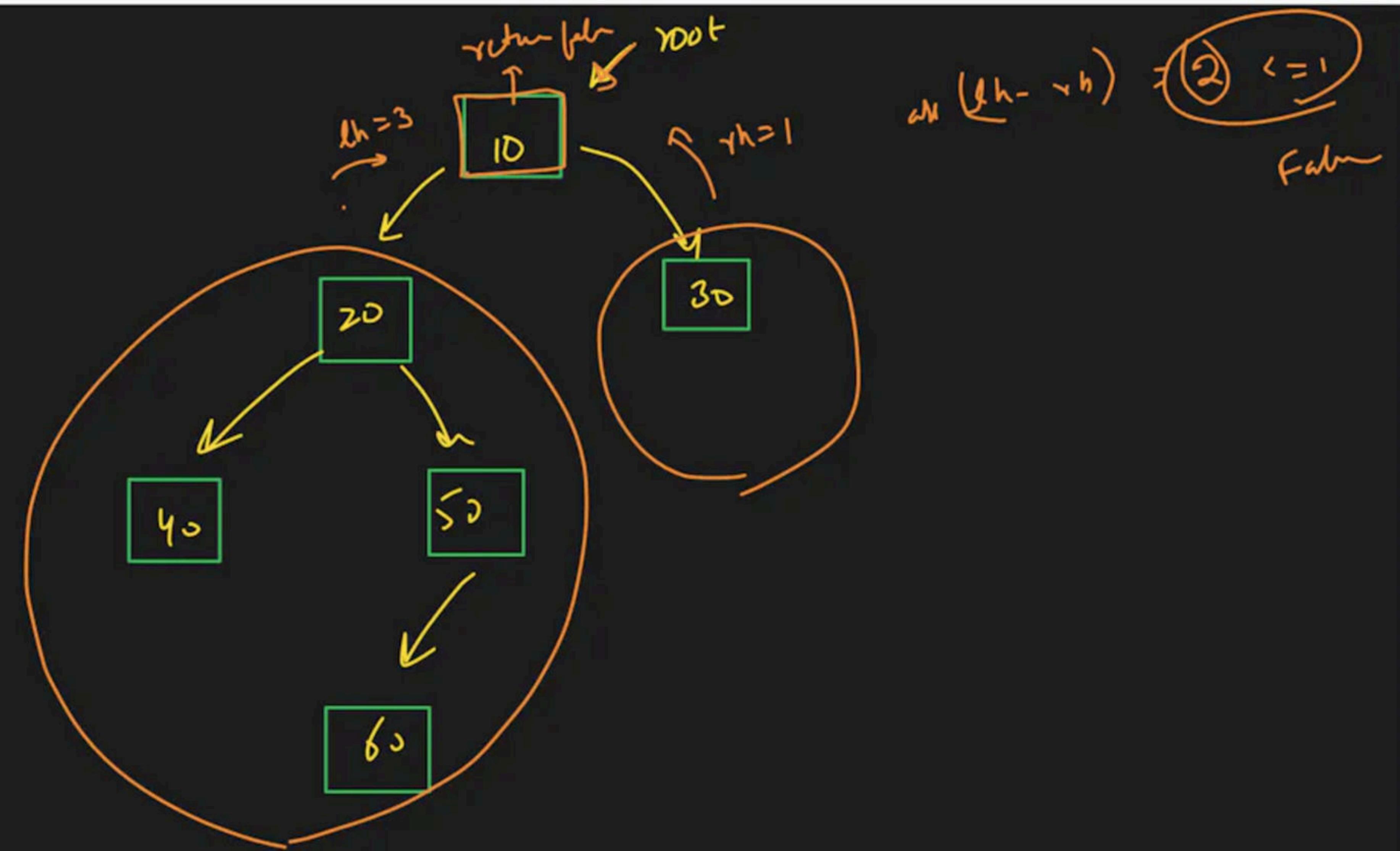


Binary Trees Class - 2

Special class

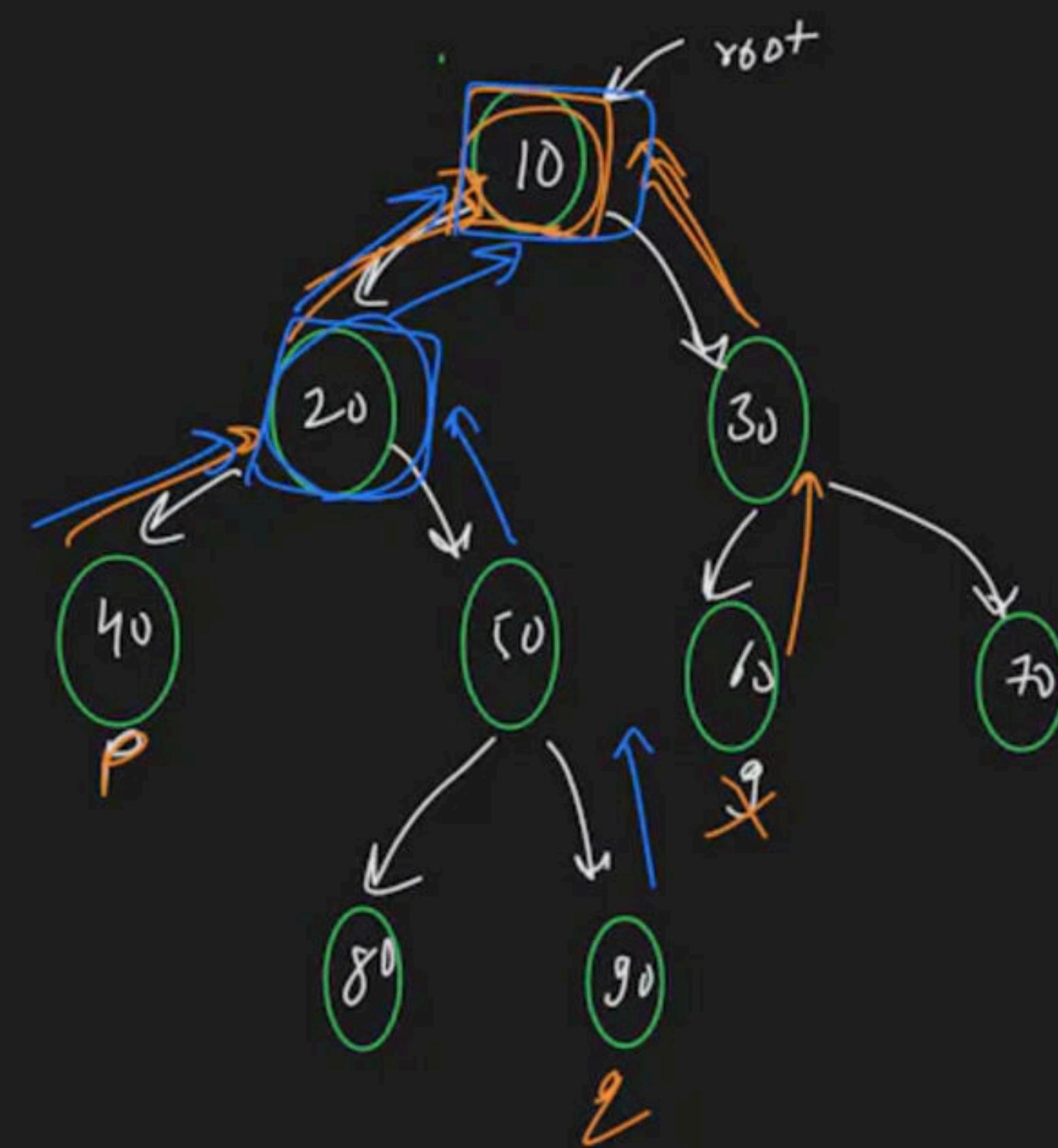


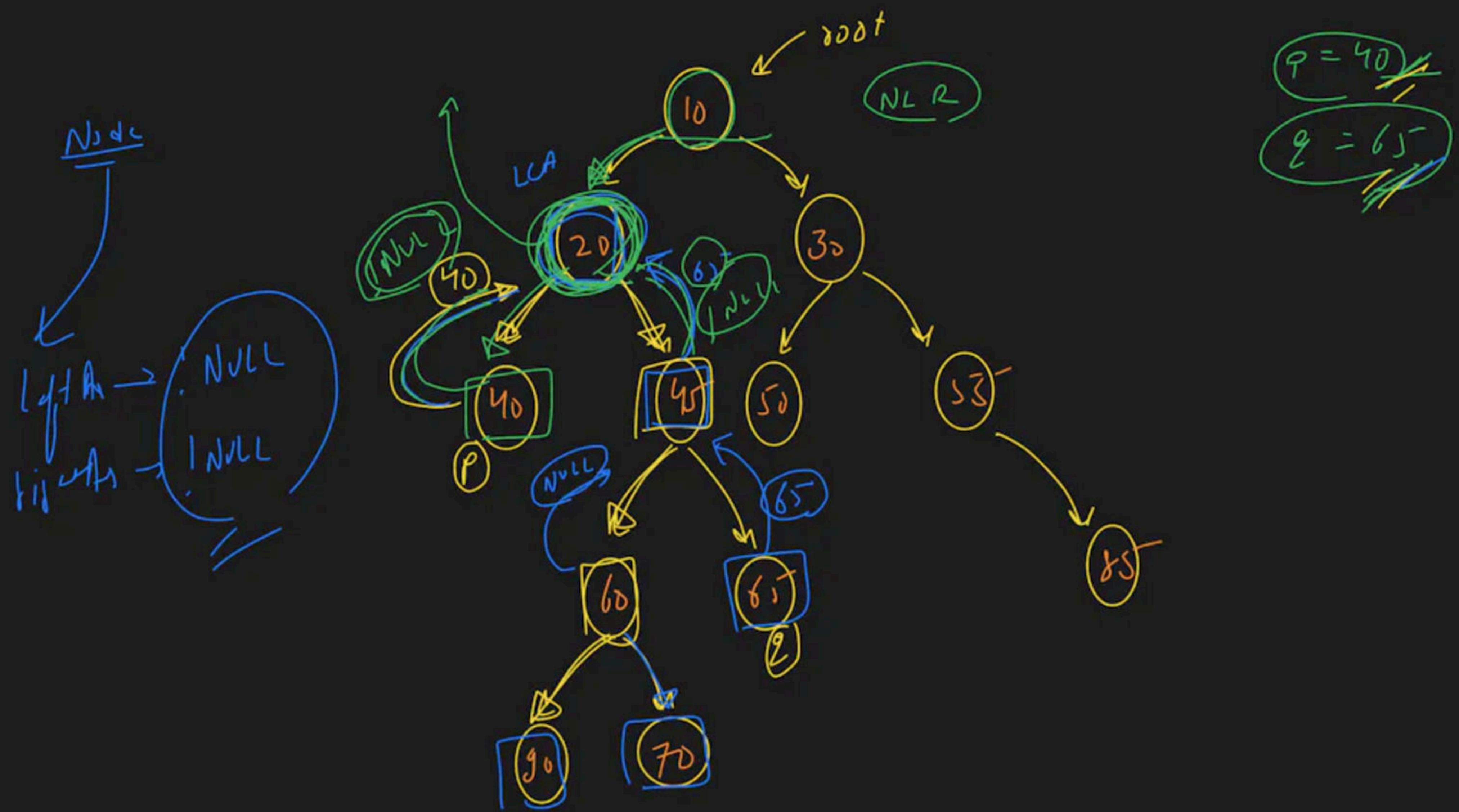


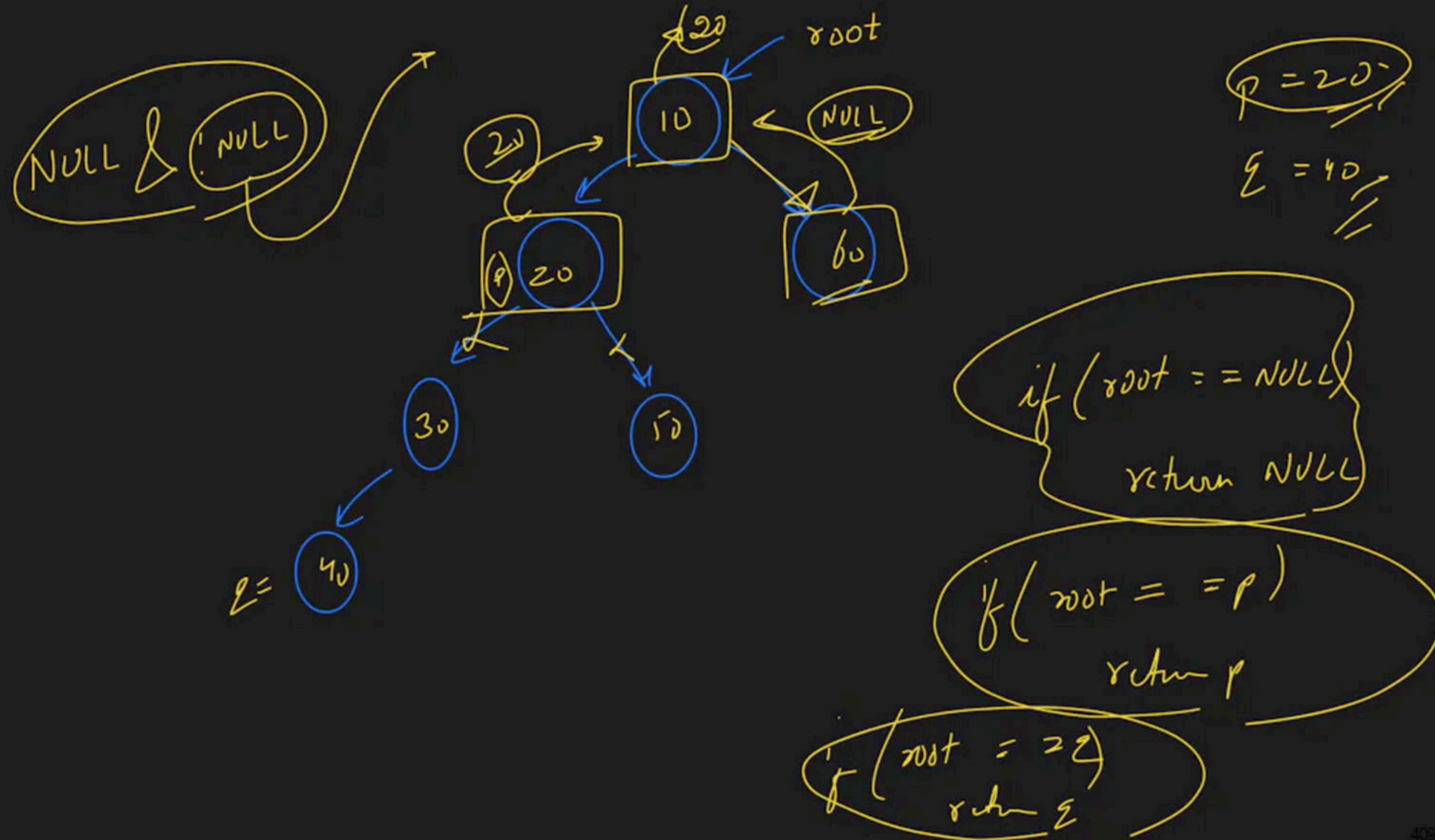


