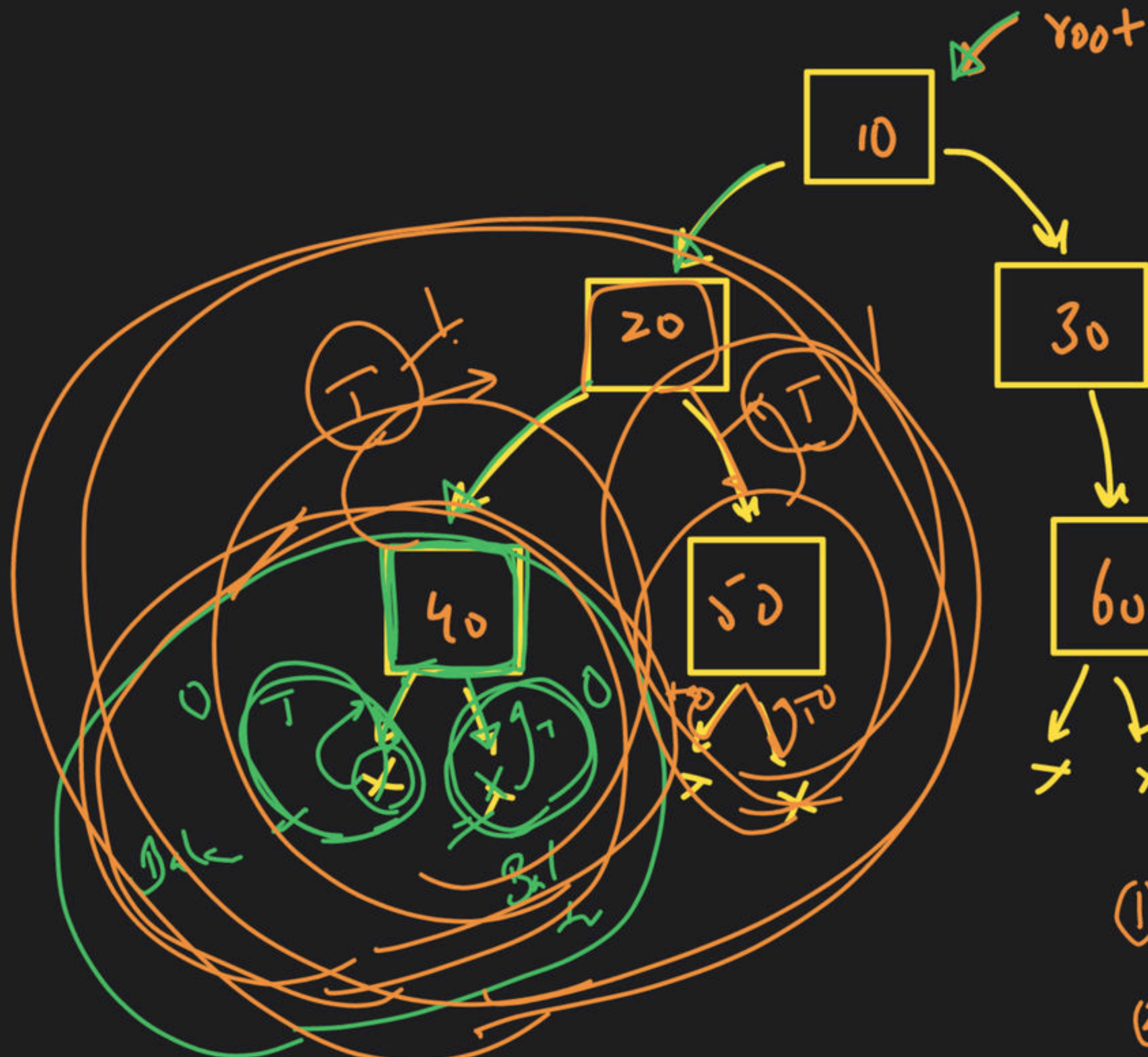




Binary Trees Class - 2

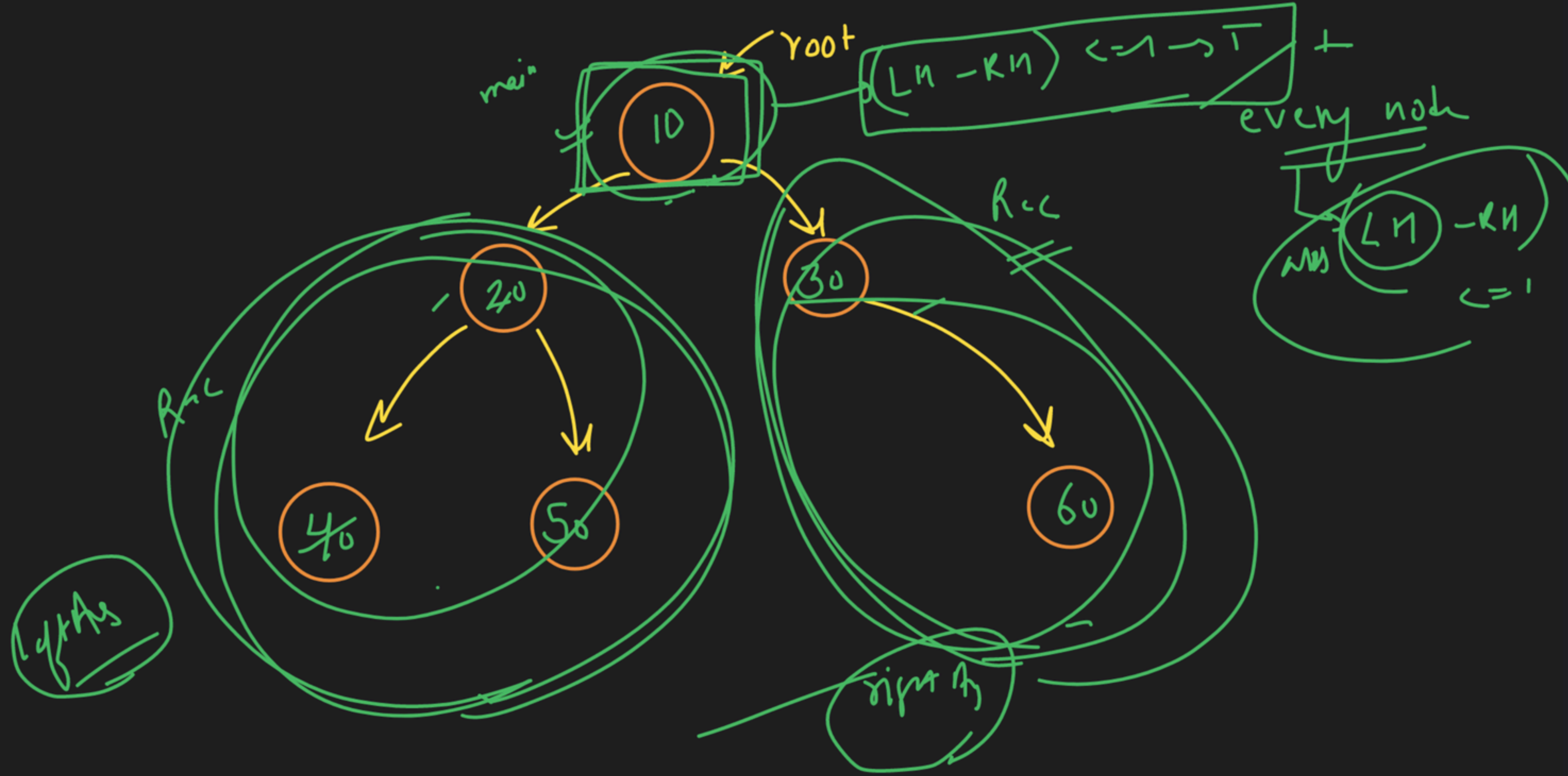
Special class

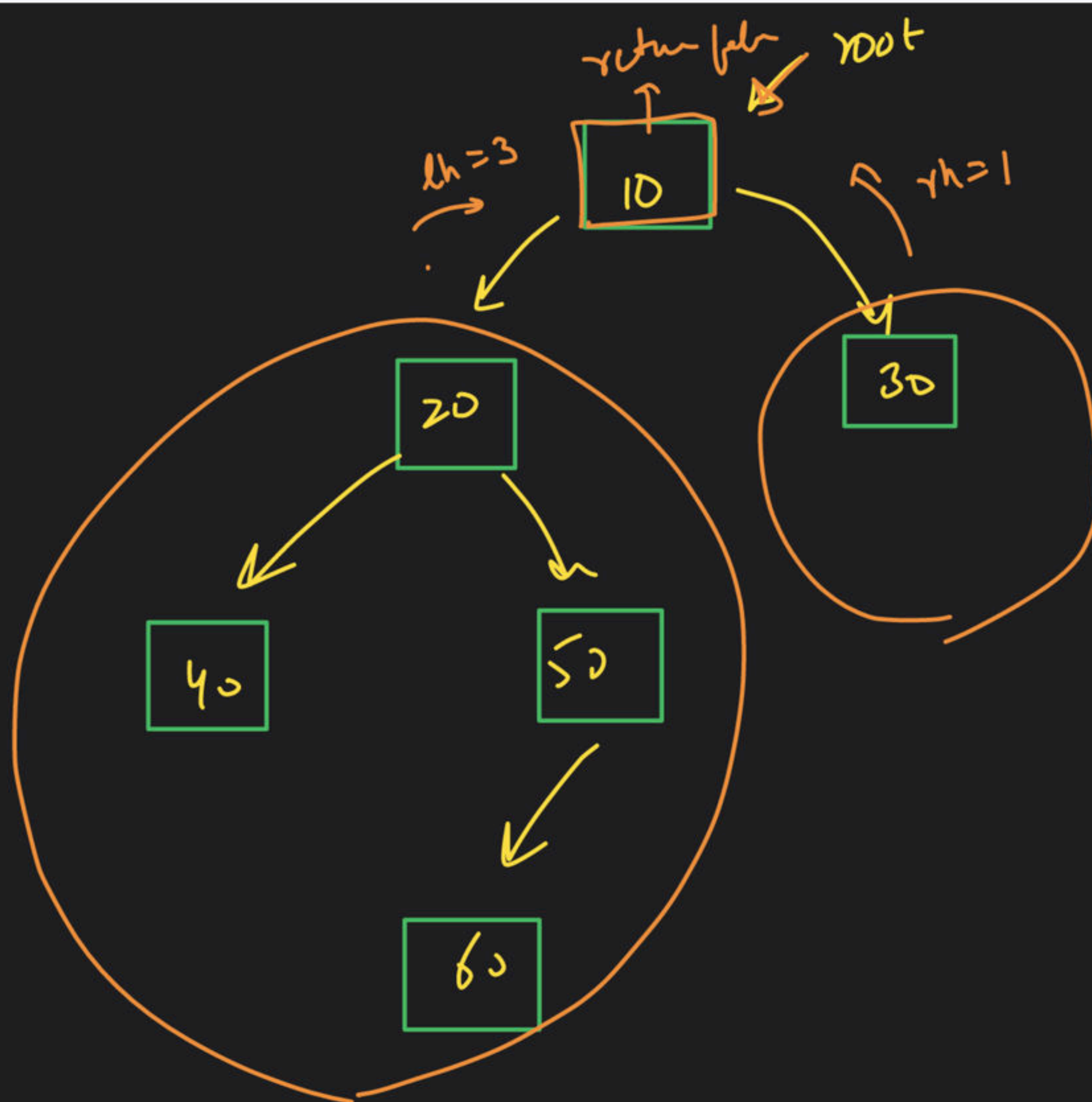


B.C
 if (root == NULL)
 return true

$0 - 0 = 0 \leq 1$

- (1) Left Subtree \rightarrow Bal
 - (2) Right Subtree \rightarrow Bal
- } \rightarrow Bal
- (3) $abs(LH - RH) \leq 1$





