Trees

Foundation Course on Data Structures & Algorithms - Part II

Love Babbar • Lesson 6 • July 7, 2022

Trees :-

Node

root

child

Parent

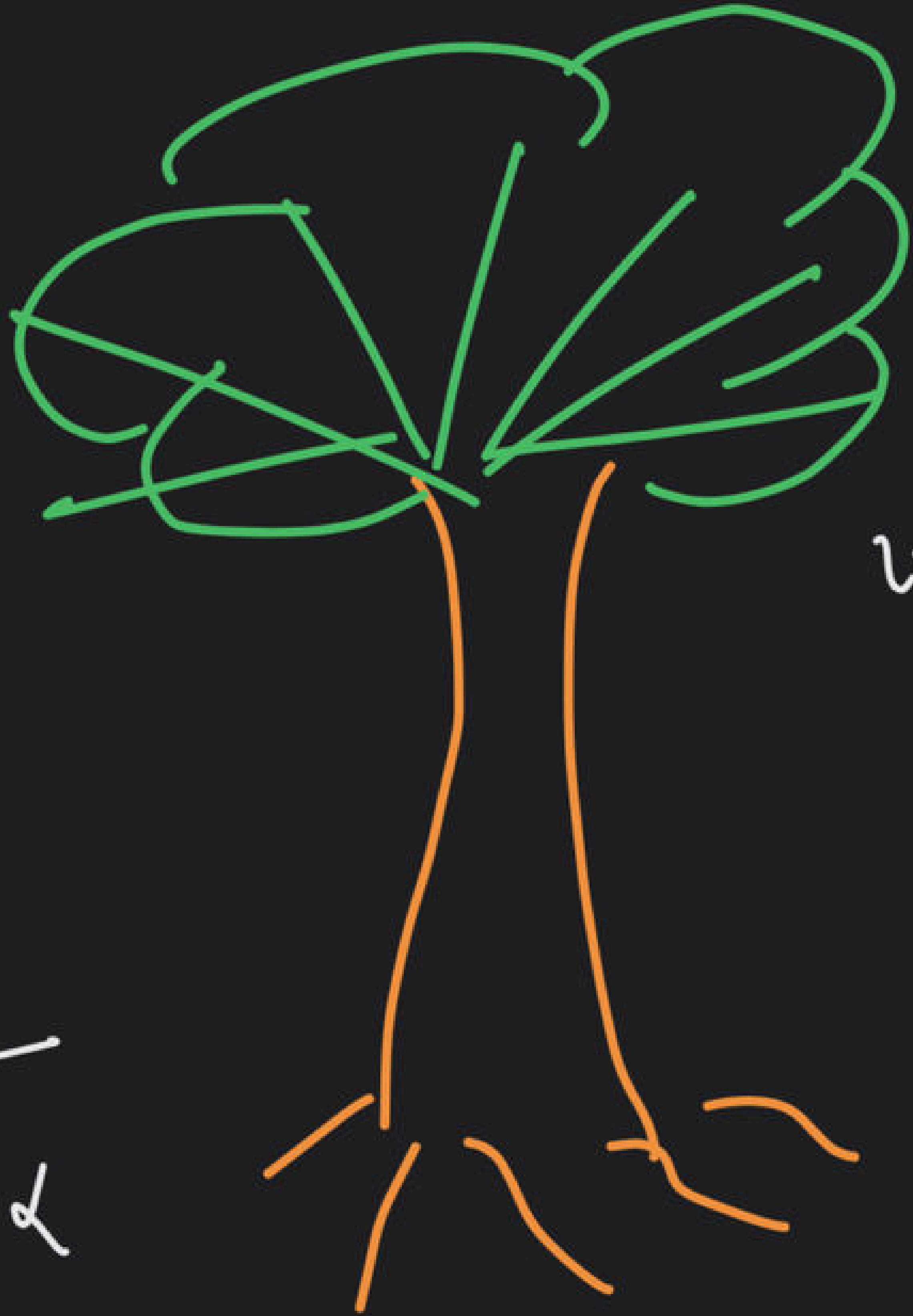
Sibling