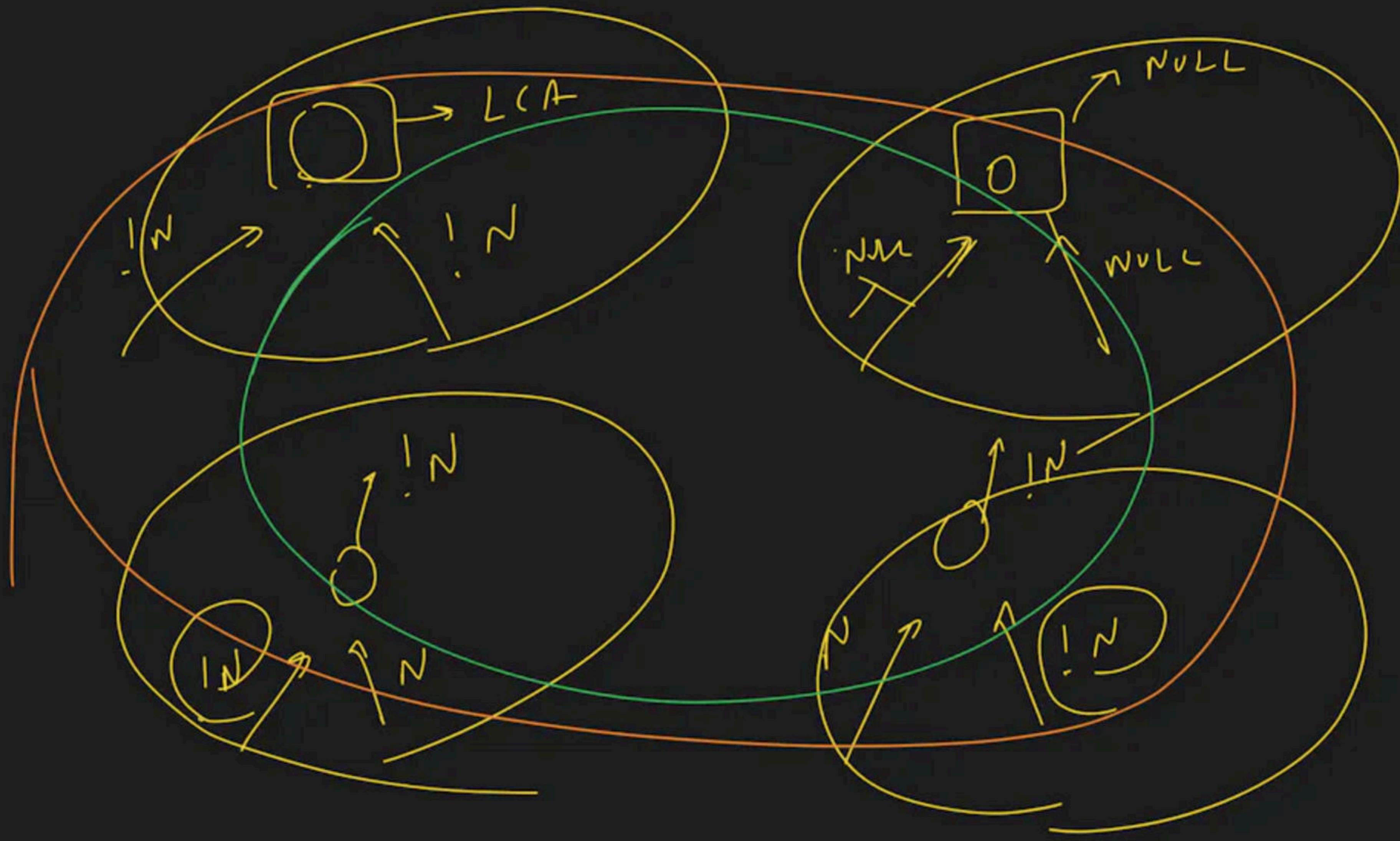
LCA → ?

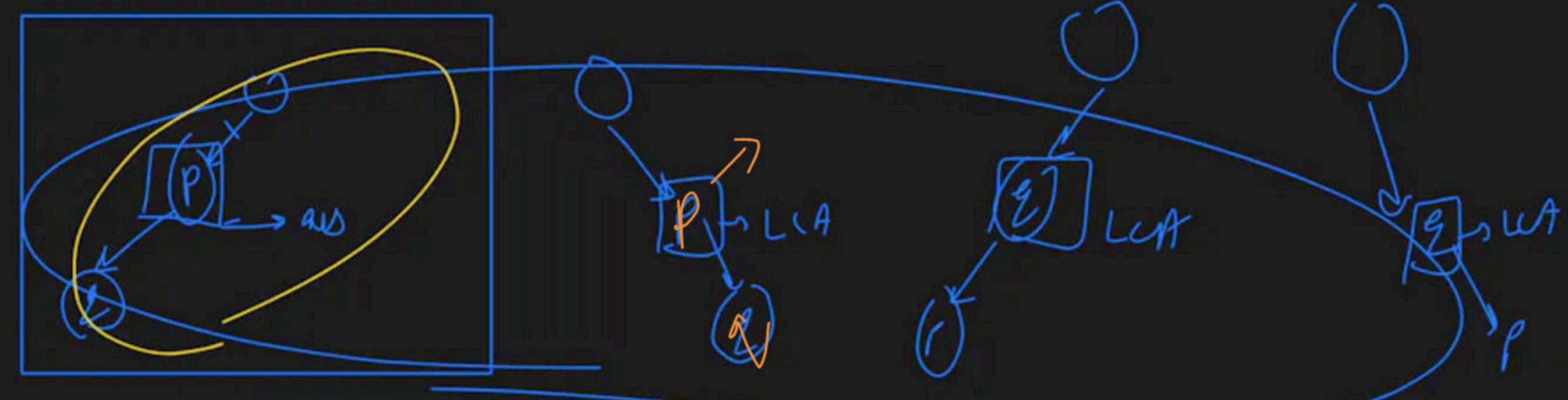
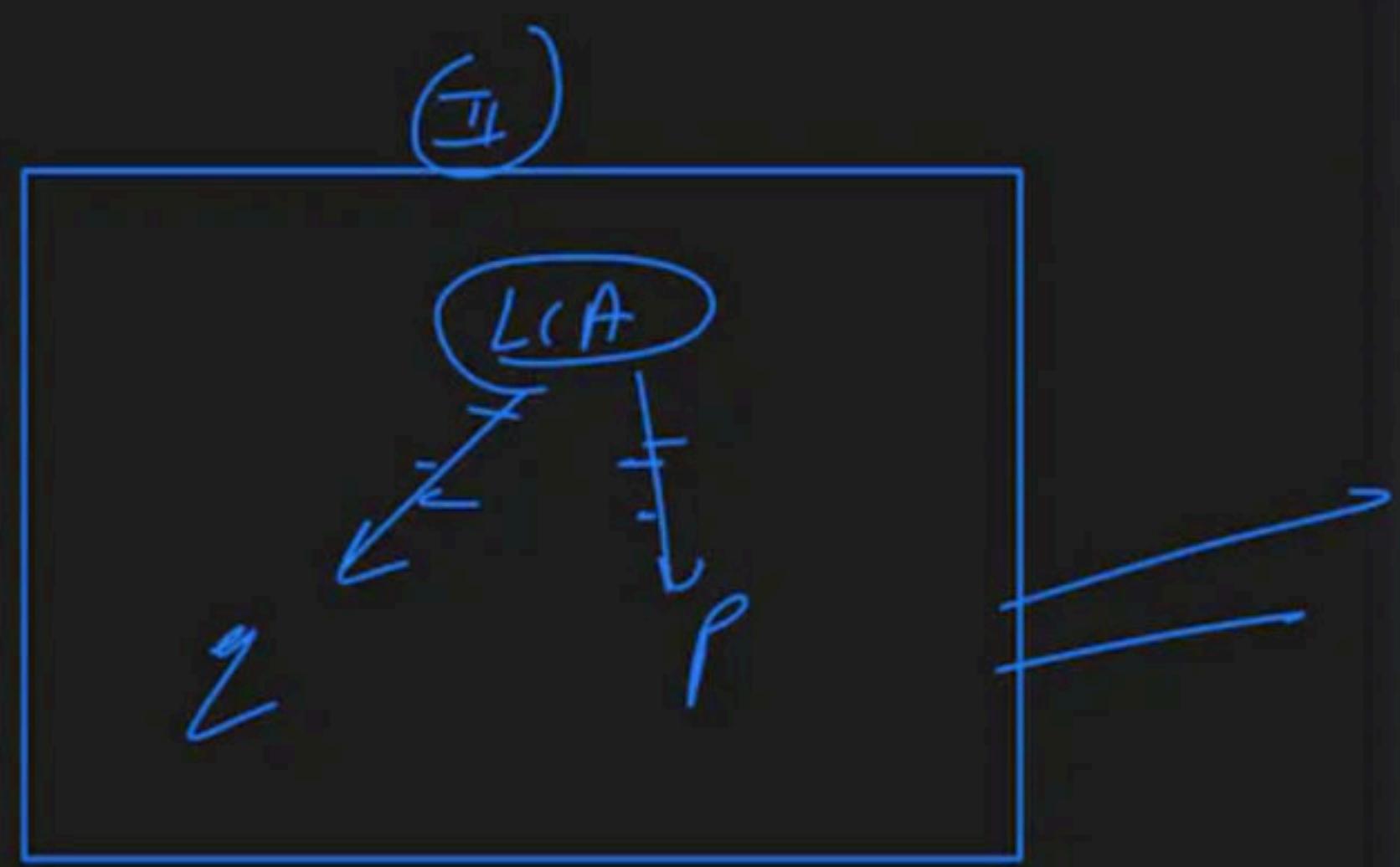
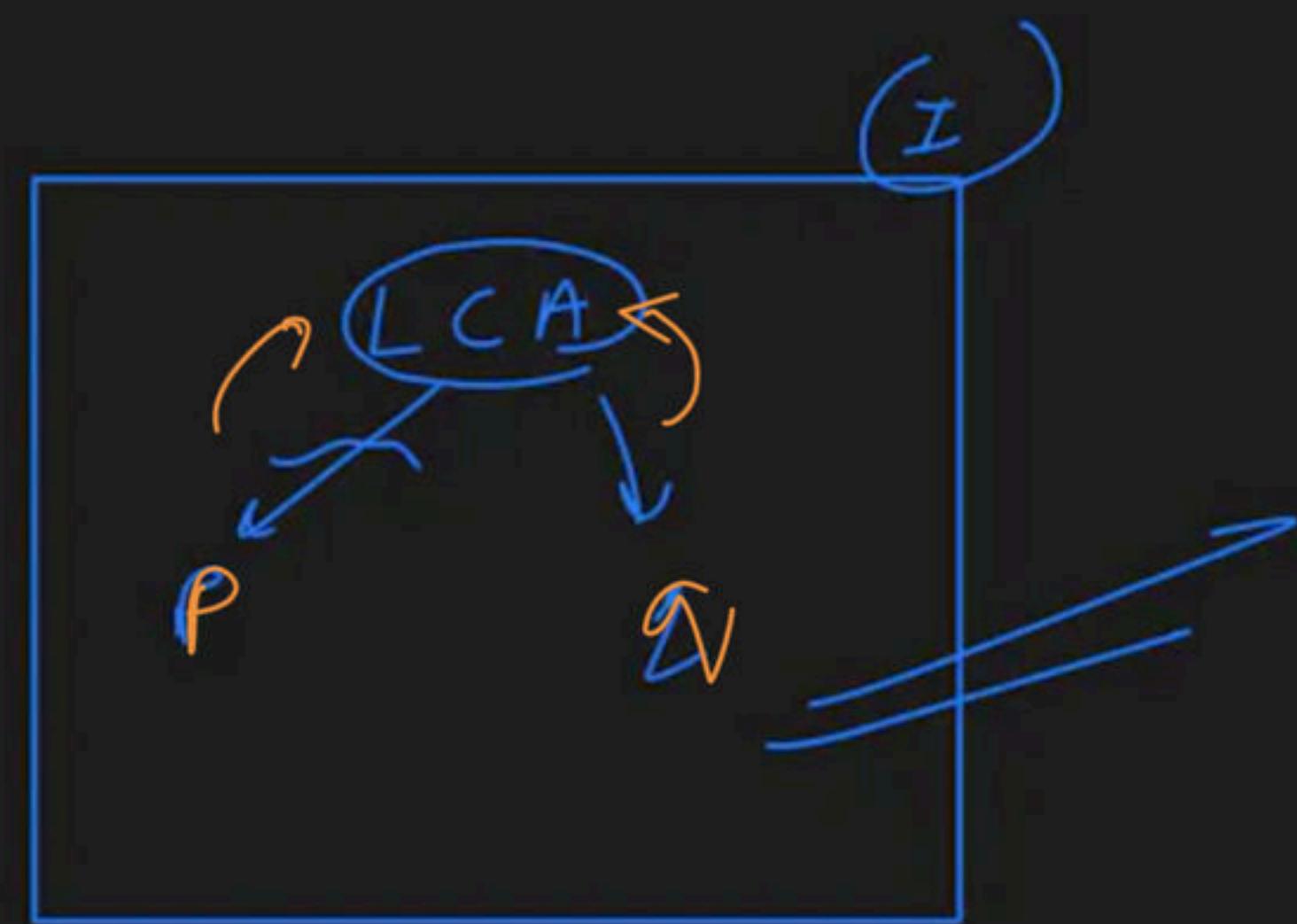
$\wp \rightarrow P \cup I$