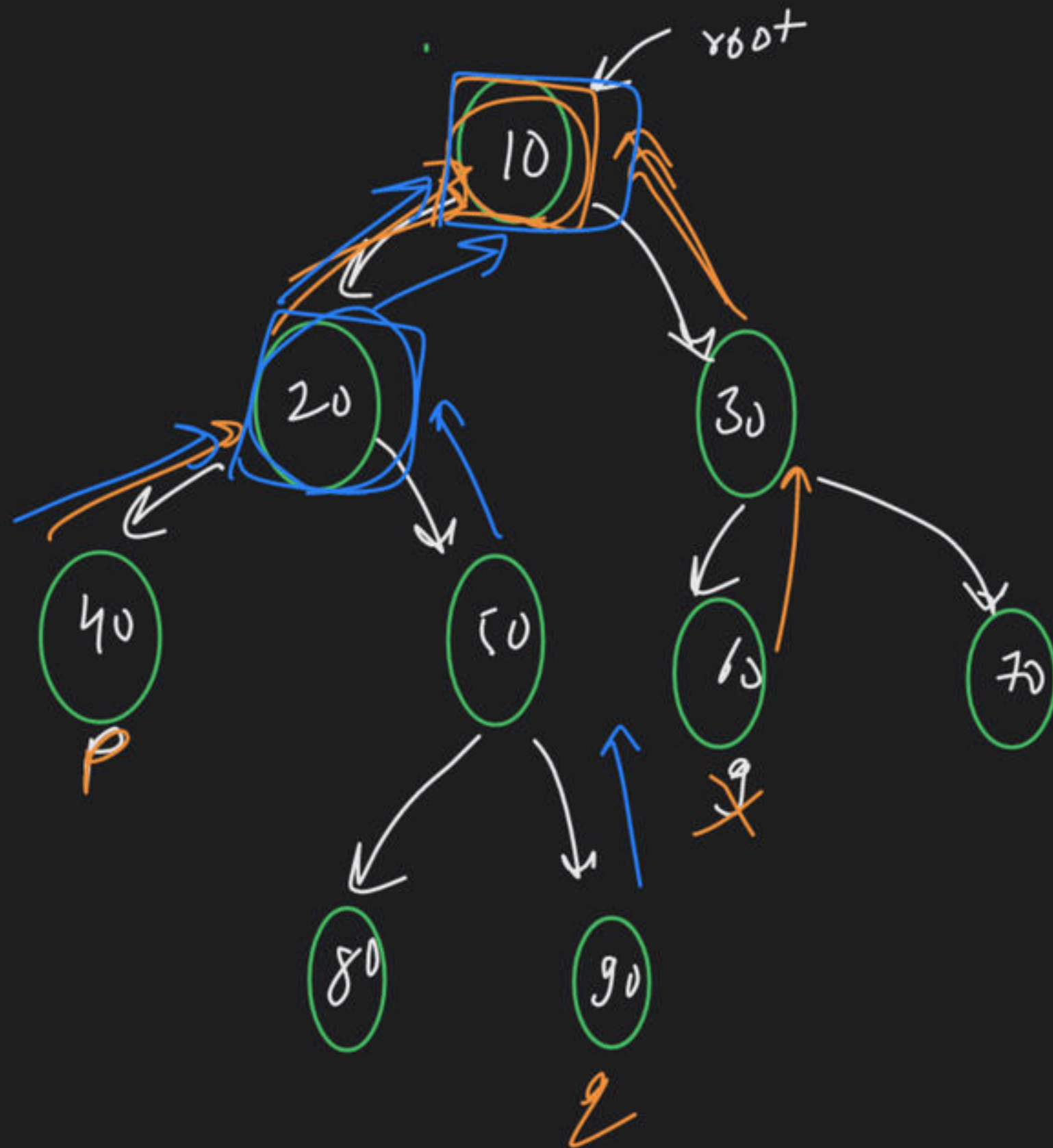
all $(lh - rh) \leq 1$

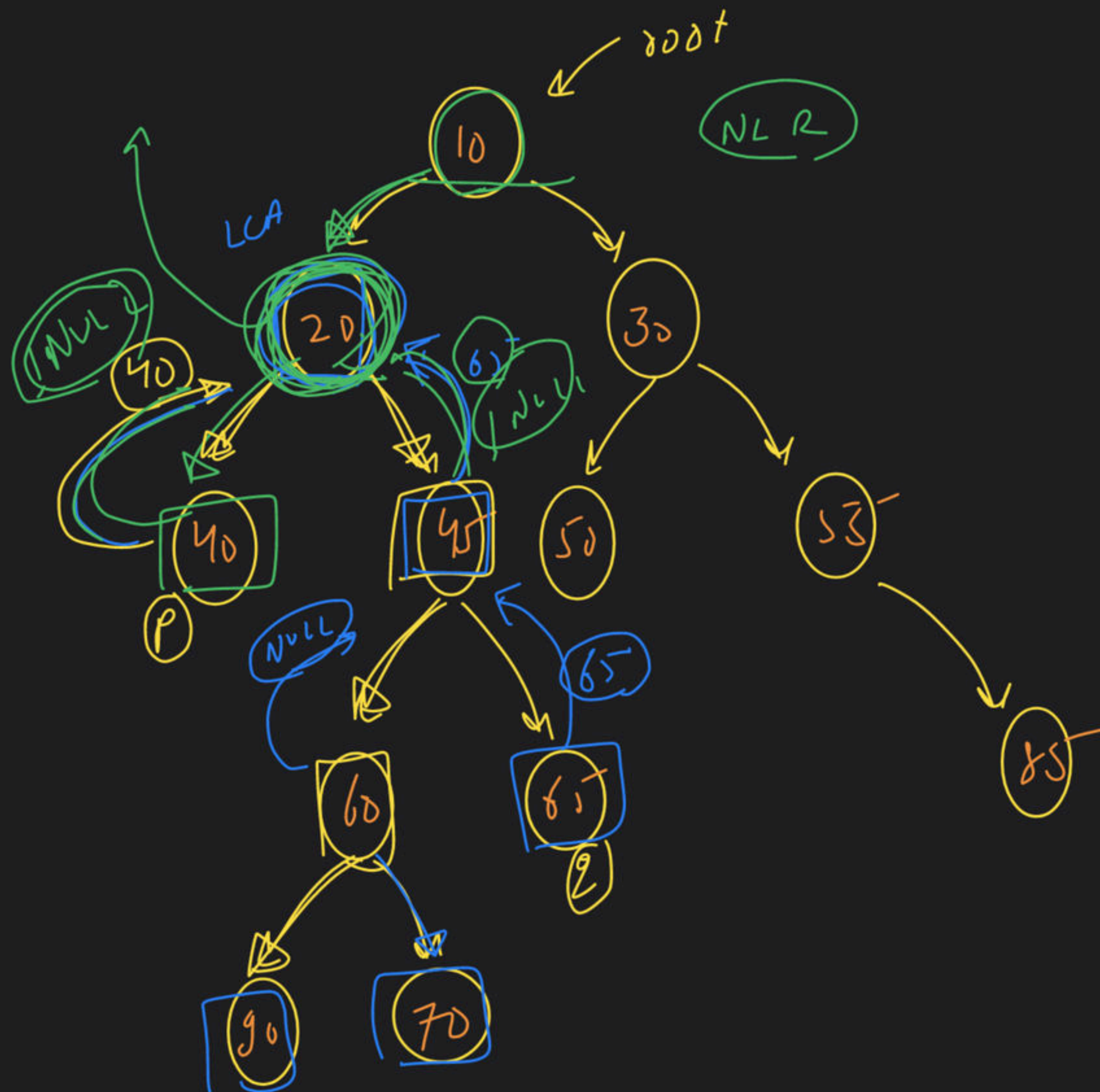
False

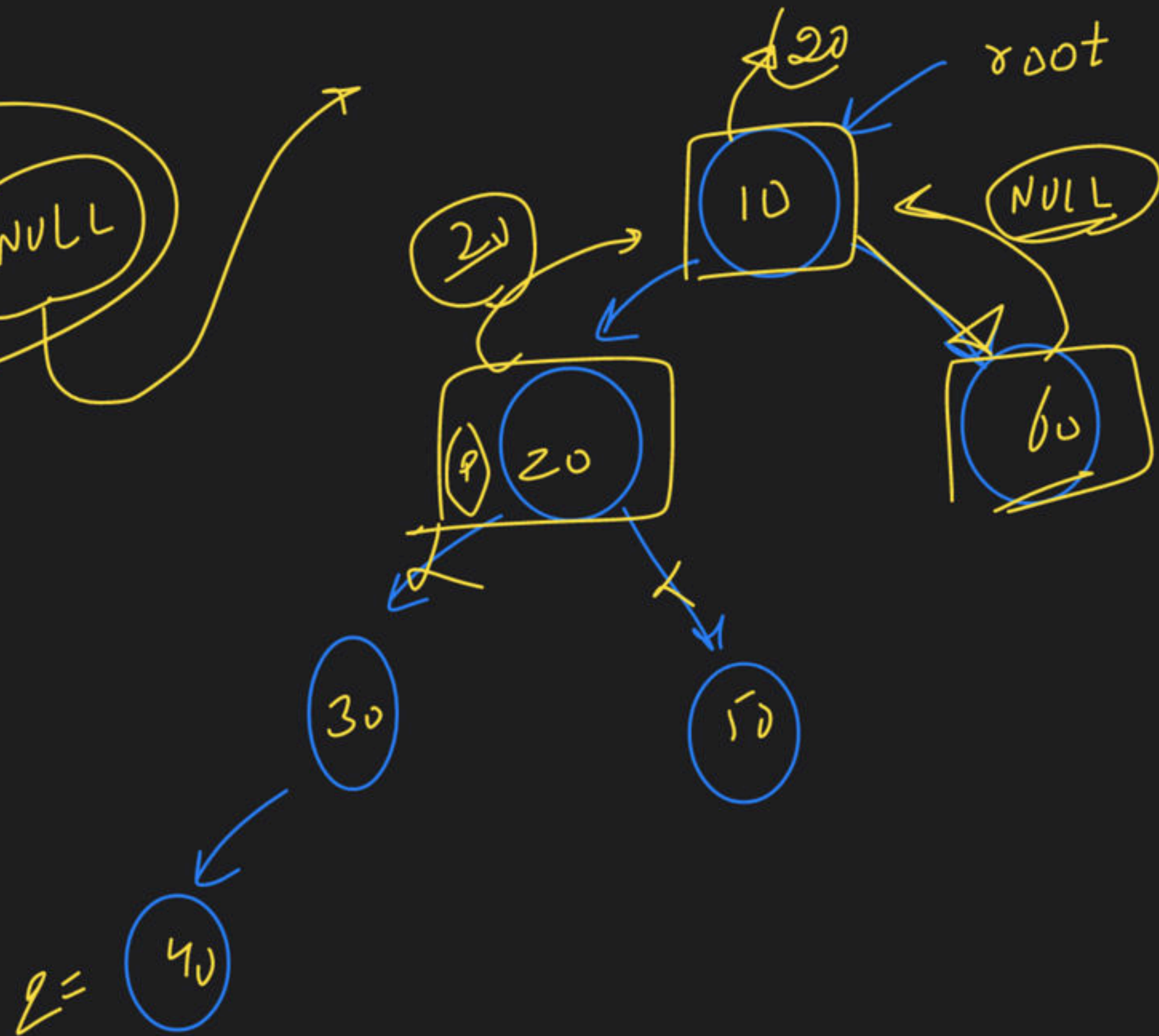
→ L.C → 236 LCA of a Binary tree
Lowest Common Ancestor

LCA → ?

i/p → (p) (q)