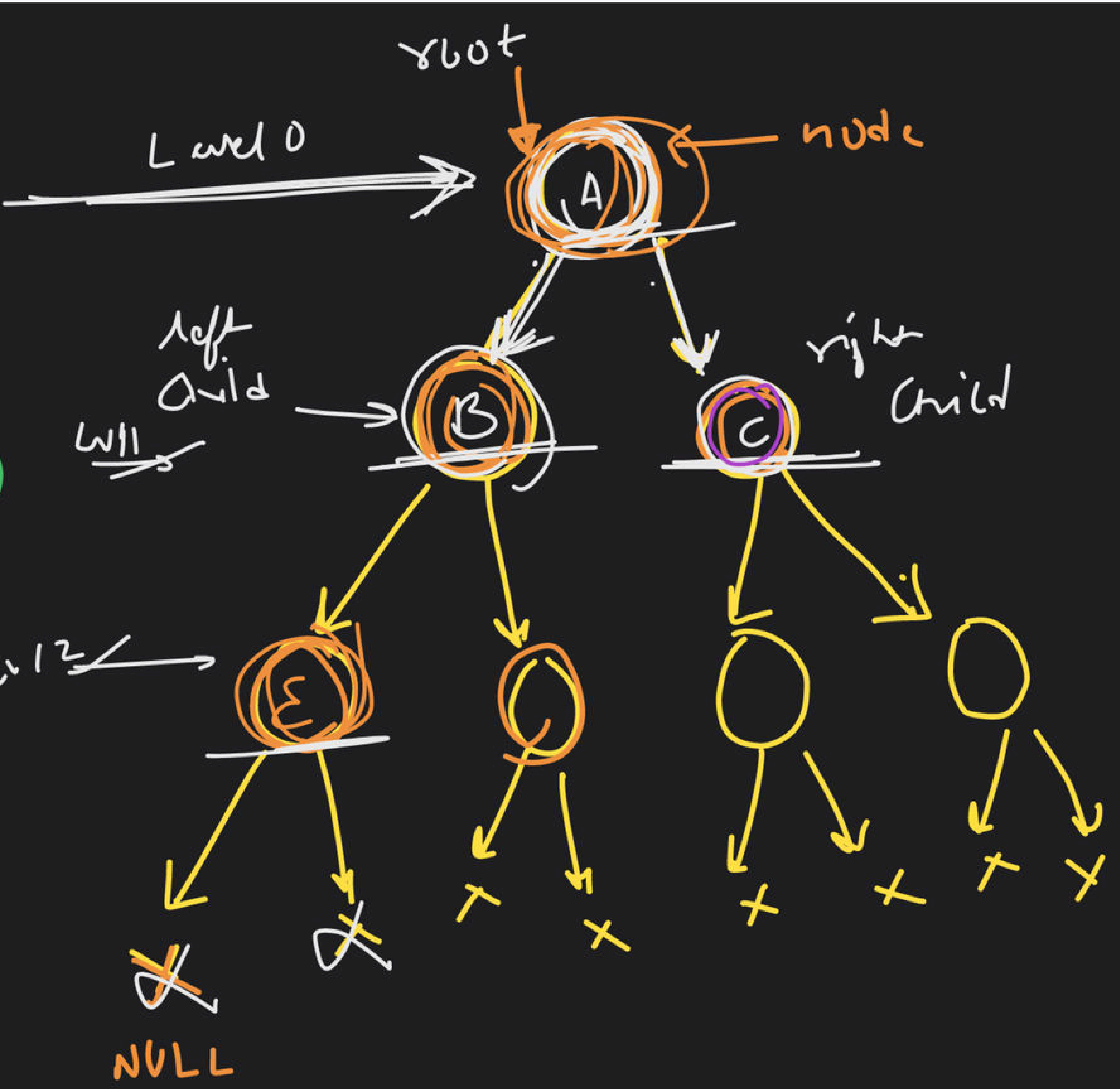
ancestor

leaf node

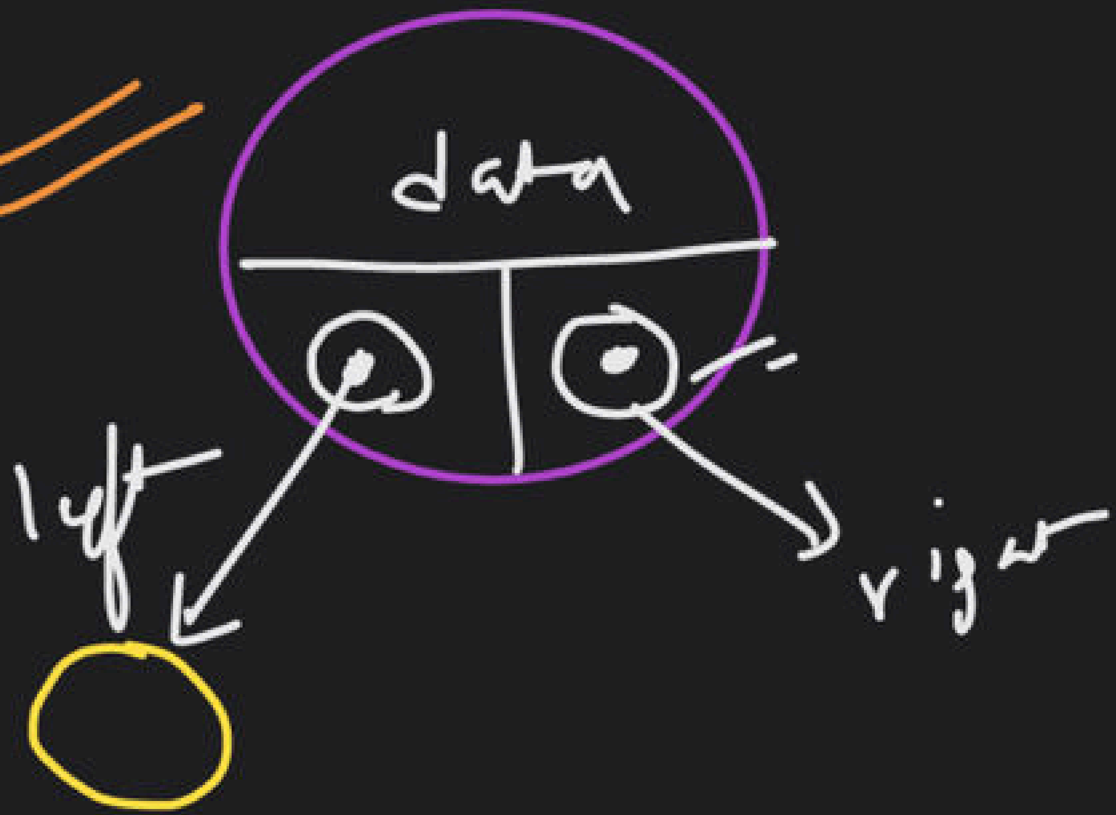
↓

child





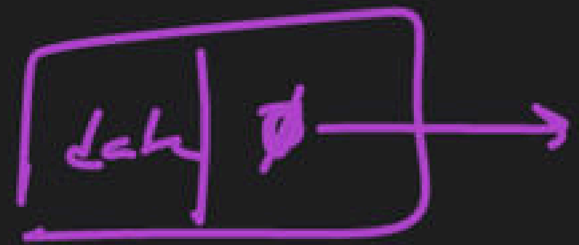
Node →



```
class   Node
{
    int  data;
    Node* left;