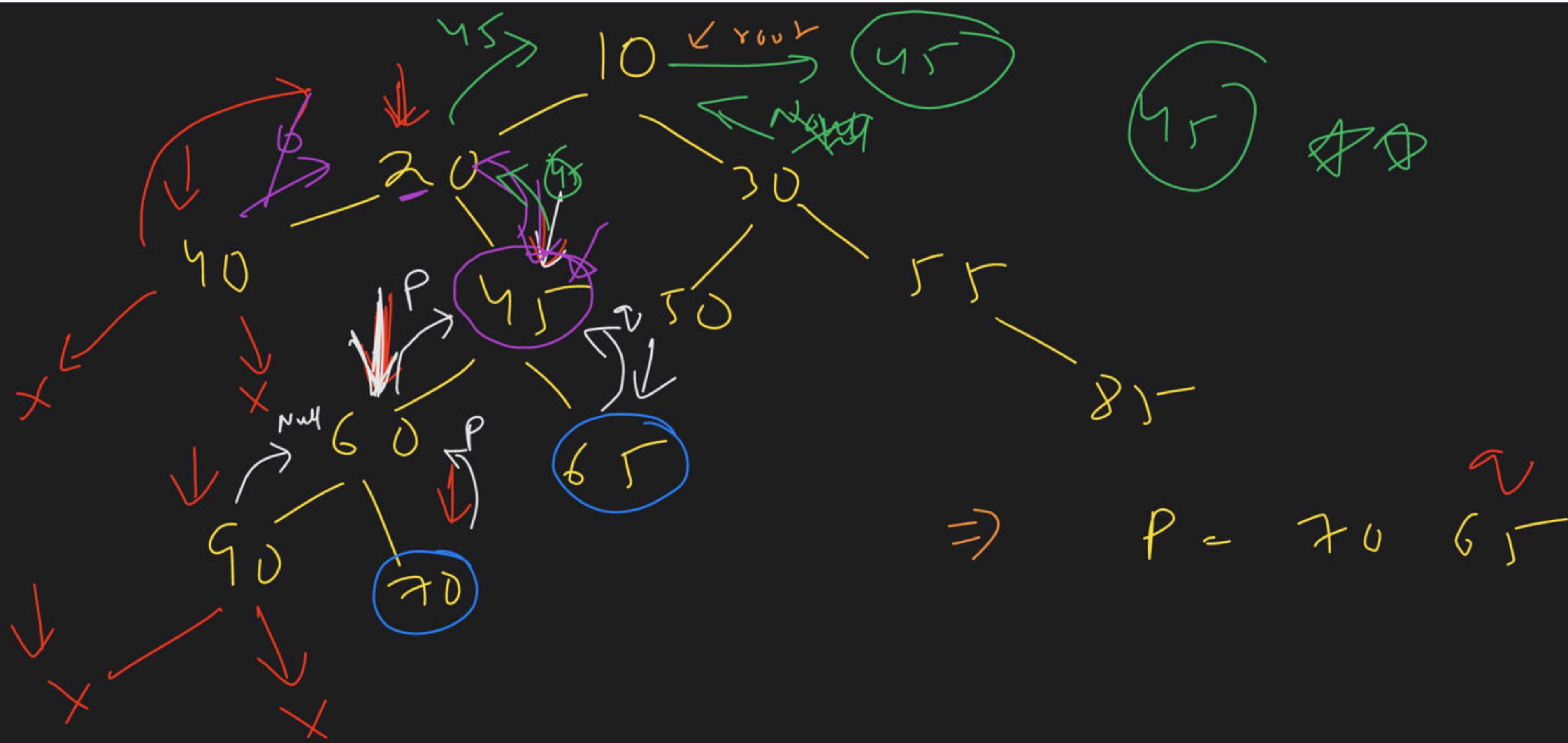




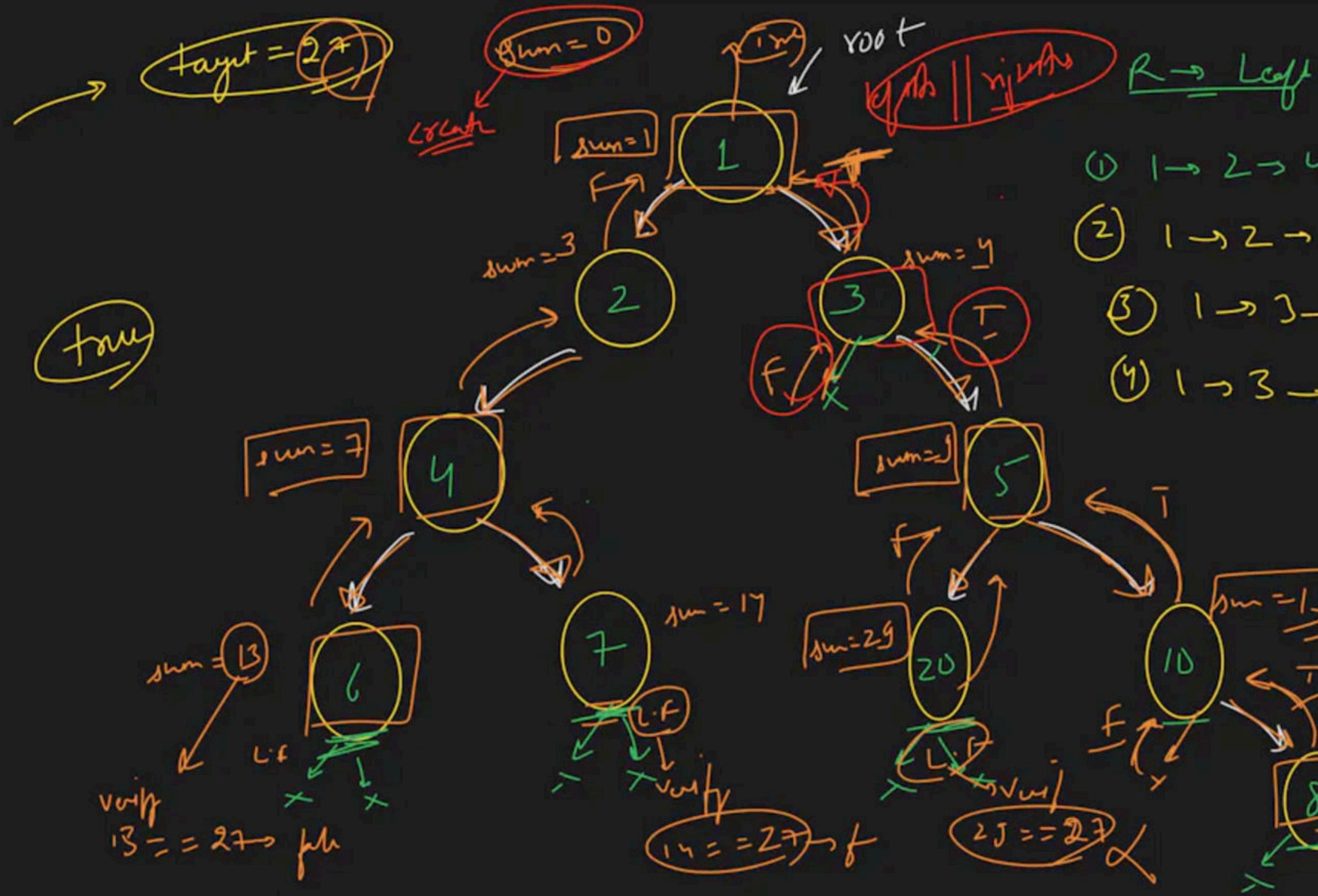


TreNode * lowestCommonAncestor (TreNode * root, int p, int q) By Recursion

```
{  
    if (root == NULL) return NULL;  
  
    => if (root->val == p) return p;  
    => if (root->val == q) return q;  
  
    if (left == null) {  
        if (right == null) {  
            if (ans == null) ans = root;  
        } else if (right->val == p || right->val == q) return right;  
    } else if (right == null) {  
        if (left->val == p || left->val == q) return left;  
    } else {  
        if (left->val == p && right->val == q) return root;  
        if (left->val == q && right->val == p) return root;  
    }  
}
```