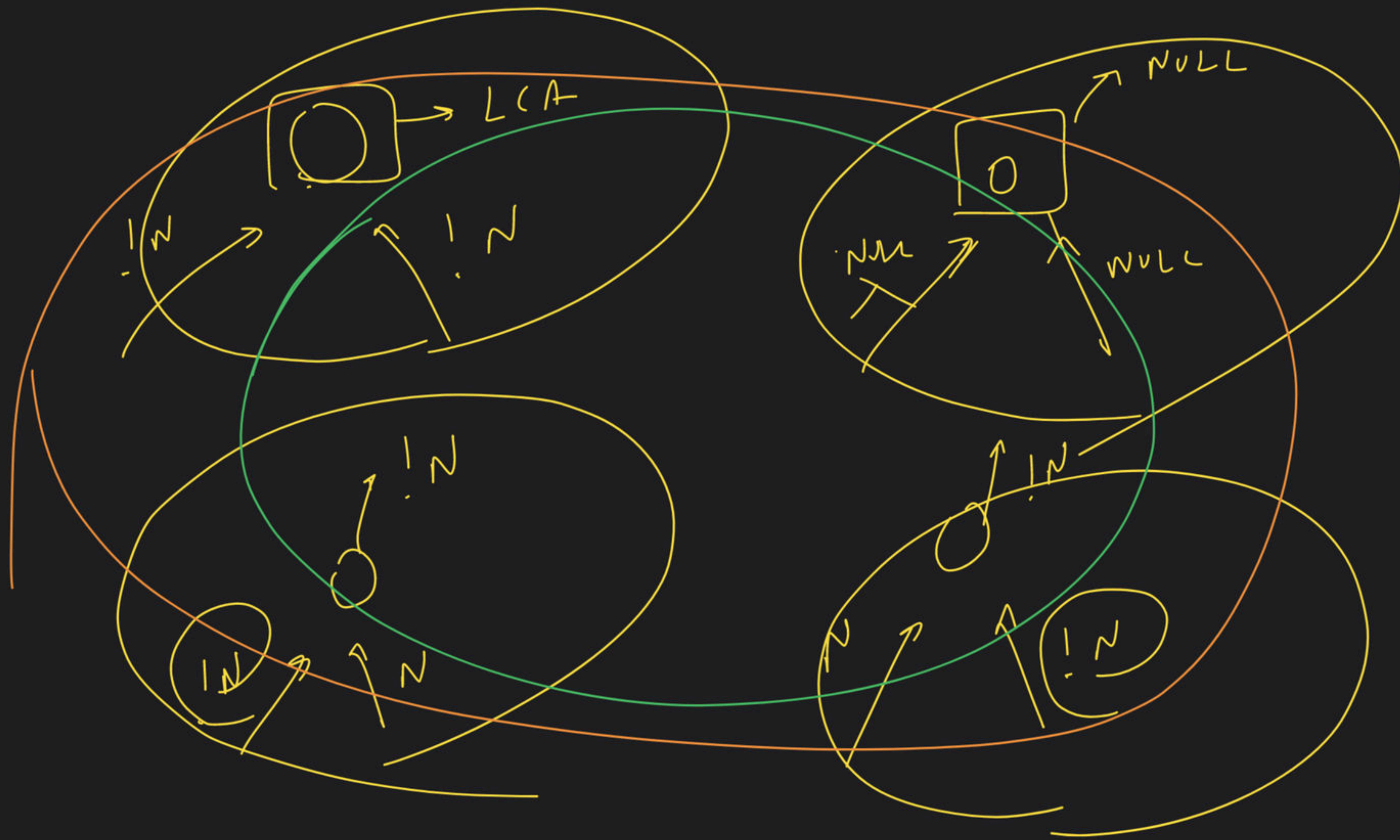


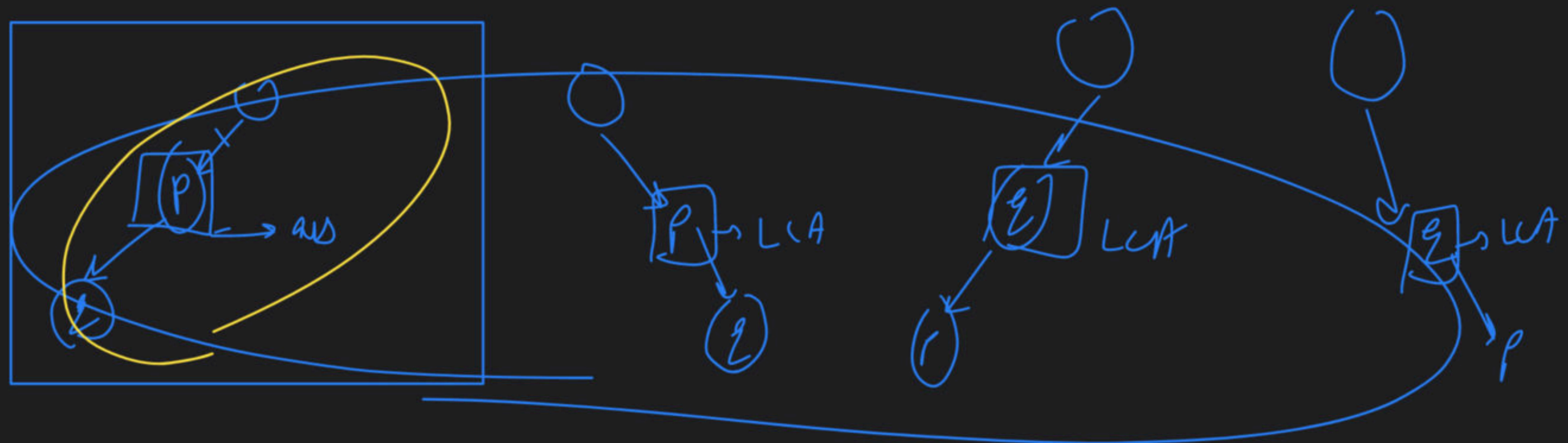