    Node* right;
};
```

Binary tree
↳ Node
↳ atmax
↳ 2 child

LL



n-ary tree =
↳ Node
↳ n child

array of pointer =
↳ n size

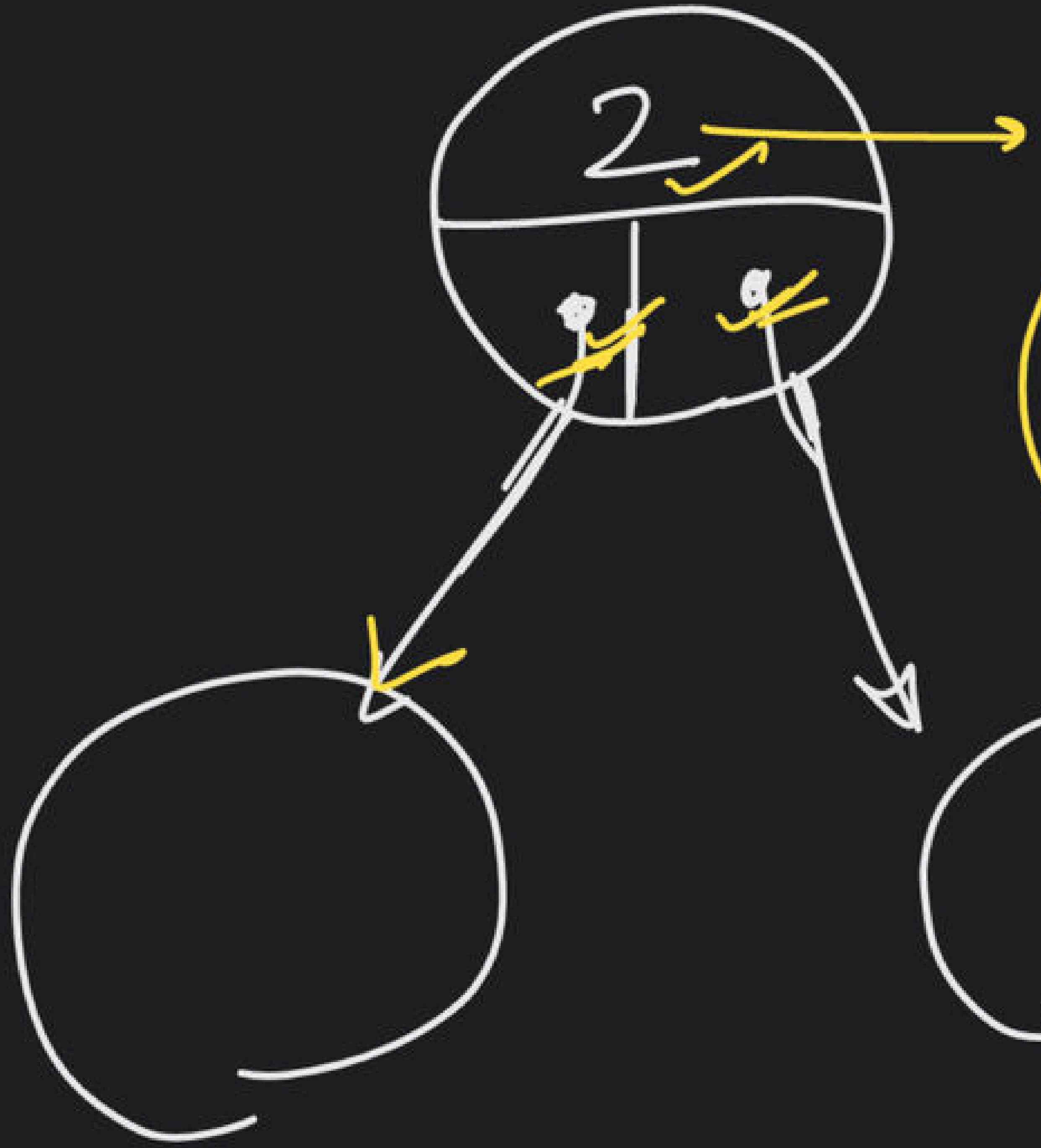
n pointers

int arr[n]

B+D
?

int* arr[n]