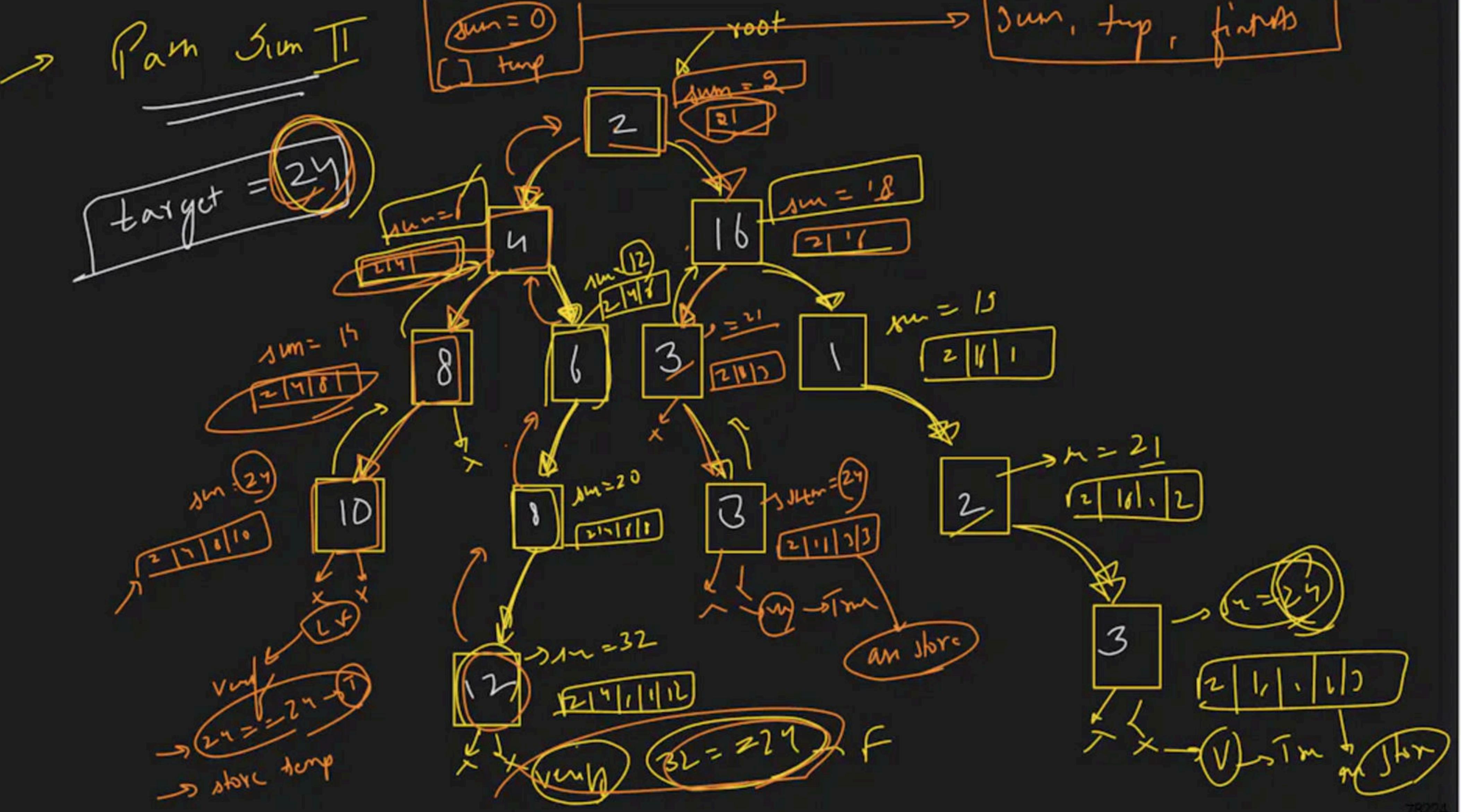


min
Boek

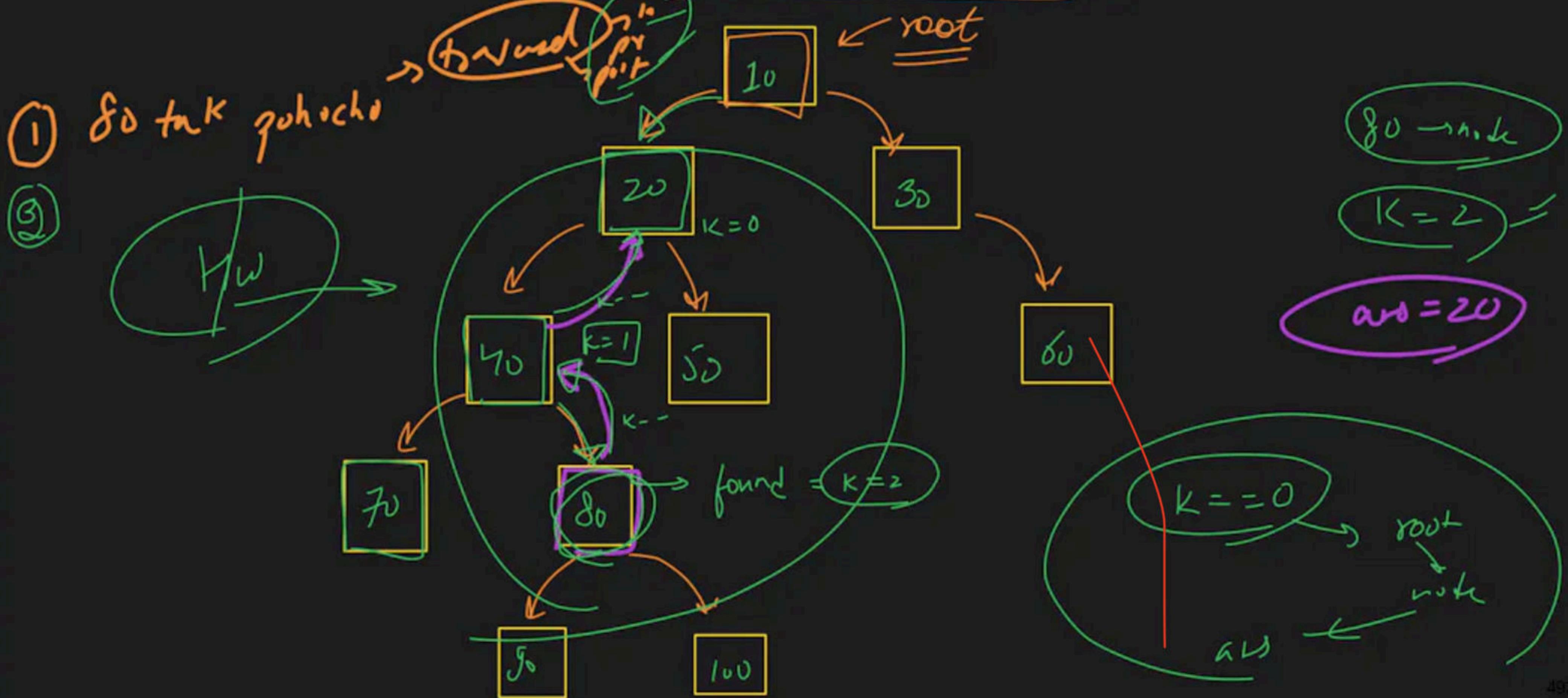


→ Pan Sim II

$$\text{target} = 2y$$

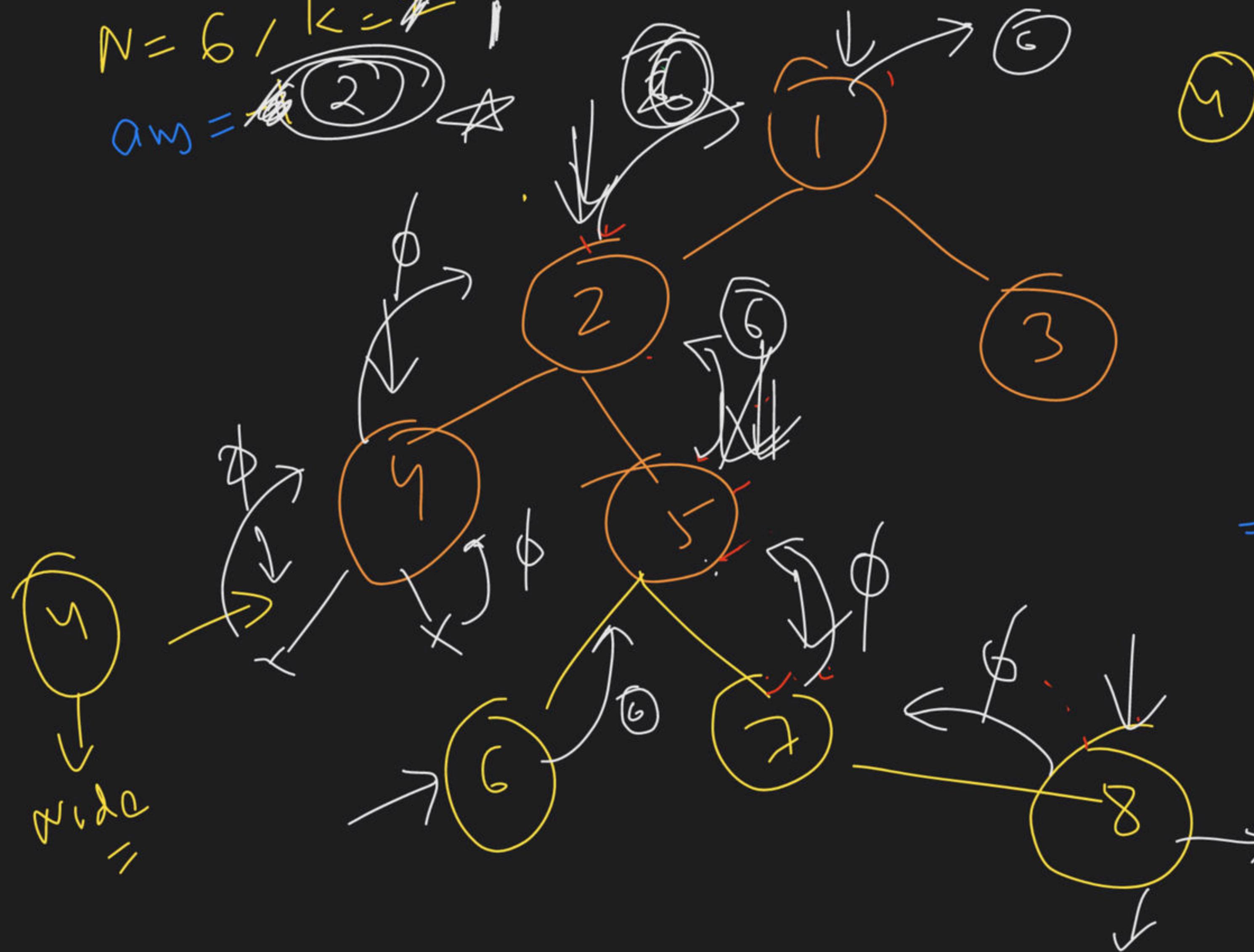


$\Rightarrow K^{th}$ Ancestor of a Node in Binary Tree



$N = 6, k = 2$

$a_{w_2} = 2$



$N = 8, k = 4$

$\Rightarrow \text{find the node}$

$k = 4$



```
if(!root) return 0;
if(root->data == node) return root;

if(ans != -1){
    // answer mil chuka hai, toh aage mt jao
    return root;
}

Node* l = solve(root->left, k, node, ans);
auto r = solve(root->right, k, node, ans);

// node yaha hogi
if(!l && !r) return 0;