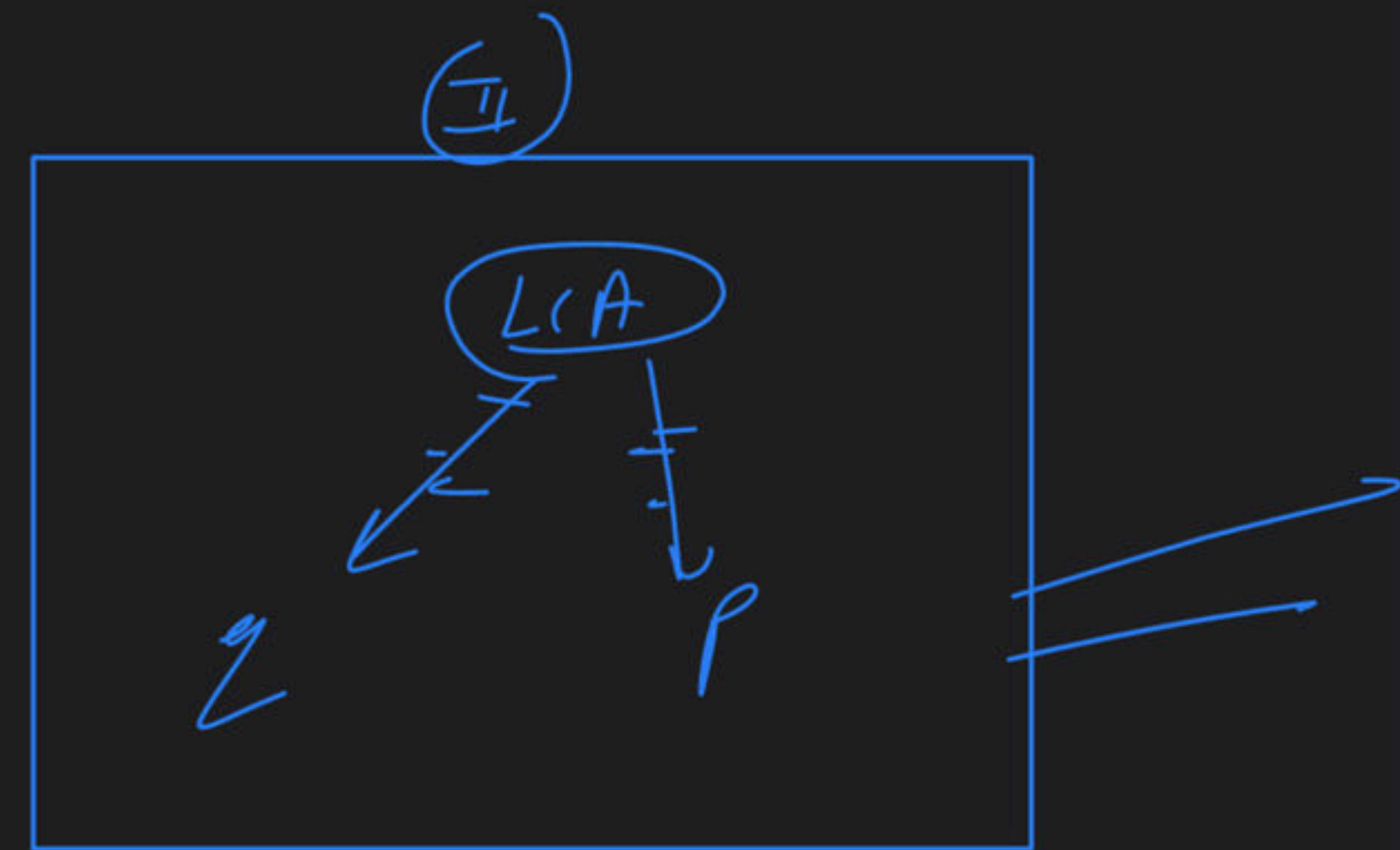
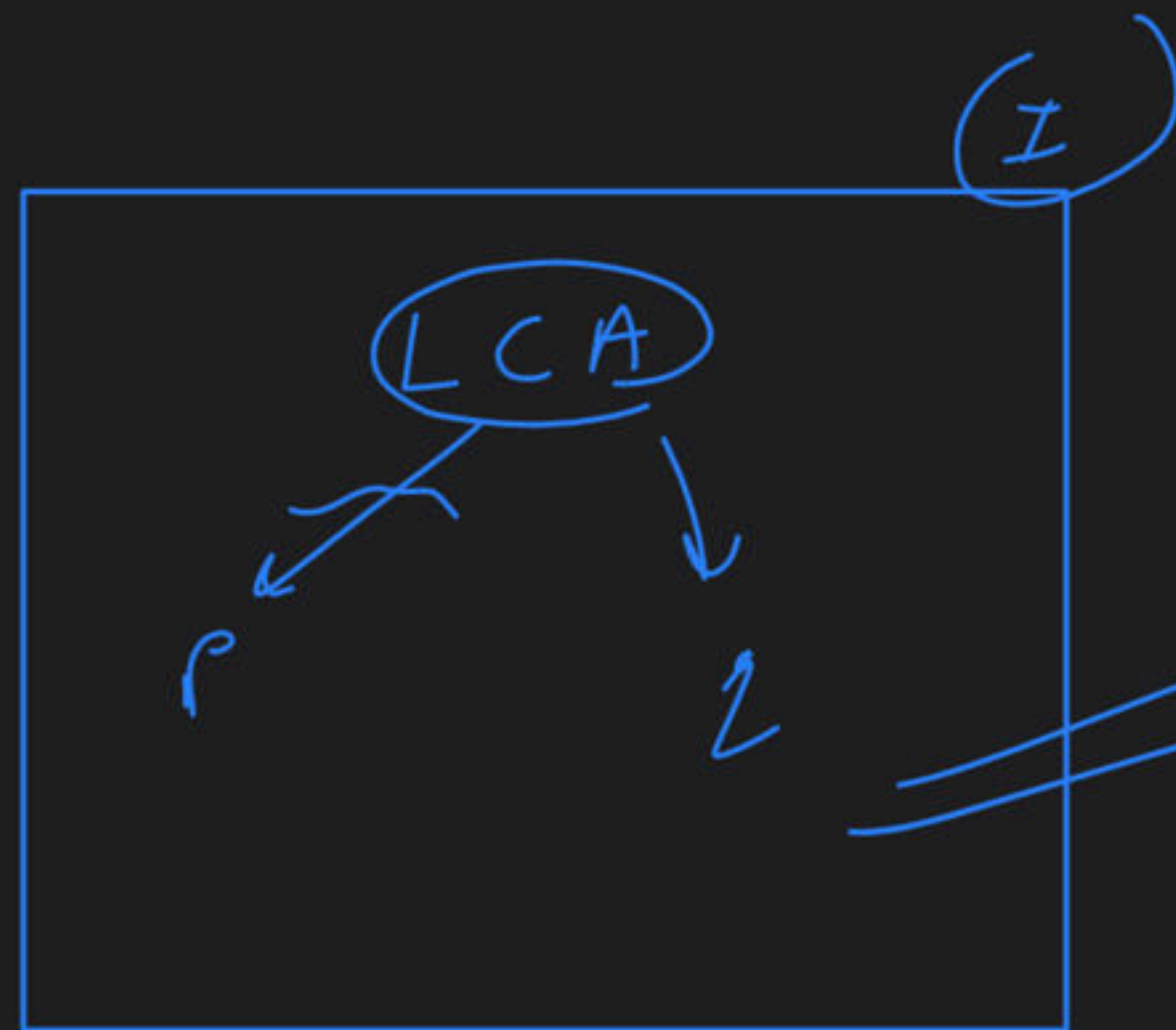
$p = 20$
 $\Sigma = 40$