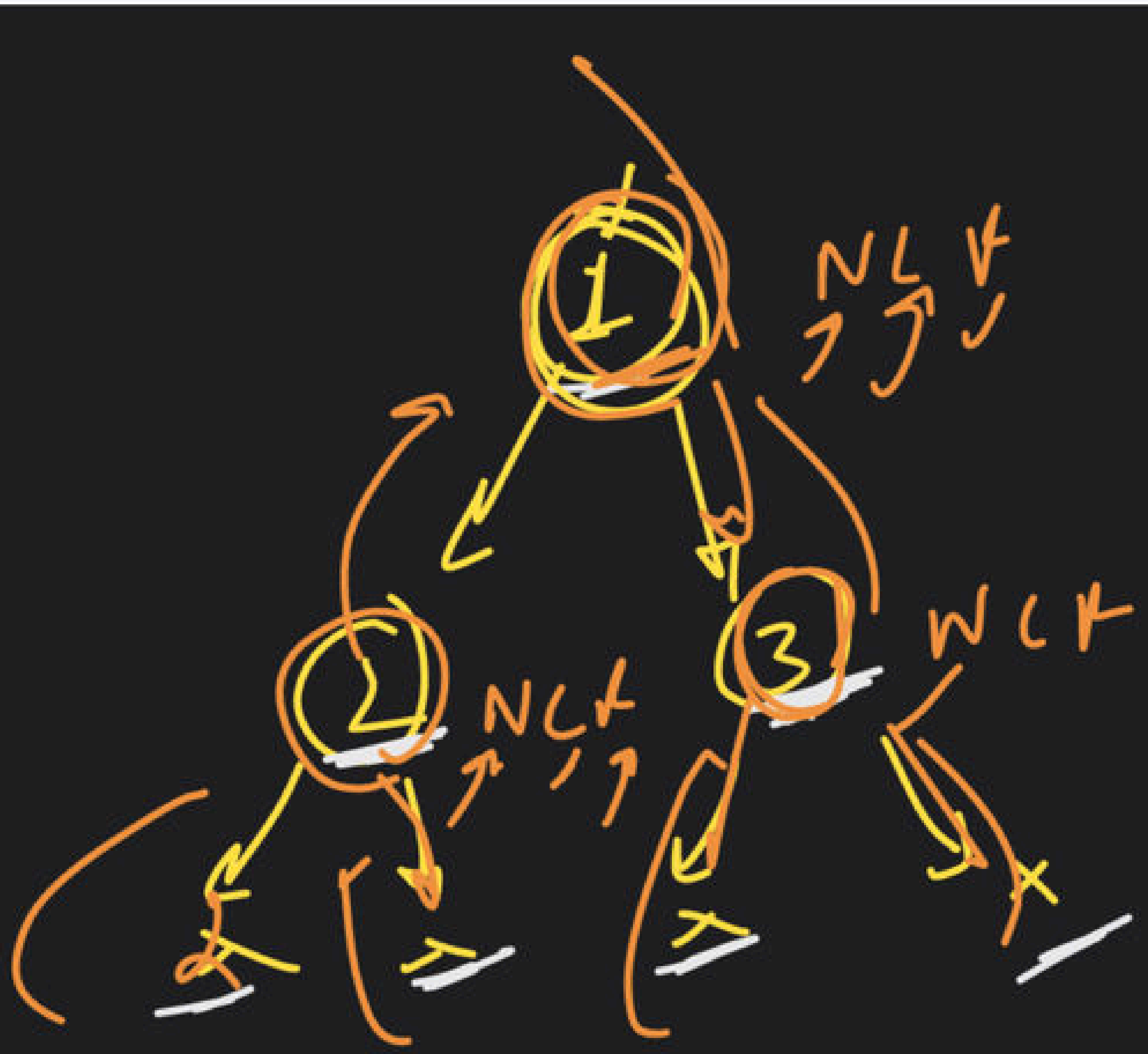
Binary tree:-



```
int data;
Node* left;
Node* right;
```

Dream child <= 2;