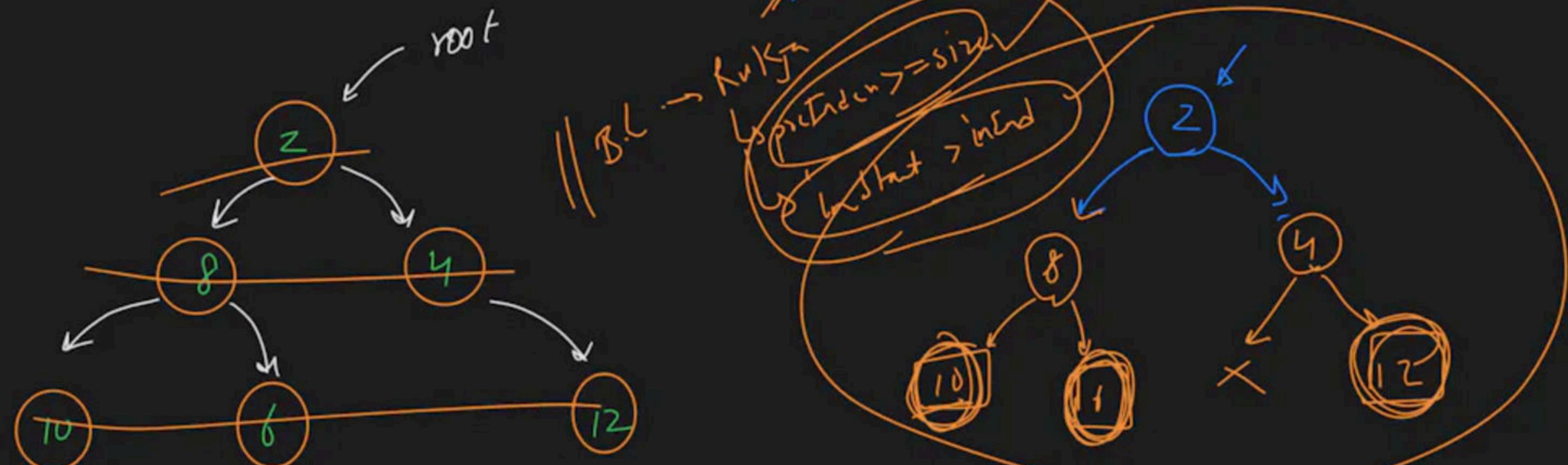
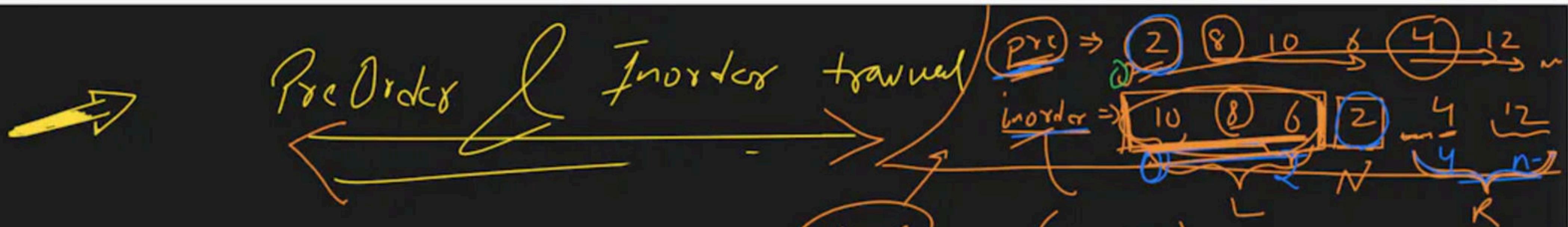
k--;
if(k == 0){
    ans = root->data;
    return root;
}

if(l && !r) return l;
else if(!l && r) return r;
else return root.
```

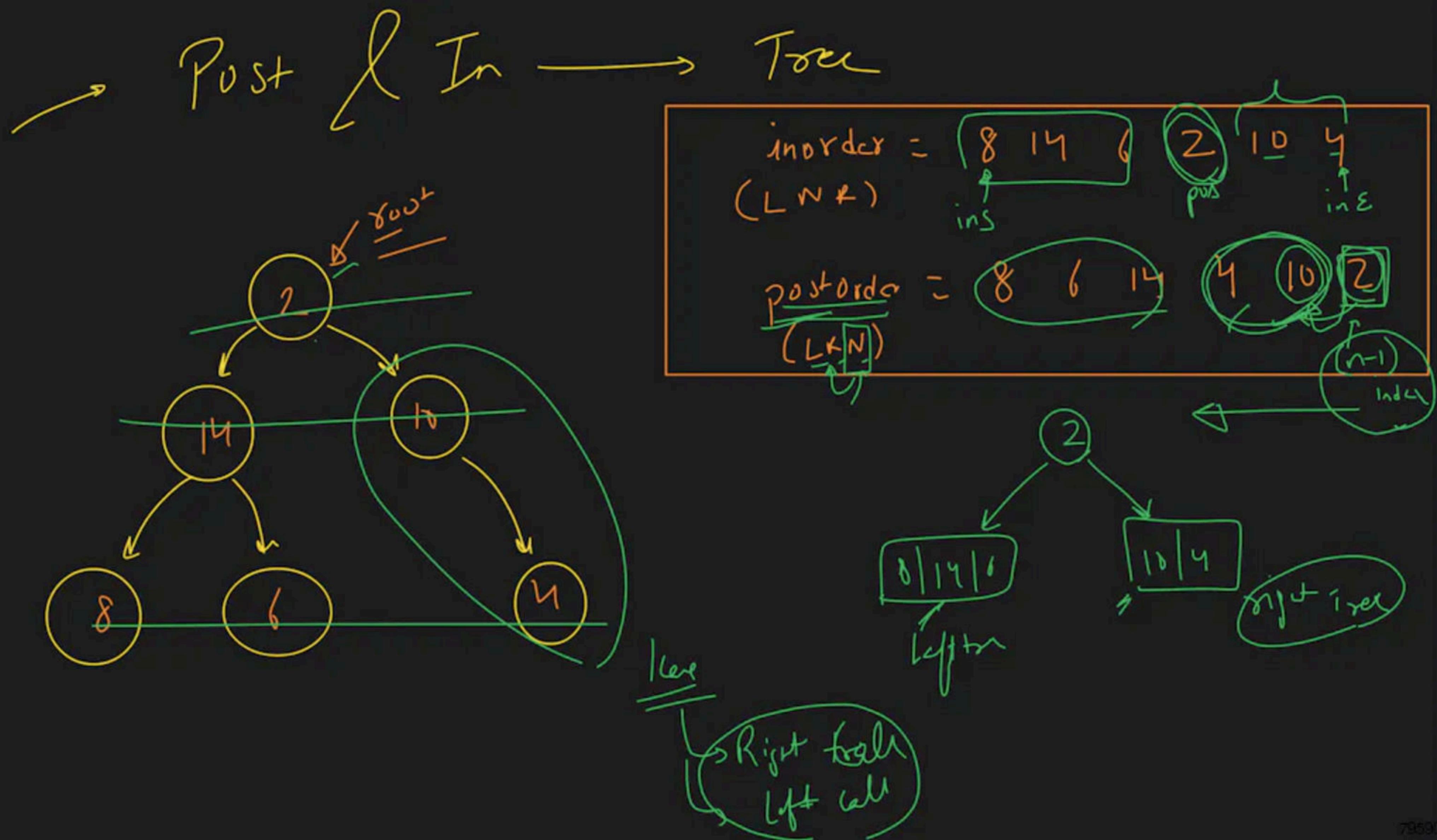
$l \rightarrow l$

$l \rightarrow l$

$l \rightarrow l$



solve (preOrder, preOrderIndex, inorder, inorderStart, inorderEnd, size)



indexer →

10 8 6 2 4 12
0 1 2 3 5 11

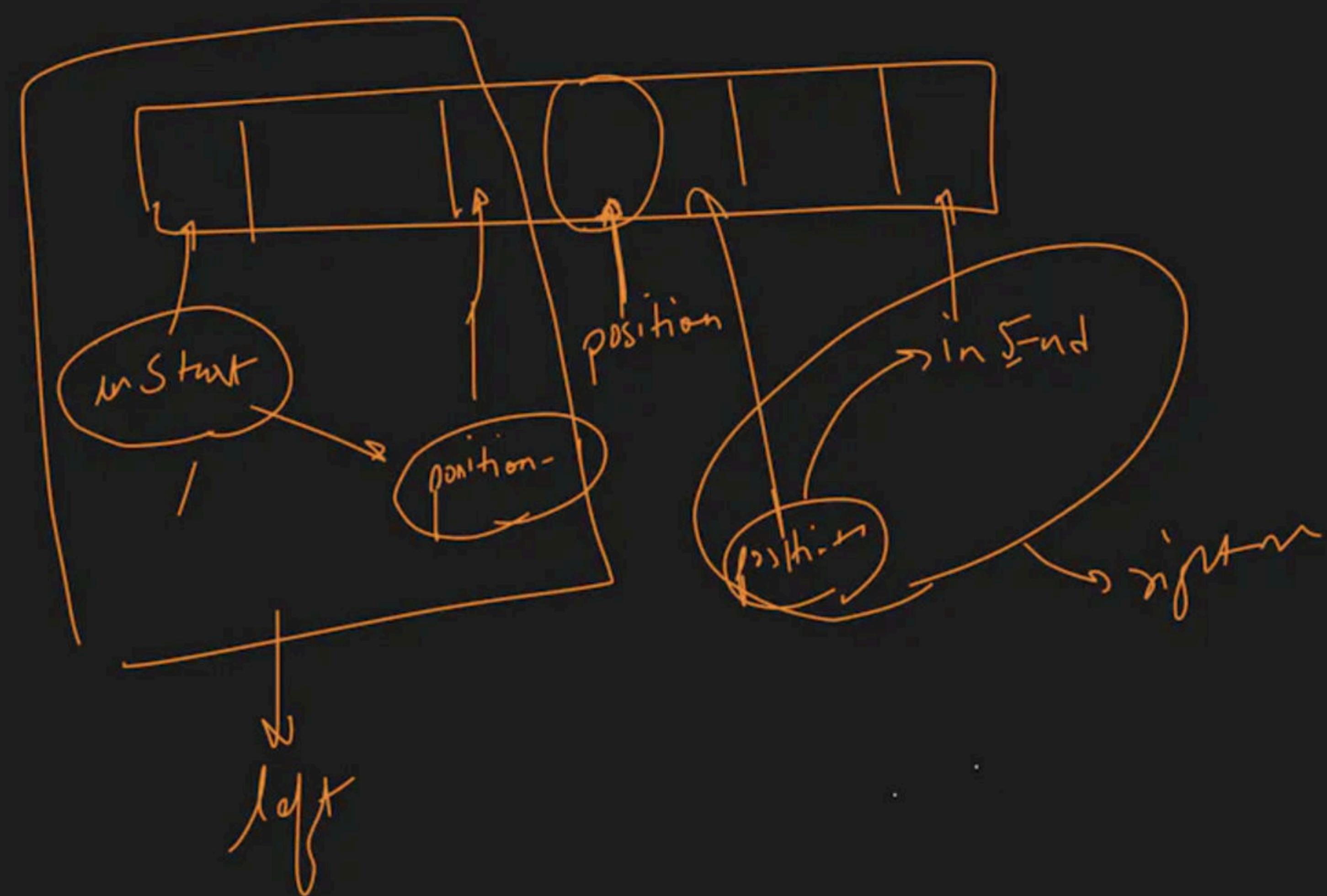
valueToIndexMap [6] → 2

value	Index
10	0
8	1
6	2
2	3
4	4
12	5

map <int, int> valueToIndexMap;

m [10] = 0

for (i = 0 < n'