if (root == NULL)
return NULL

if (root == p)
return p

if (root == Σ)
return Σ





TreeNode* lowestCommonAncestor (root, p, q)

{

if (root == NULL) return NULL;

if (root->val == p) return p;

if (root->val == q) return q;

TreeNode* leftAns = lowestCommonAncestor (root->left, p, q);

TreeNode* rightAns = lowestCommonAncestor (root->right, p, q);

if (leftAns == NULL && rightAns == NULL) return NULL;

else if (leftAns != NULL && rightAns == NULL) return leftAns;

else if (leftAns == NULL && rightAns != NULL) return rightAns;

else return root;

}

2 min
Break

target = 27

locate

sum = 0

root

left || right

R → Left

true

$$① 1 \rightarrow 2 \rightarrow 4 \rightarrow 6 = 13$$

$$② 1 \rightarrow 2 \rightarrow 4 \rightarrow 7 = 14$$

$$③ 1 \rightarrow 3 \rightarrow 5 \rightarrow 20 = 29$$

$$④ 1 \rightarrow 3 \rightarrow 5 \rightarrow 10 \rightarrow 8 = 27$$

OR
left || right

sum = 7

sum = 9

sum = 19

sum = 13

sum = 17

sum = 29

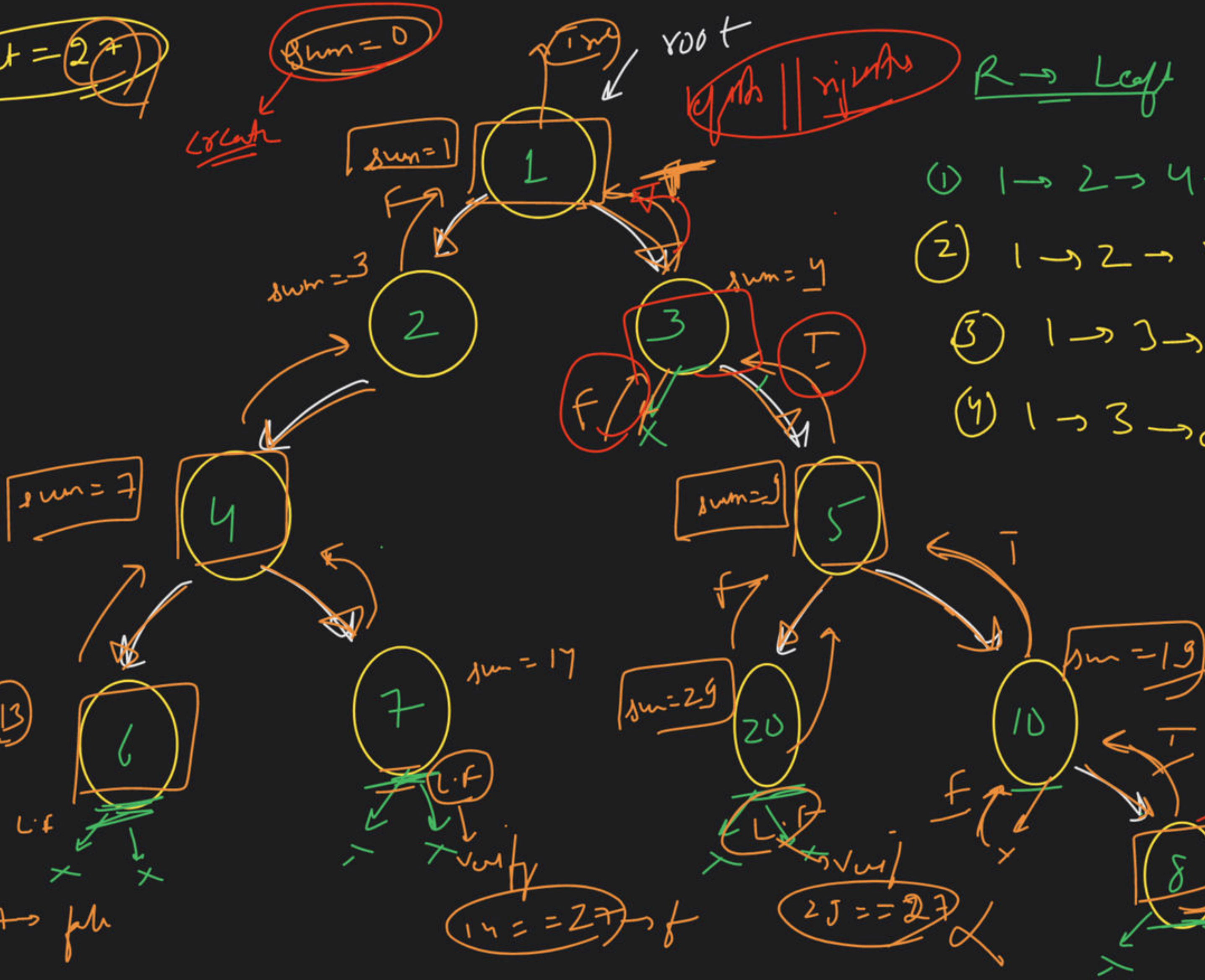
sum = 27

verify
13 == 27 → false

17 == 27 → false

29 == 27 → false

27 == 27 → true

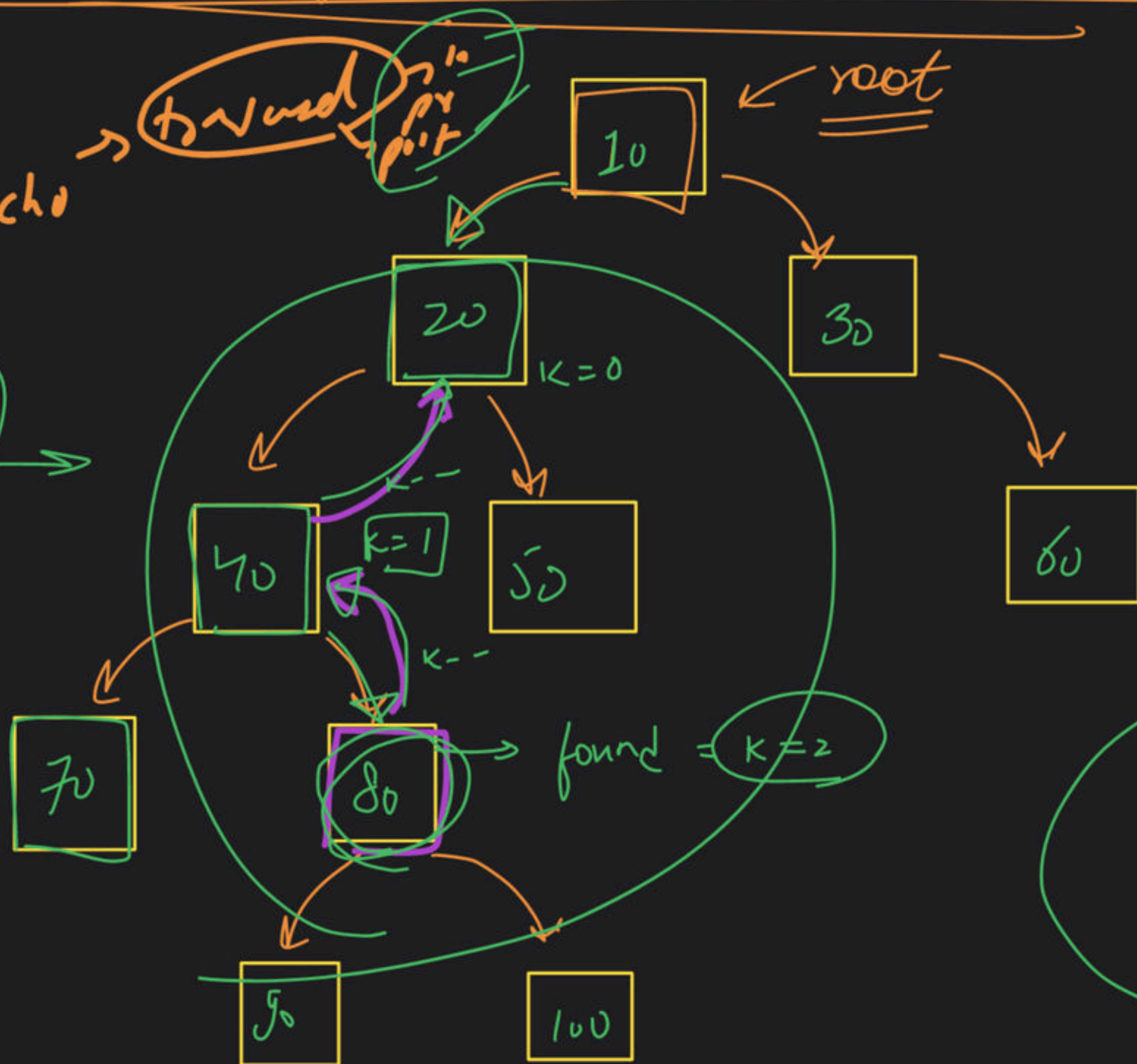


K^{th} Ancestor of a Node in Binary Tree

① So ta k gahcho

②

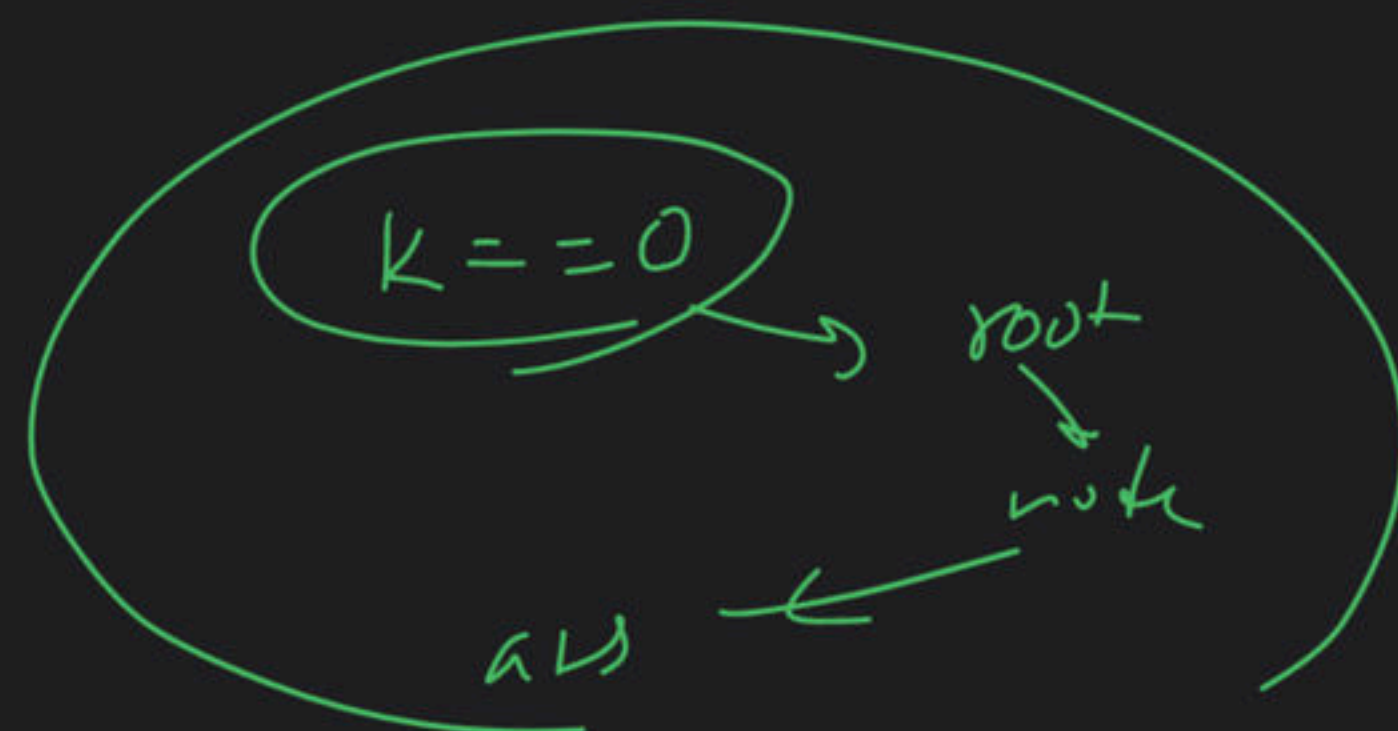
H/W



80 → node

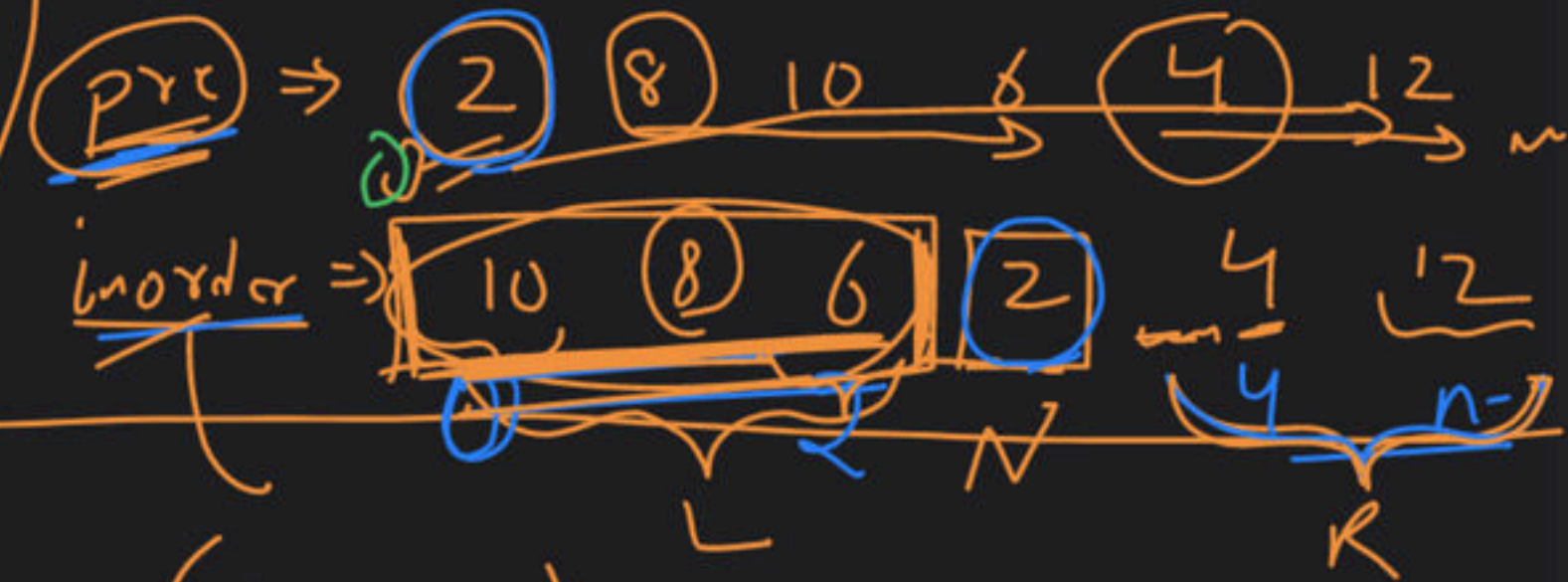
$K=2$

ans = 20





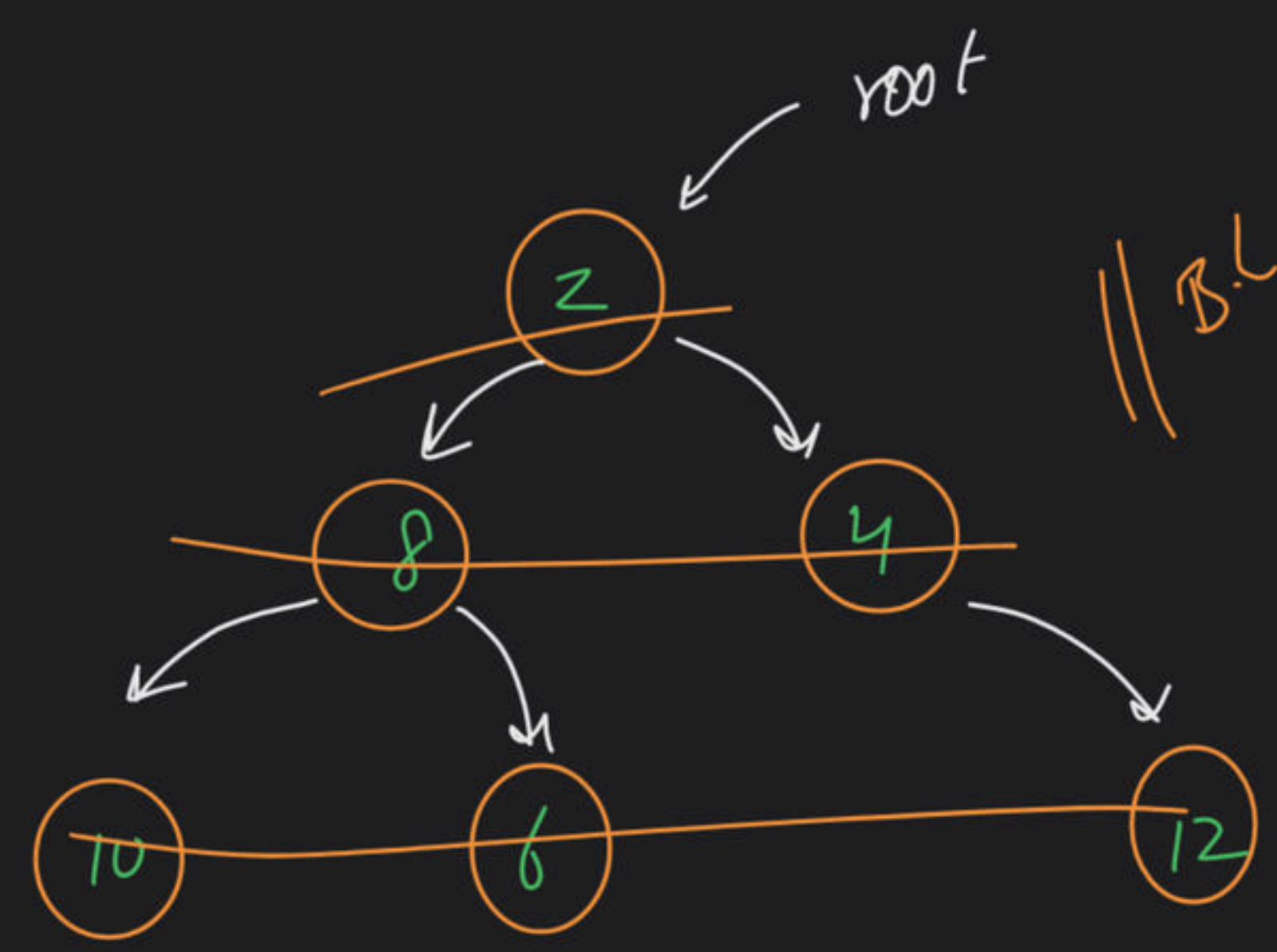
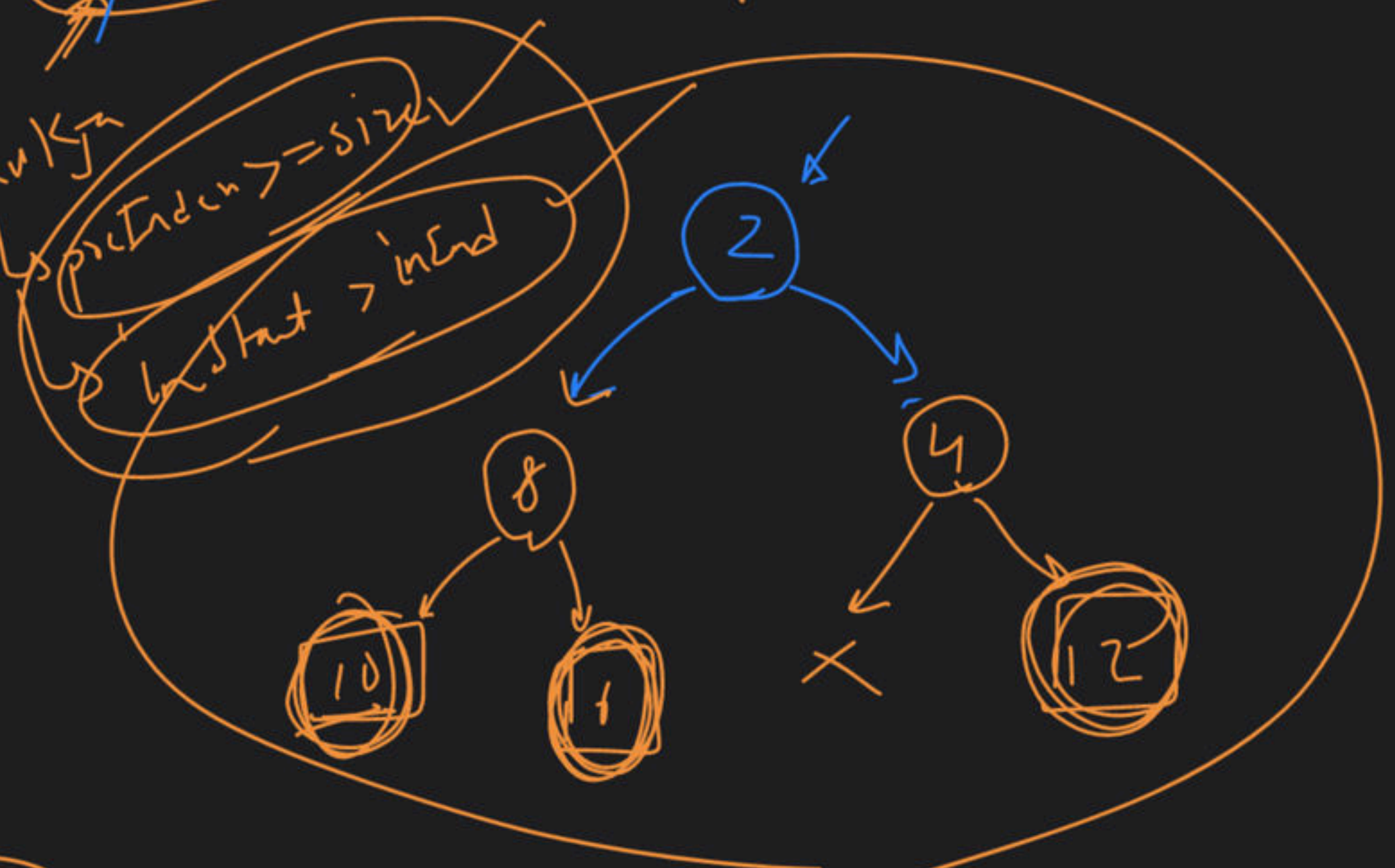
Pre Order & Inorder traversal