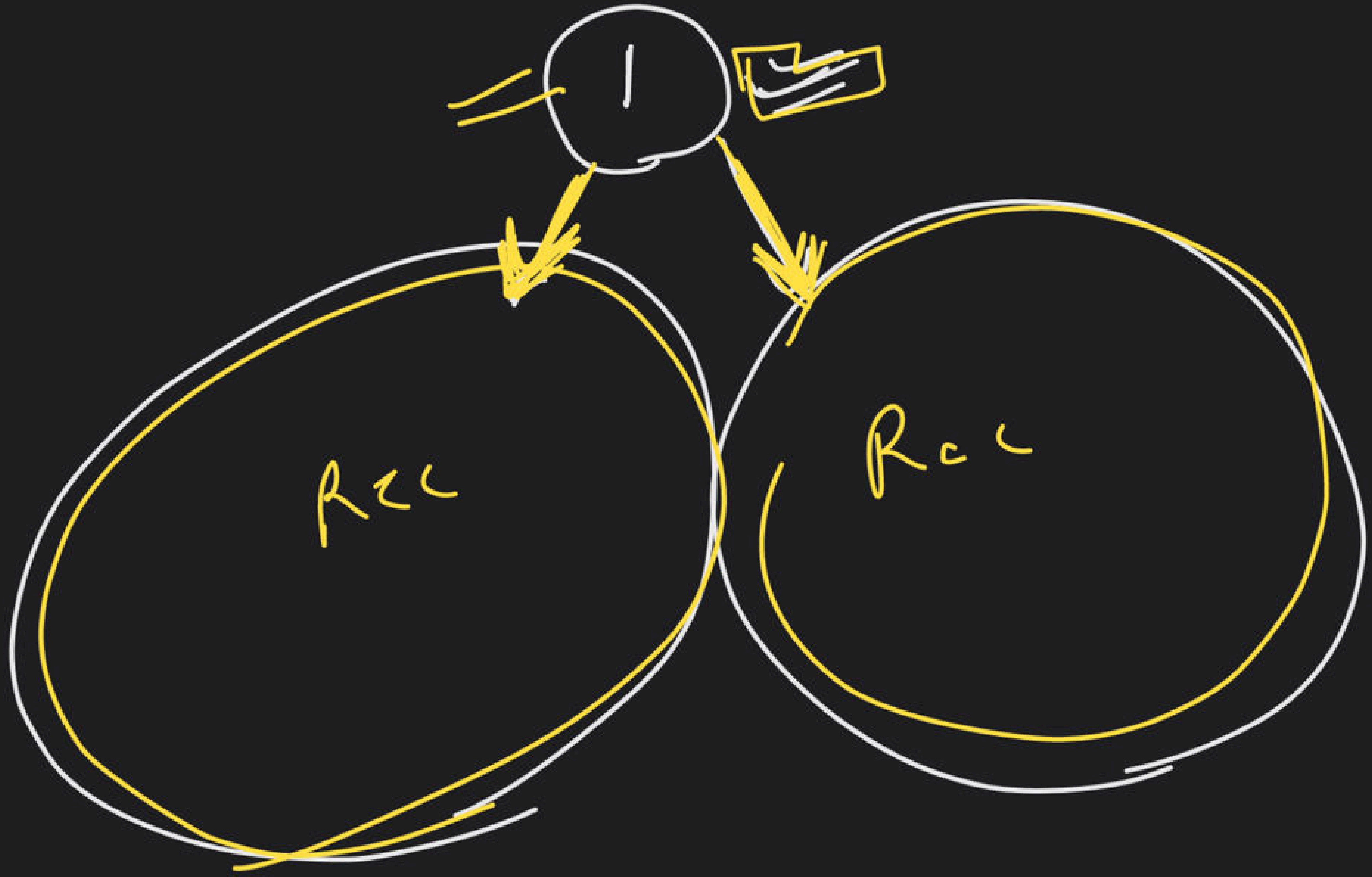
1 min



1 2 -1 -1 3 -1 -1

1 2 3

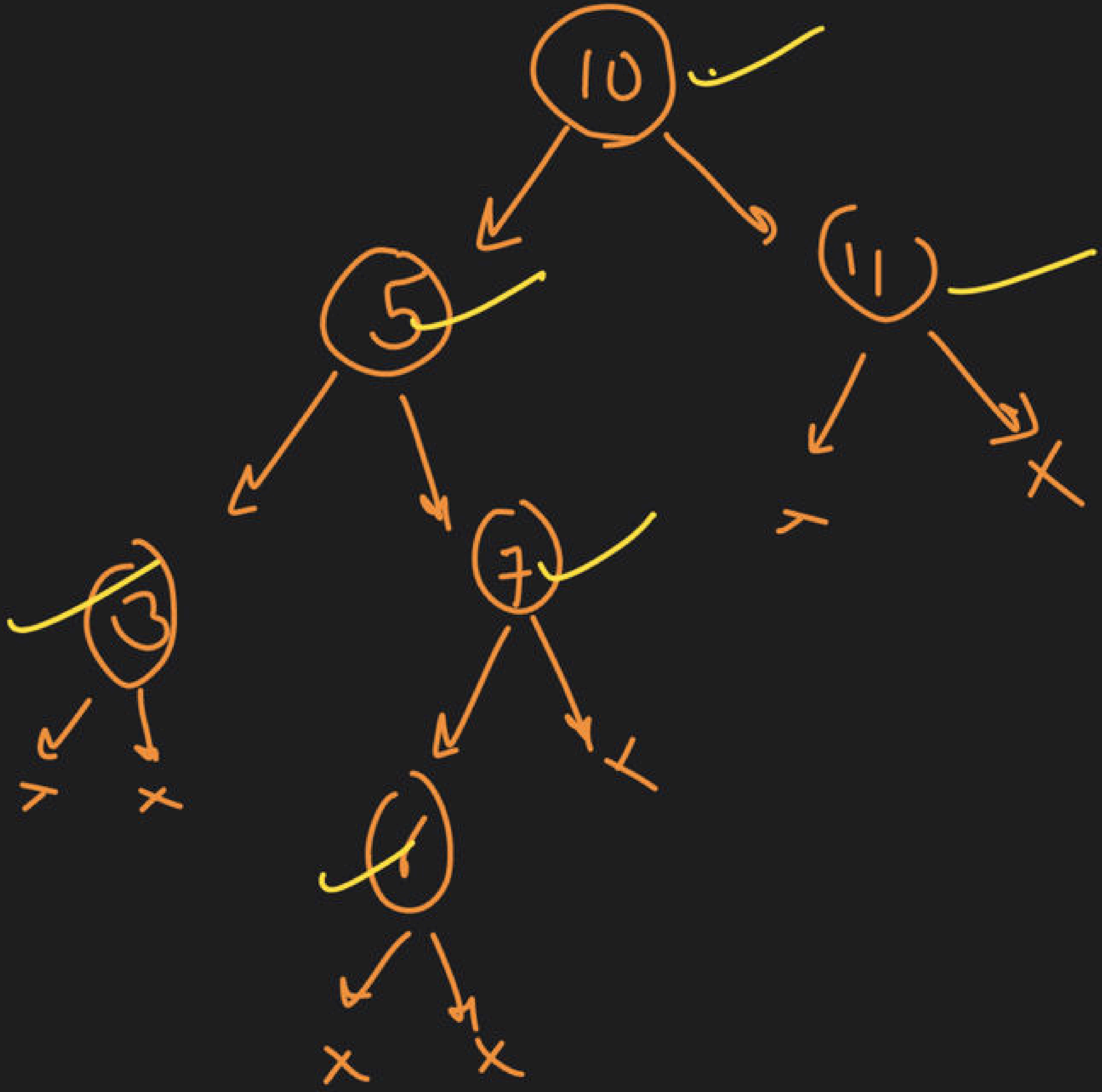
Easy



Tree

application

↳



30 sec

10 5 3 -1 -1 7 6