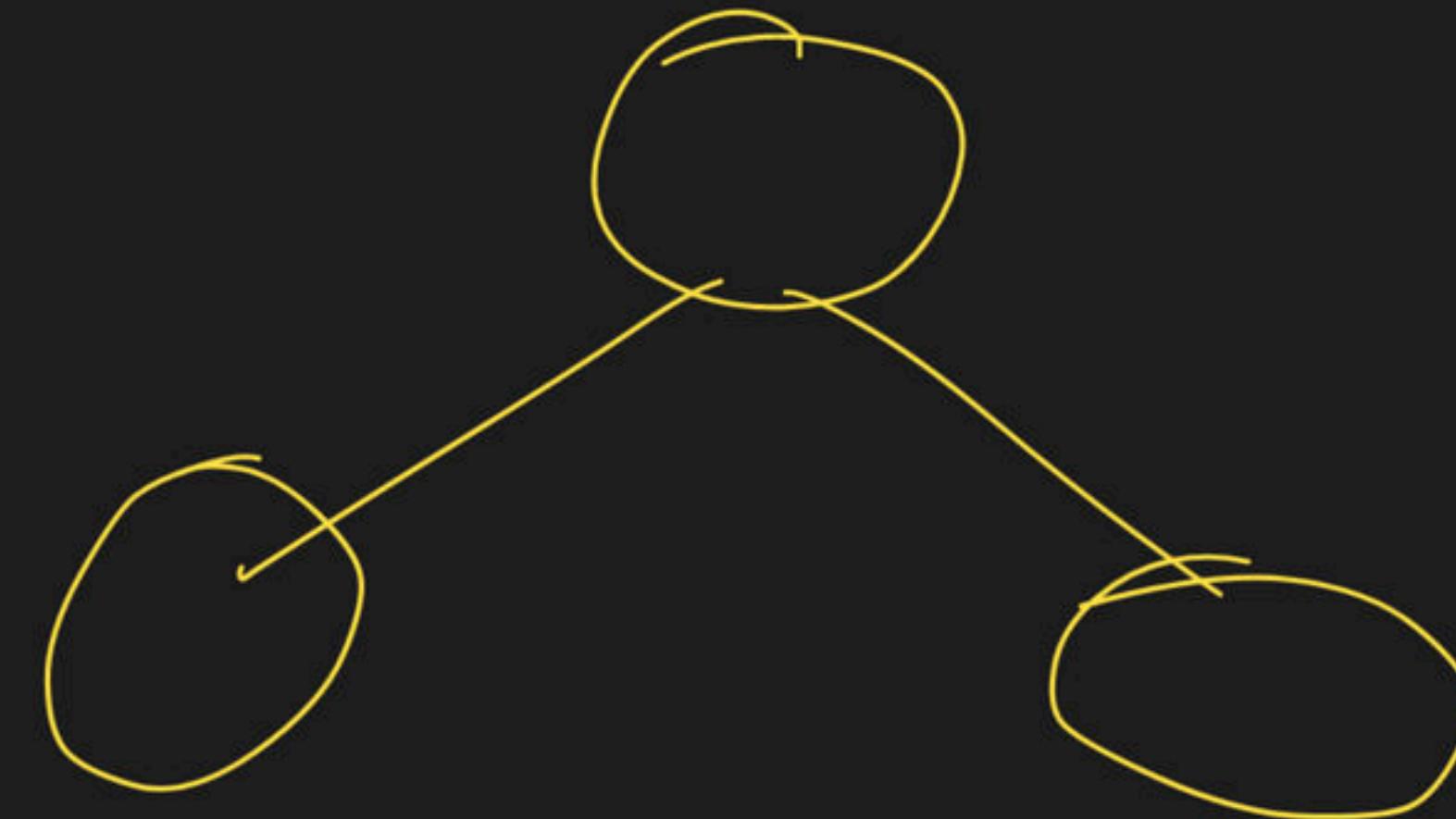
\Rightarrow hemic Tree

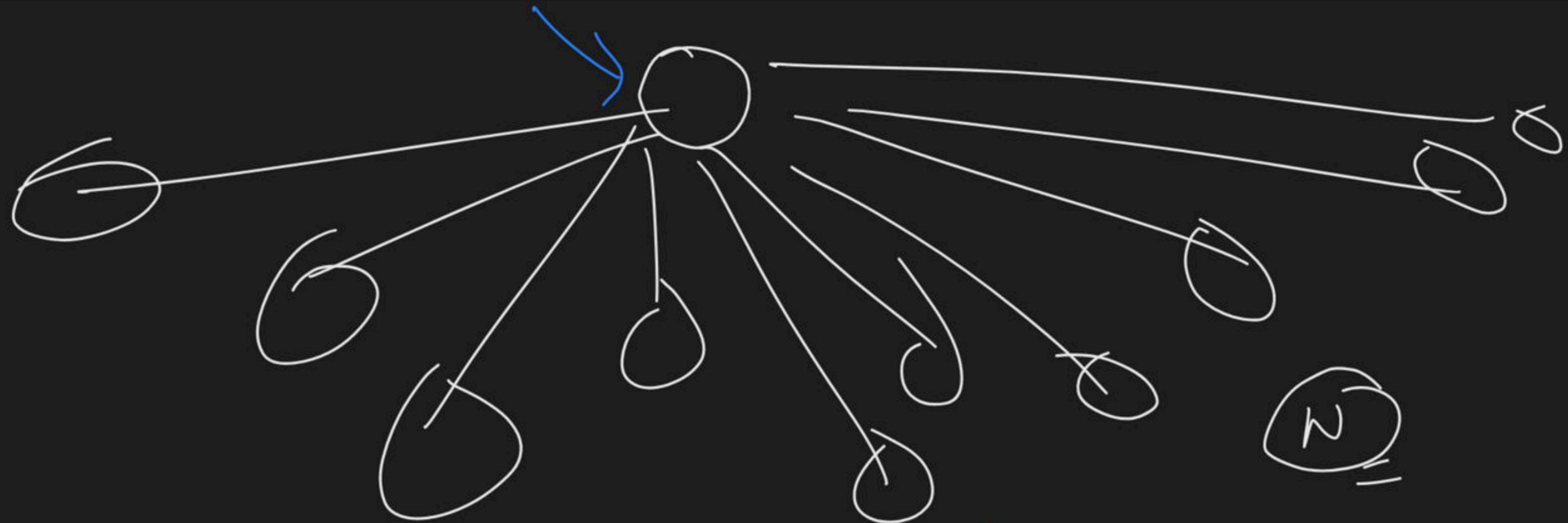
\Rightarrow Node

{
 int data;

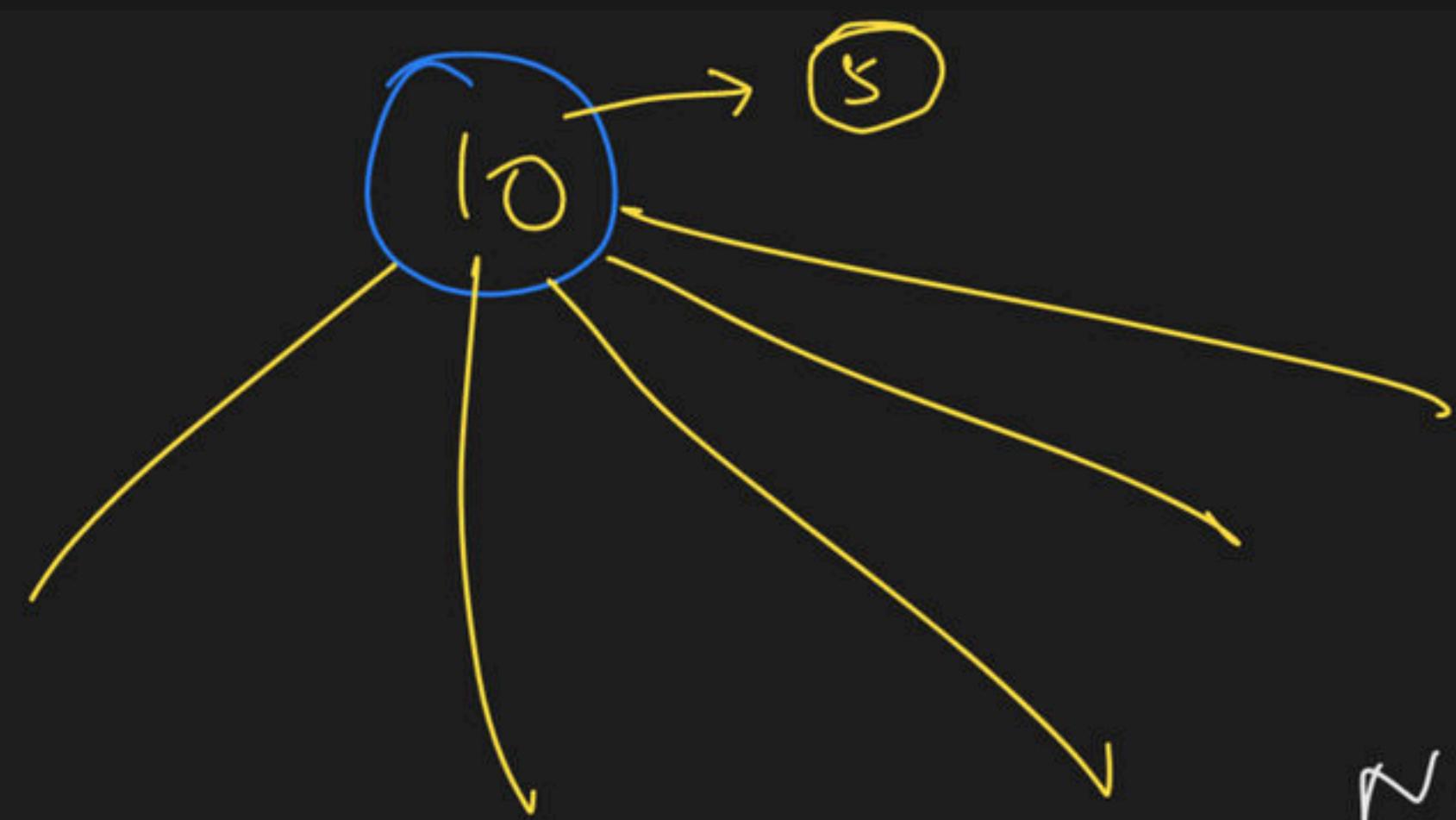
 Node * left;

 Node * right;





```
class Node {  
    =>     int data;  
            int children_count;  
            Node **children;  
        };
```