NLR

(LNR)

B.L \rightarrow Rekya

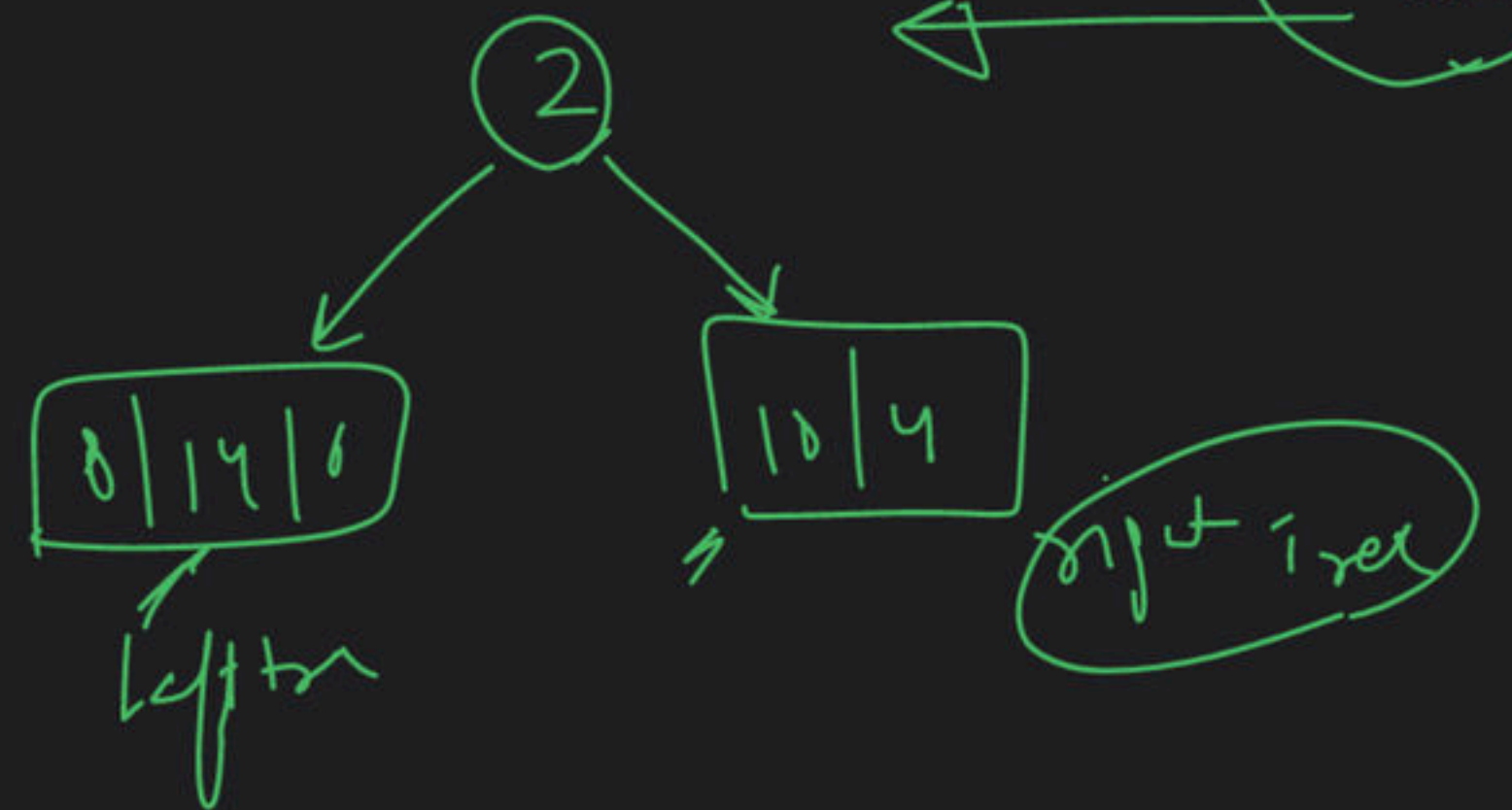
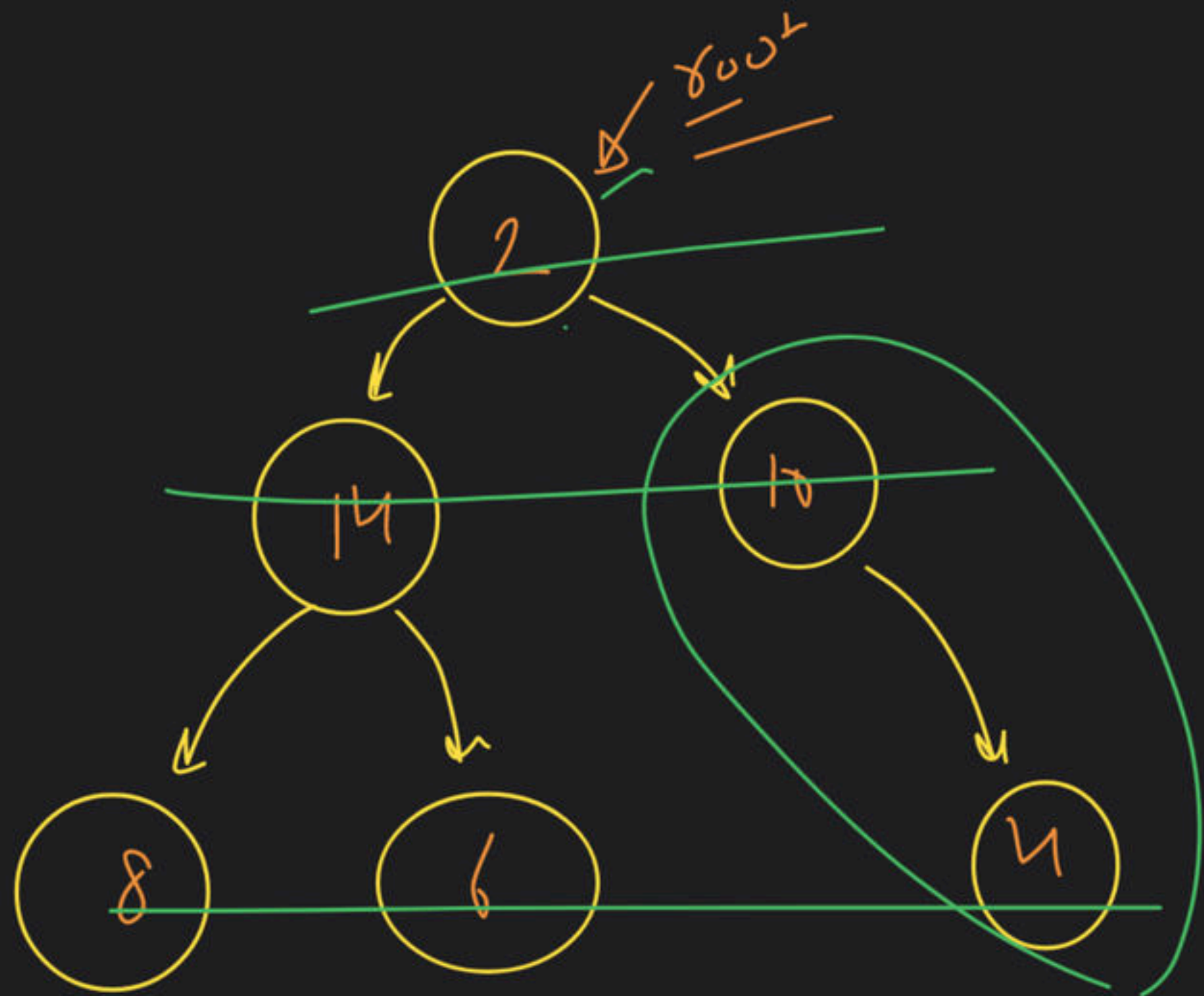


solve(preOrder, preOrderIndex, inorder, inorderStart, inorderEnd, size)

Post & In \longrightarrow Tree

inorder = (L N R) 8 14 6 2 10 4
ins pos inE

postorder = (L R N) 8 6 14 4 10 2
(L R N) (n-1) Index



Here \rightarrow Right tree
 Left call

inorder \rightarrow 10 8 6 2 4 12
0 1 2 3 4 5

valueToIndexMap[6] \rightarrow 2

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



```
map <int, int> valueToIndexMap;
```

```
m[10] = 0
```

```
for (i = 0; i < n;
```