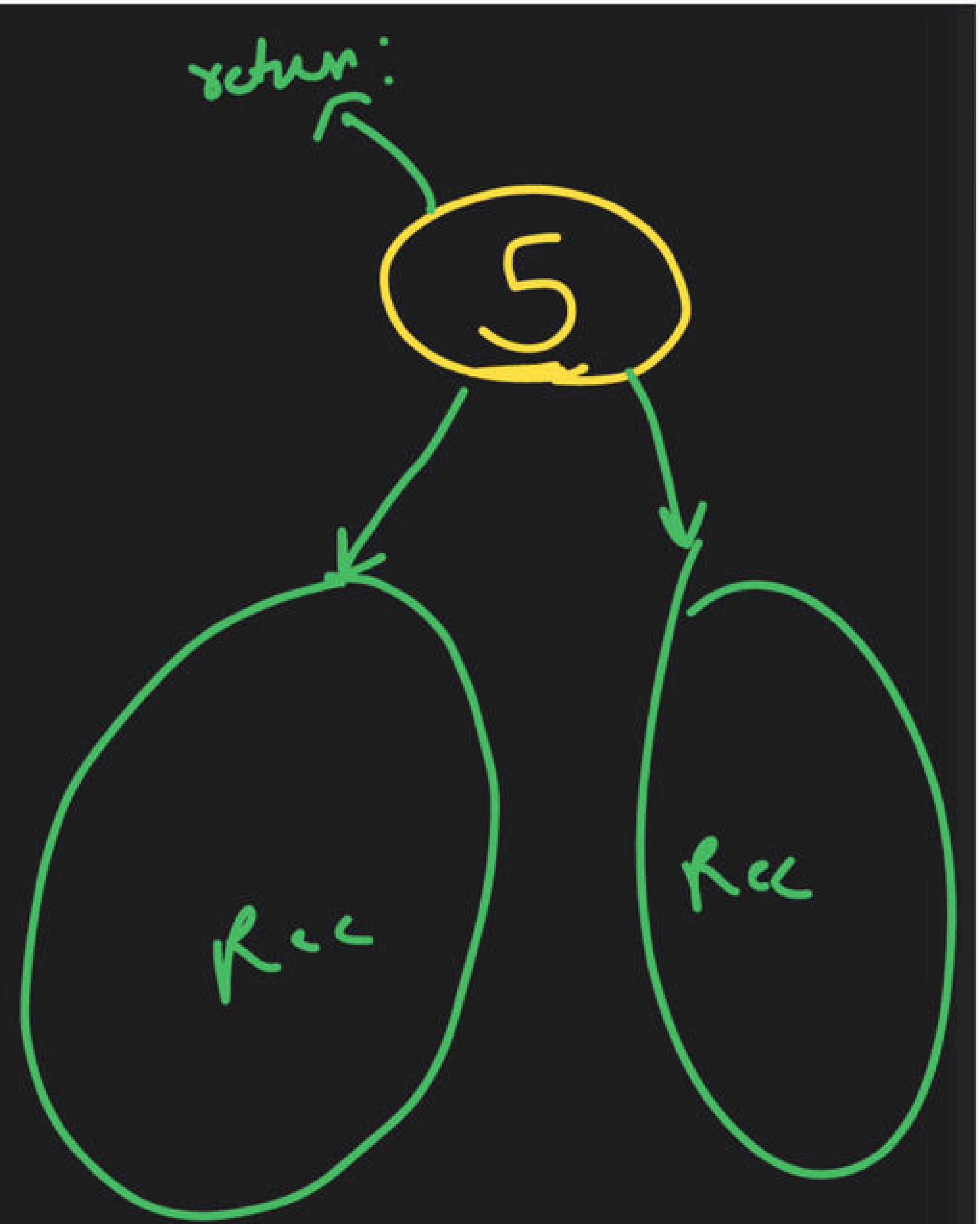
-1 -1 -1 11 -1 -1

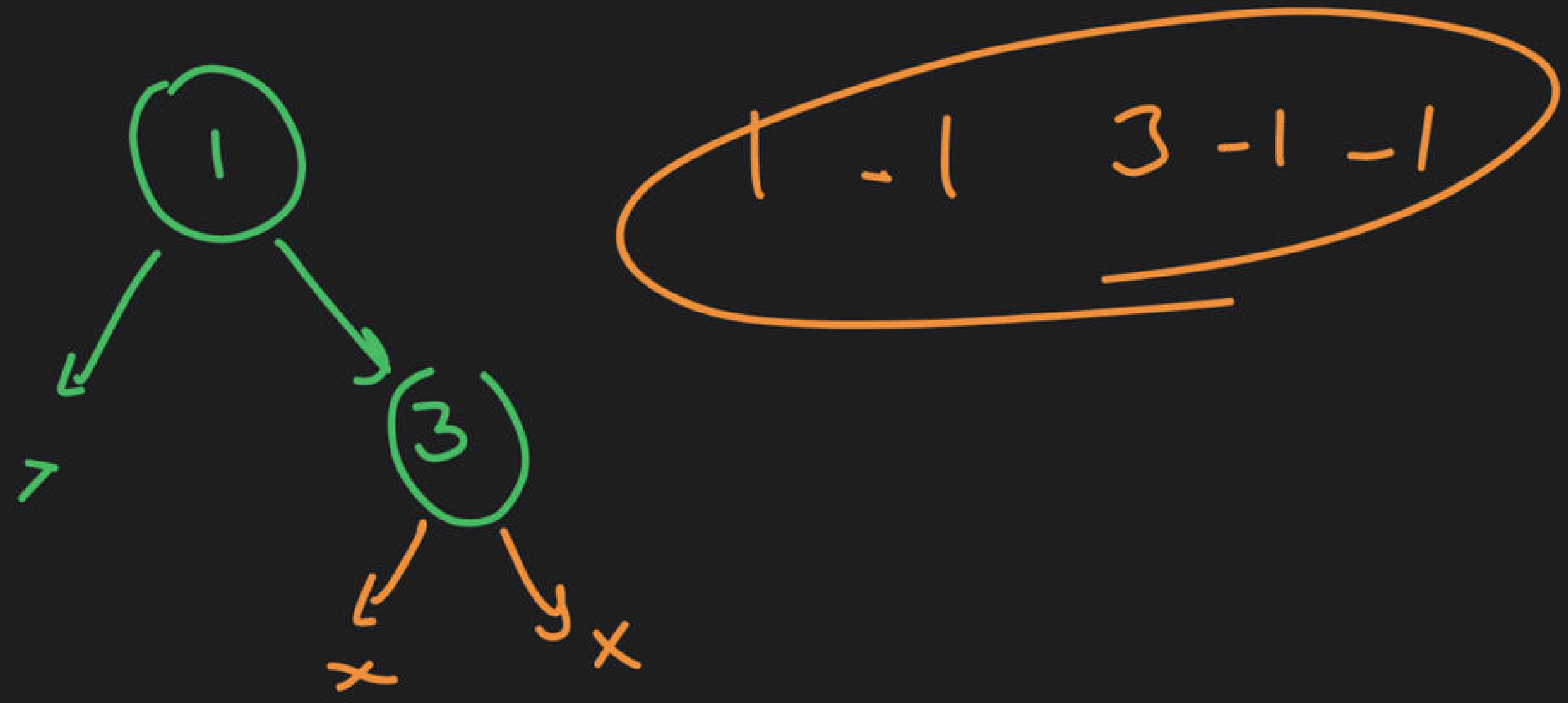
10 5 3 7 6 11

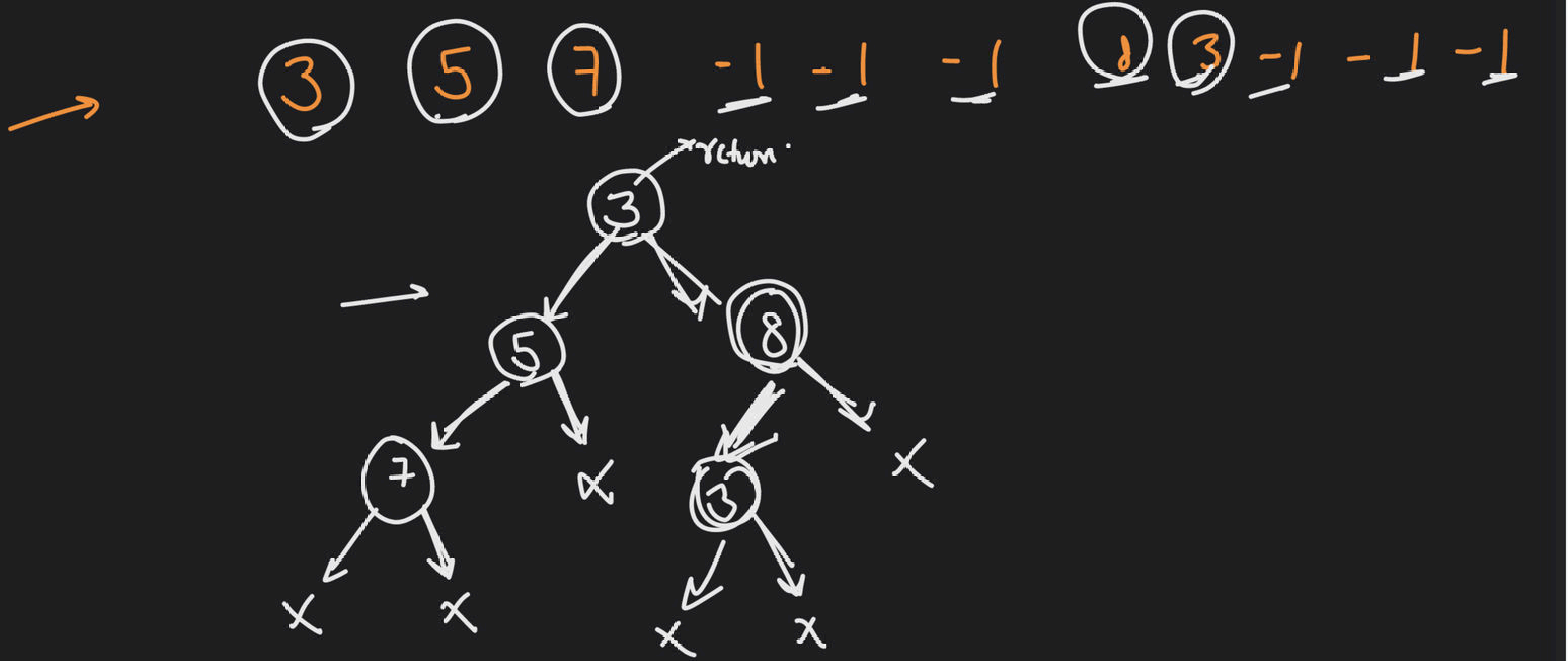
```
Node* buildTree()
{
    int data;
    cin >> data;
    Node* root = new Node(data);
        if (data == -1)
            return NULL;

    root -> left = buildTree();
    root -> right = buildTree();
      return root;
}
```