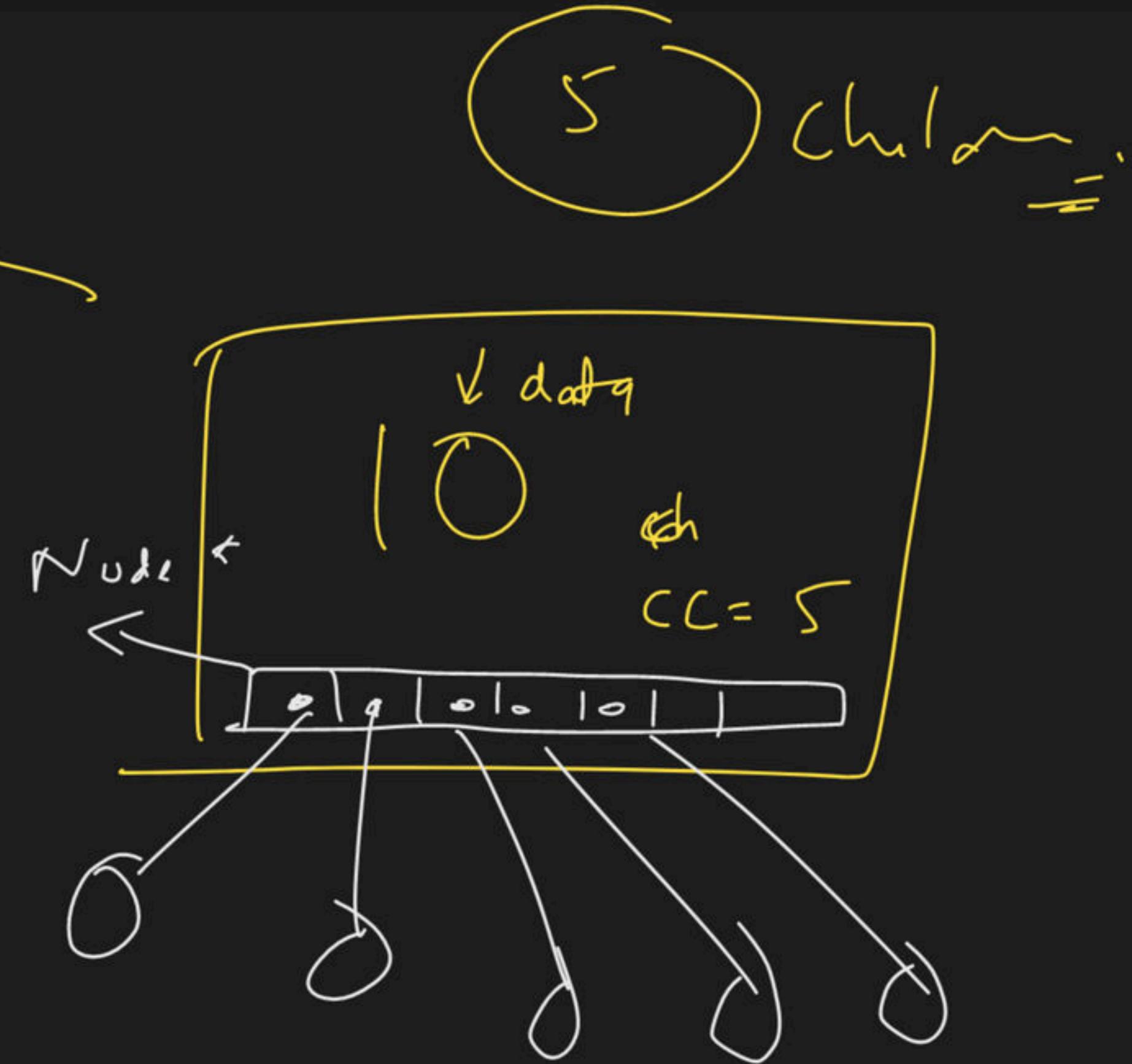


\Rightarrow Level order input

\Rightarrow sum

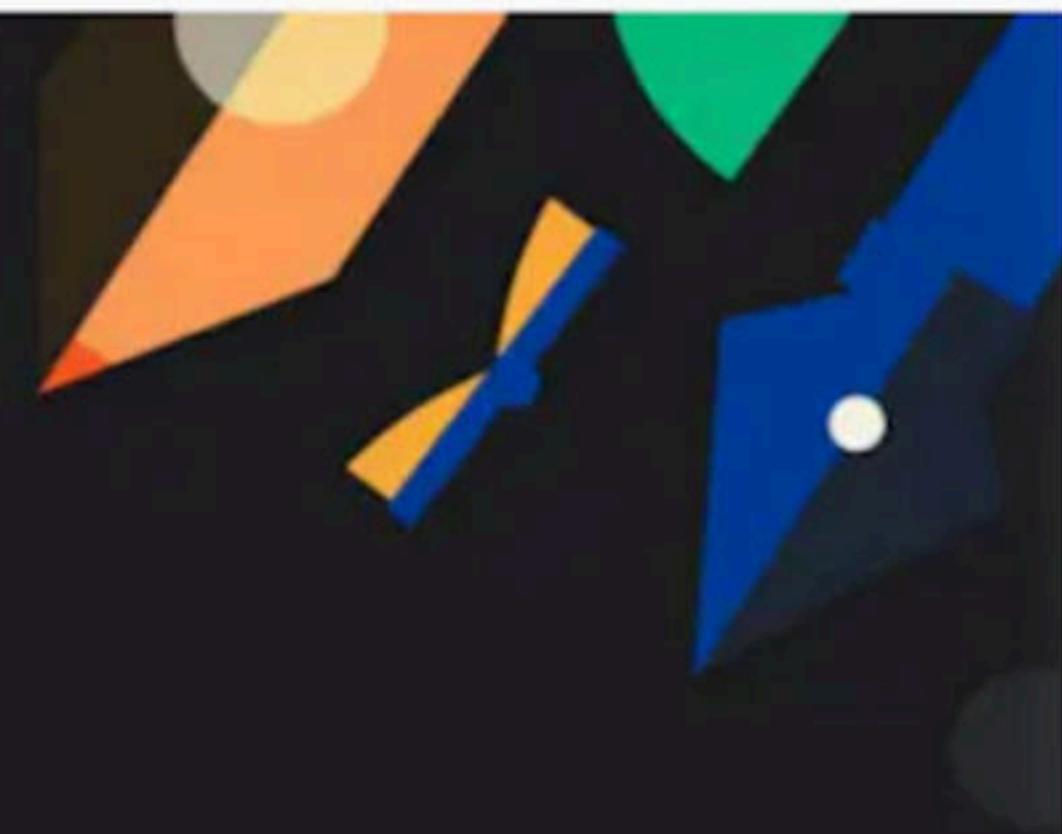
\Rightarrow print at k^{th} level

\Rightarrow height



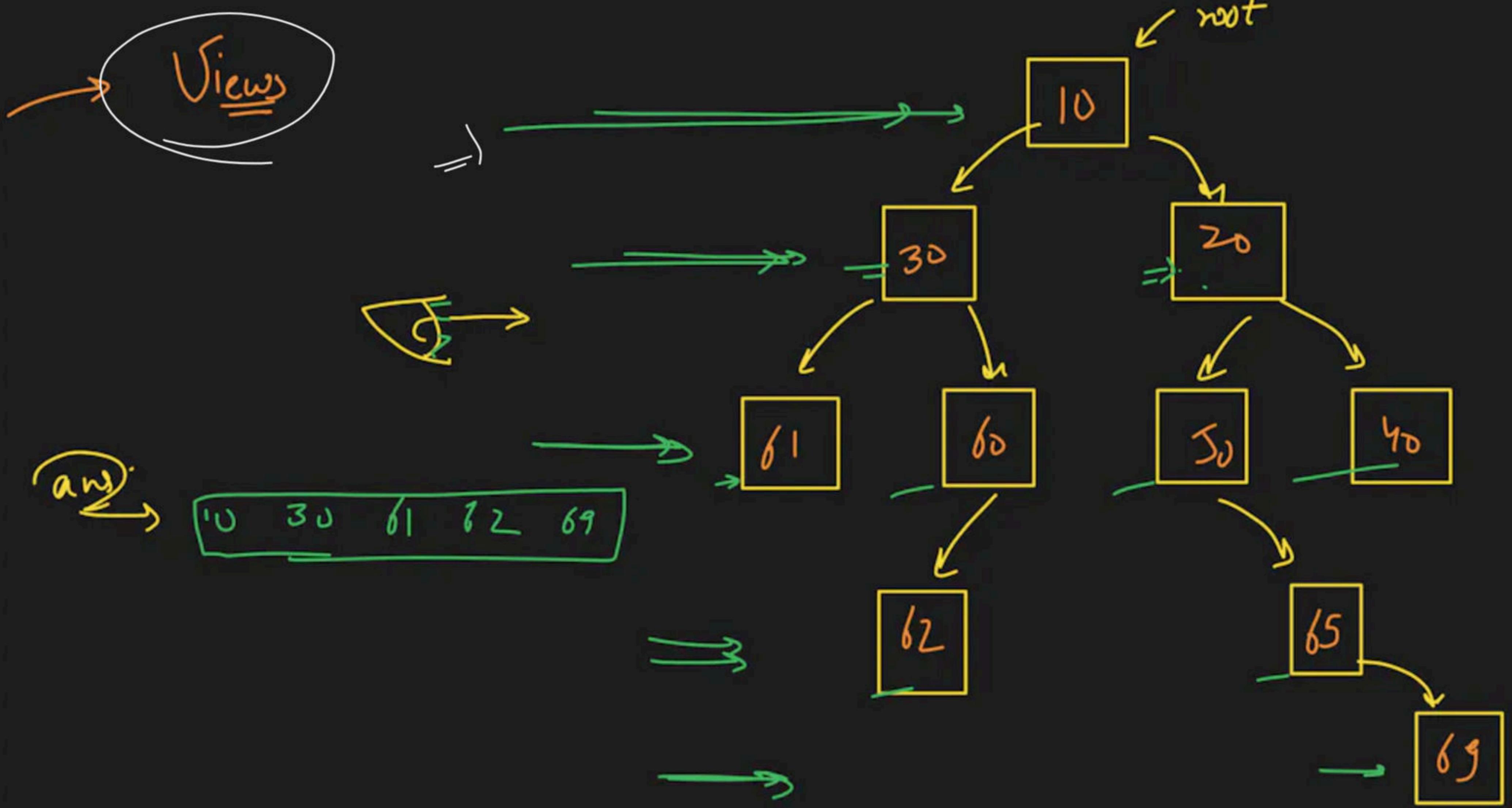
\Rightarrow find

\Rightarrow Path sum



Binary Trees Class - 3

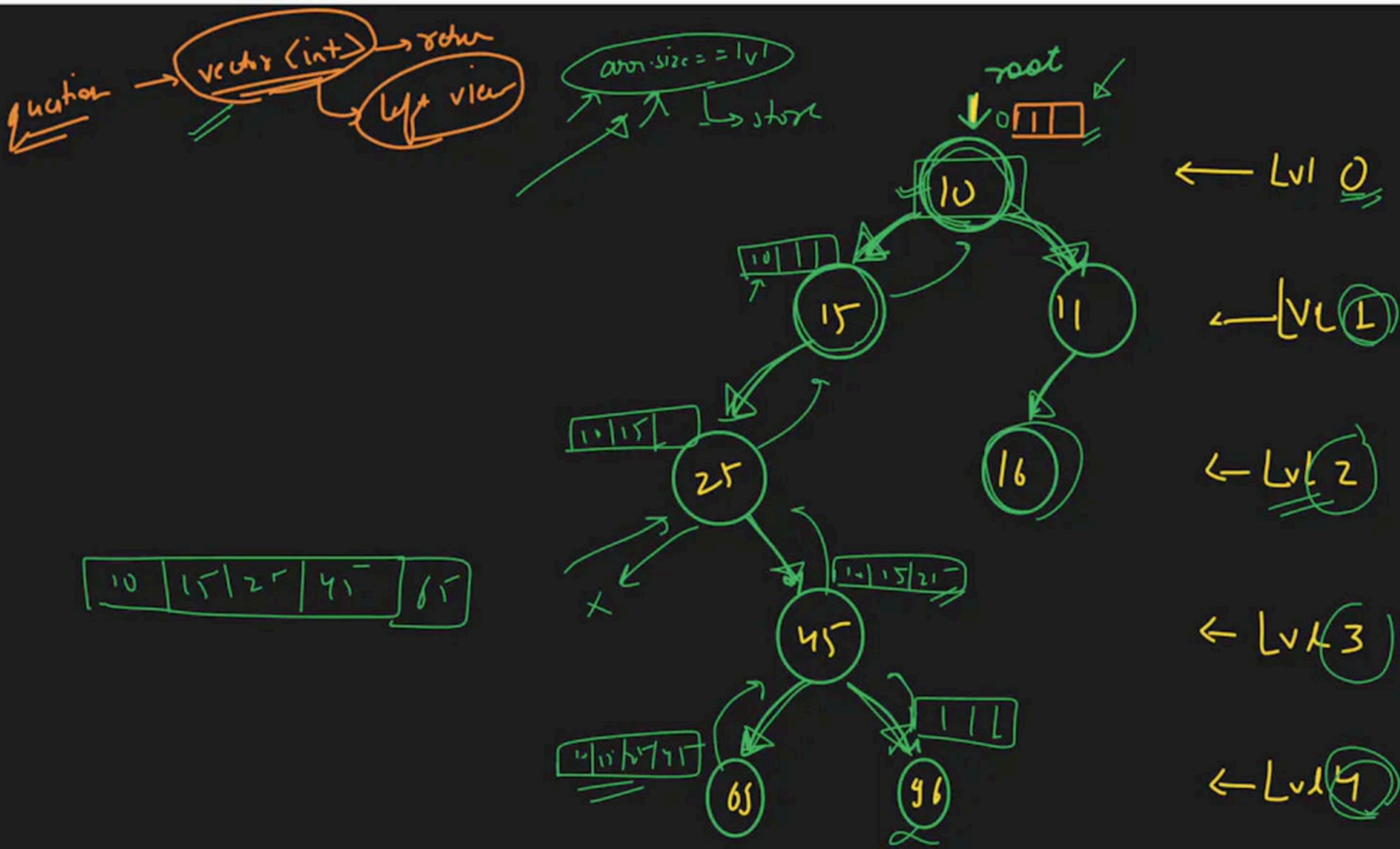
Special class

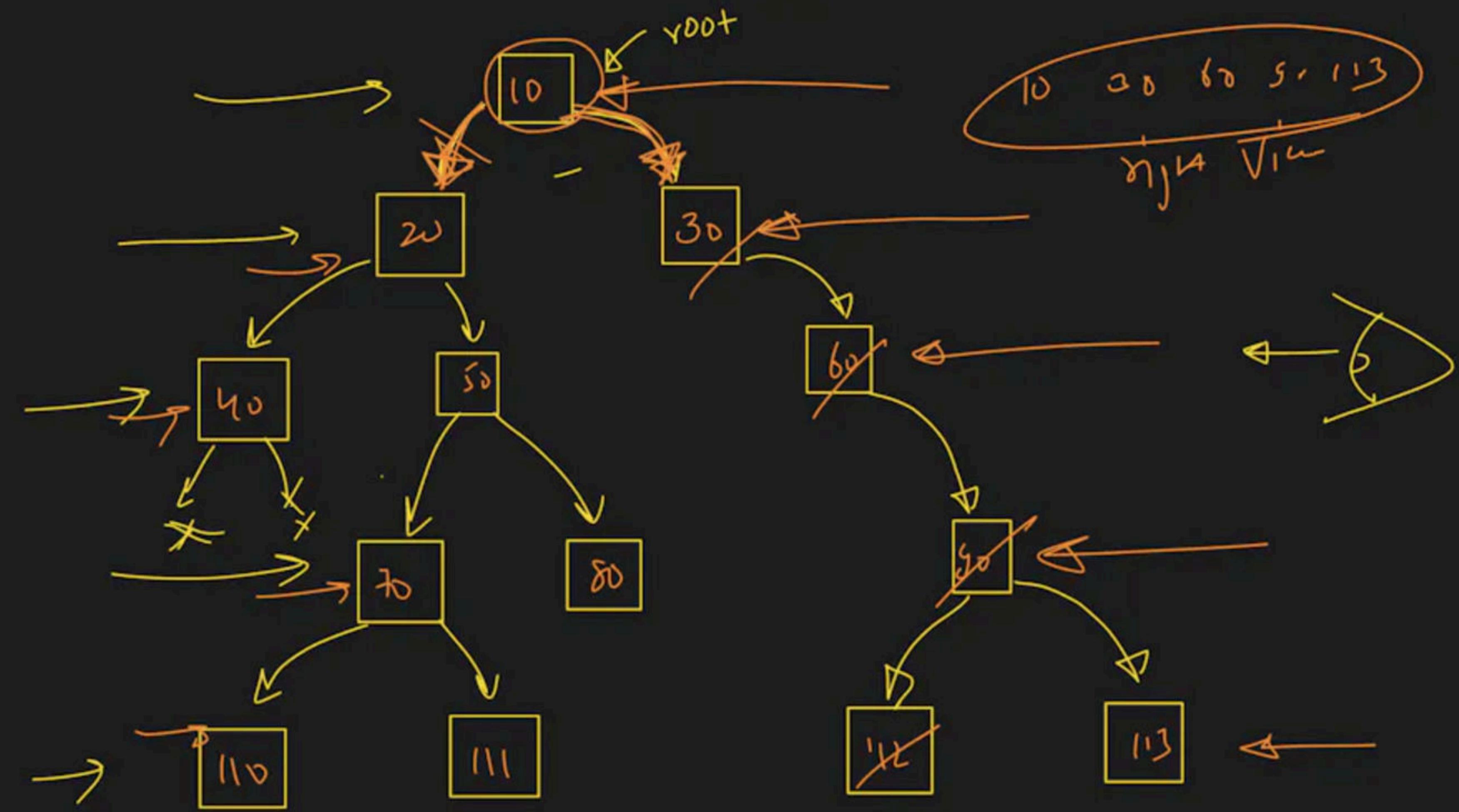


han lard ka kind clunut) → MWS

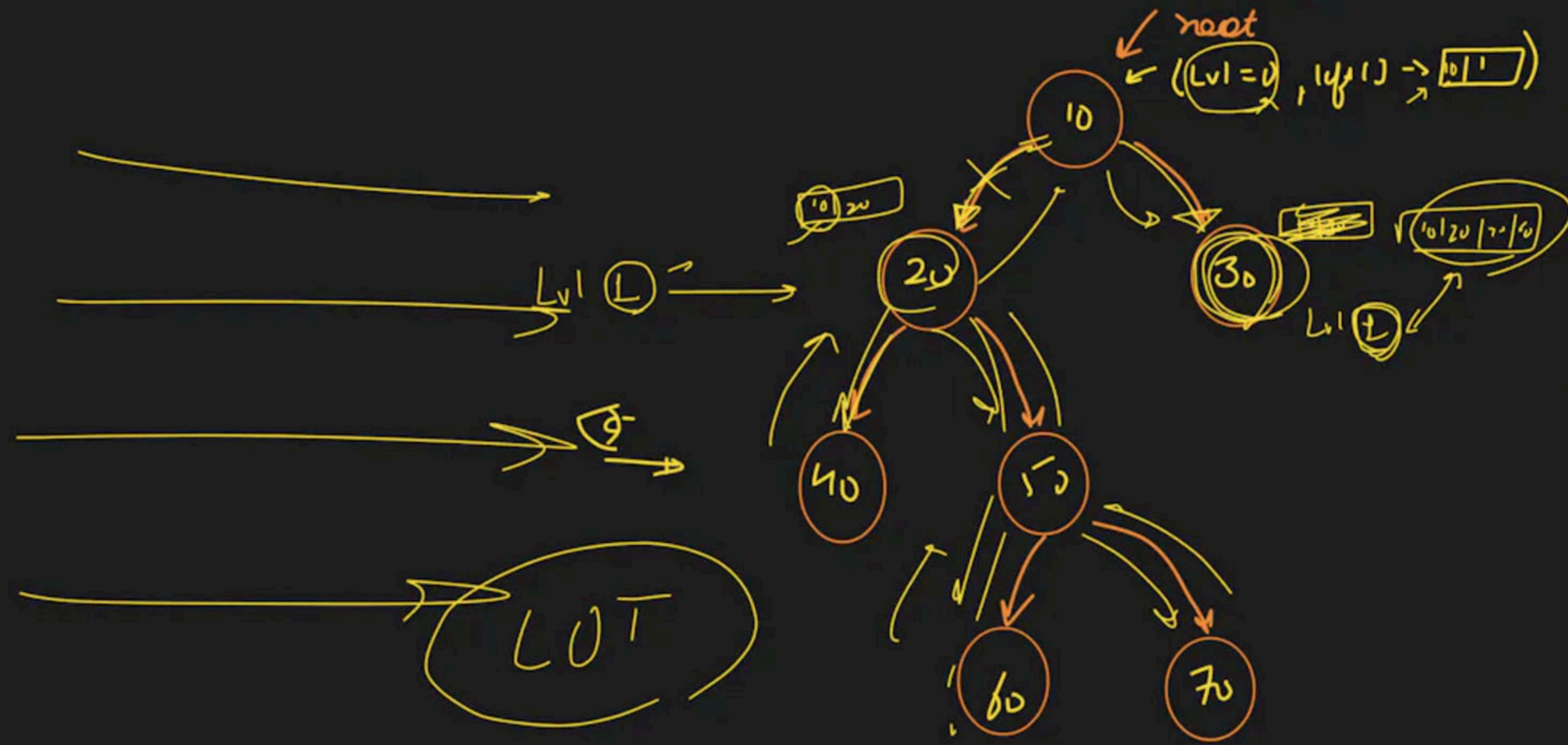
RLC







10 20 40 70 110

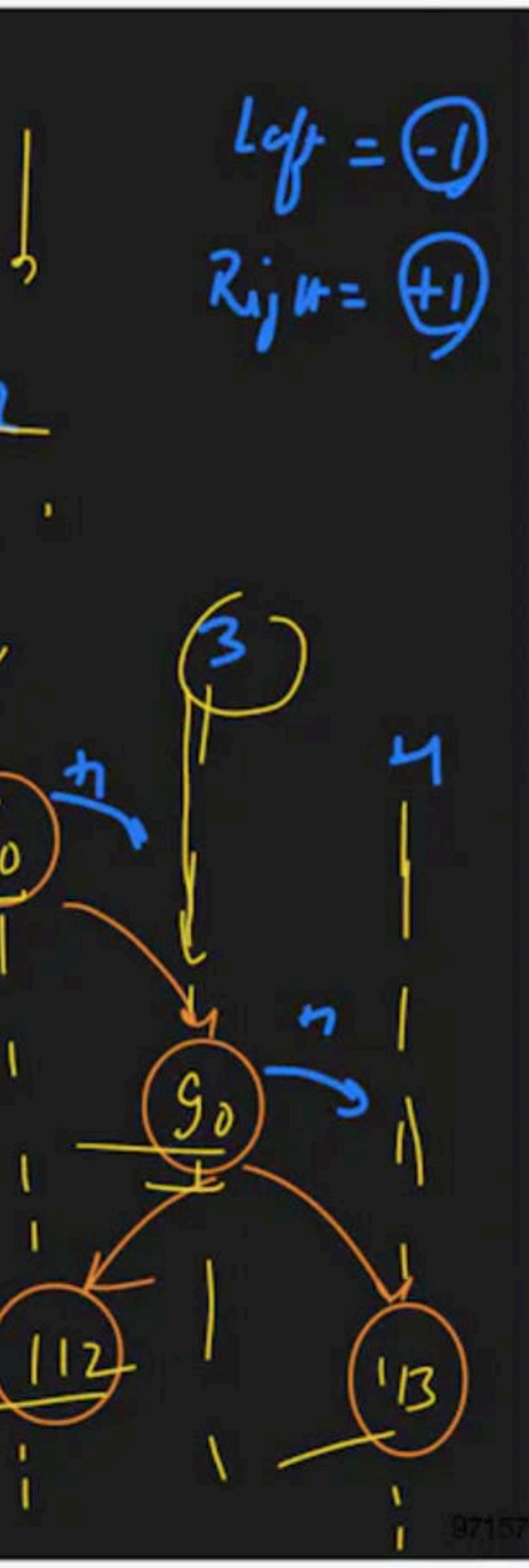


$$Left = -1$$

$$Right = +1$$



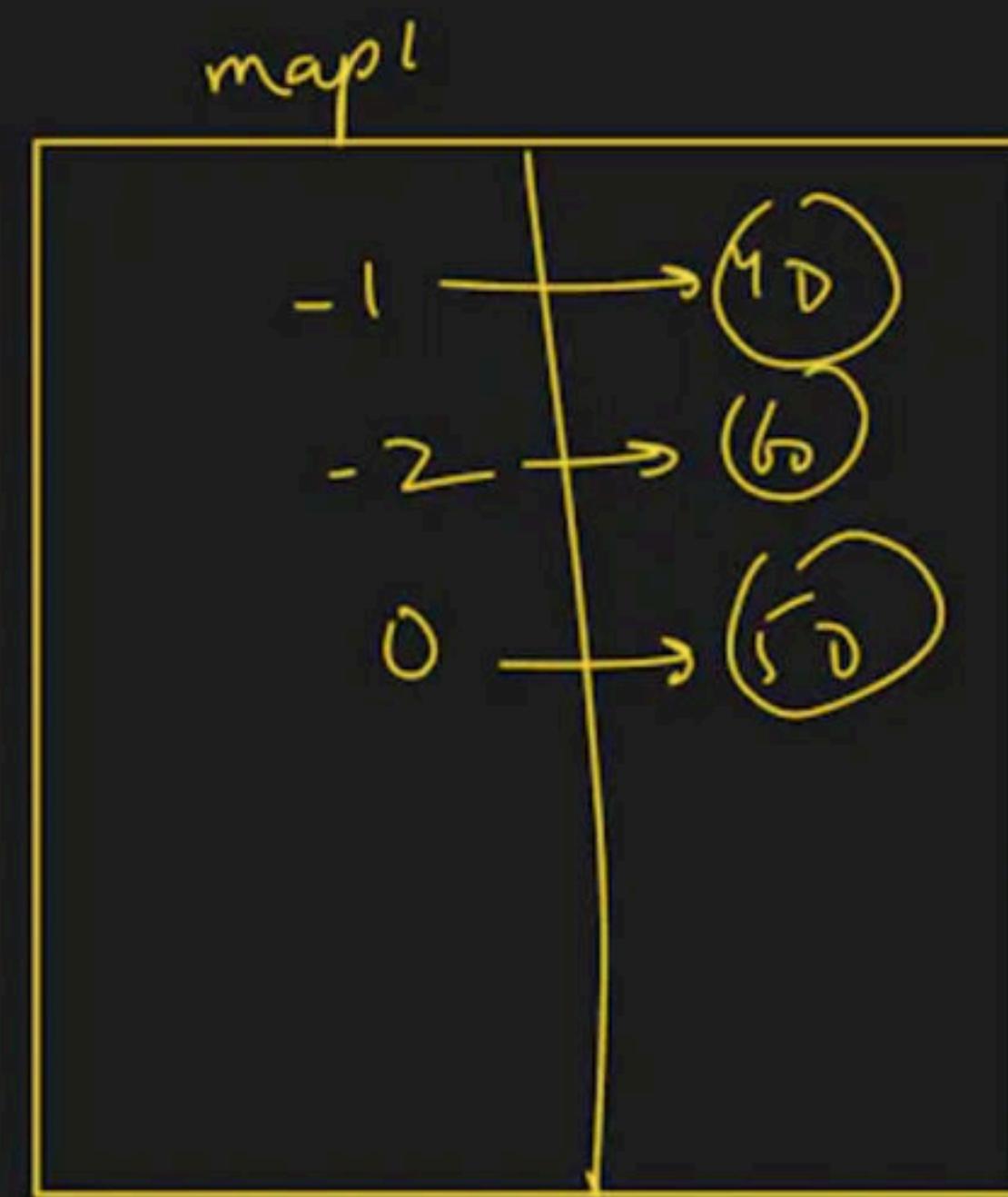
Top View



```

for (auto v : map1)
{
    cout << v.second;
}

```



v → pair → <-1, 40>
 pair → <-2, 60>
 pair → <0, 50>