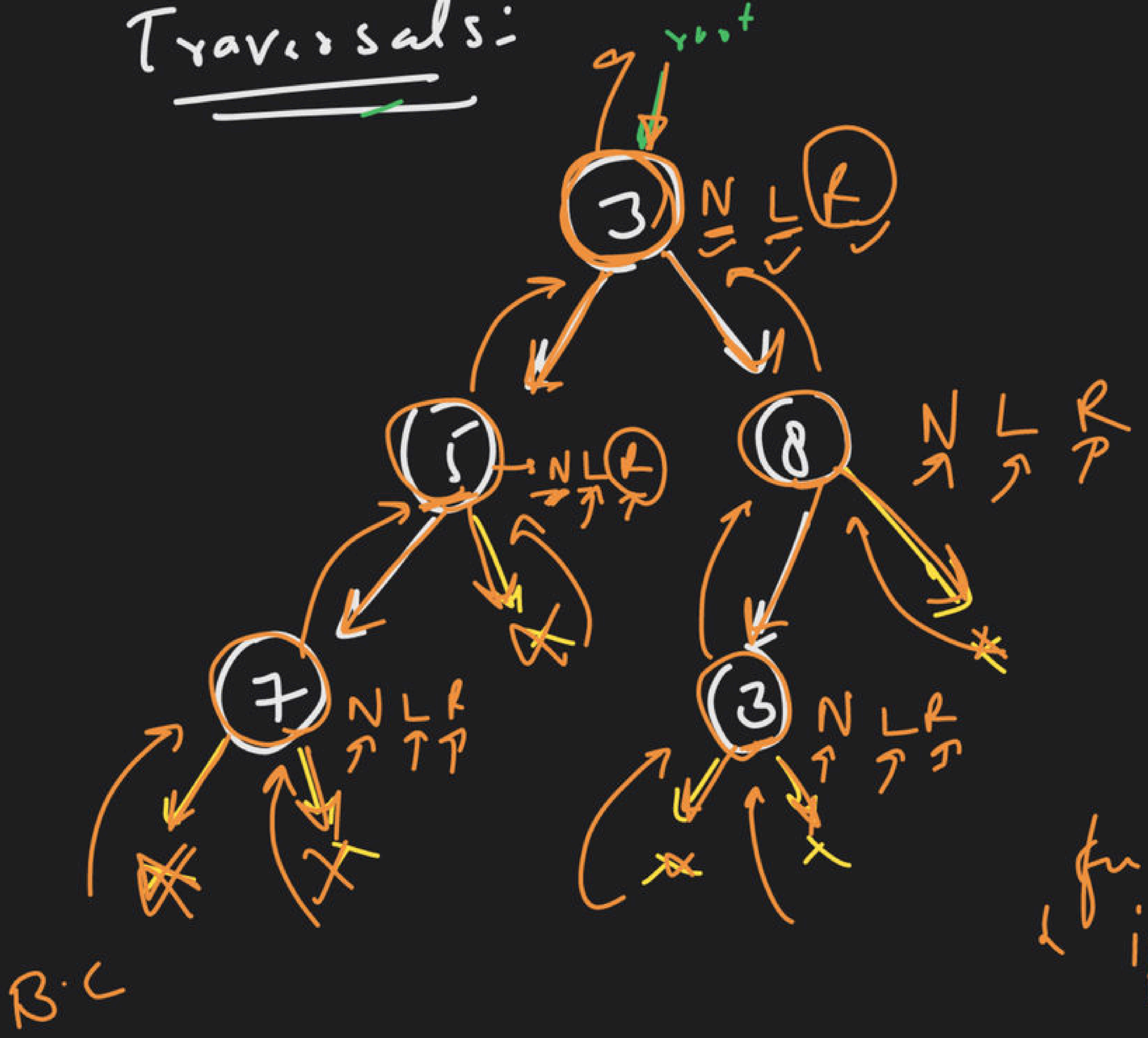
can this return NULL;



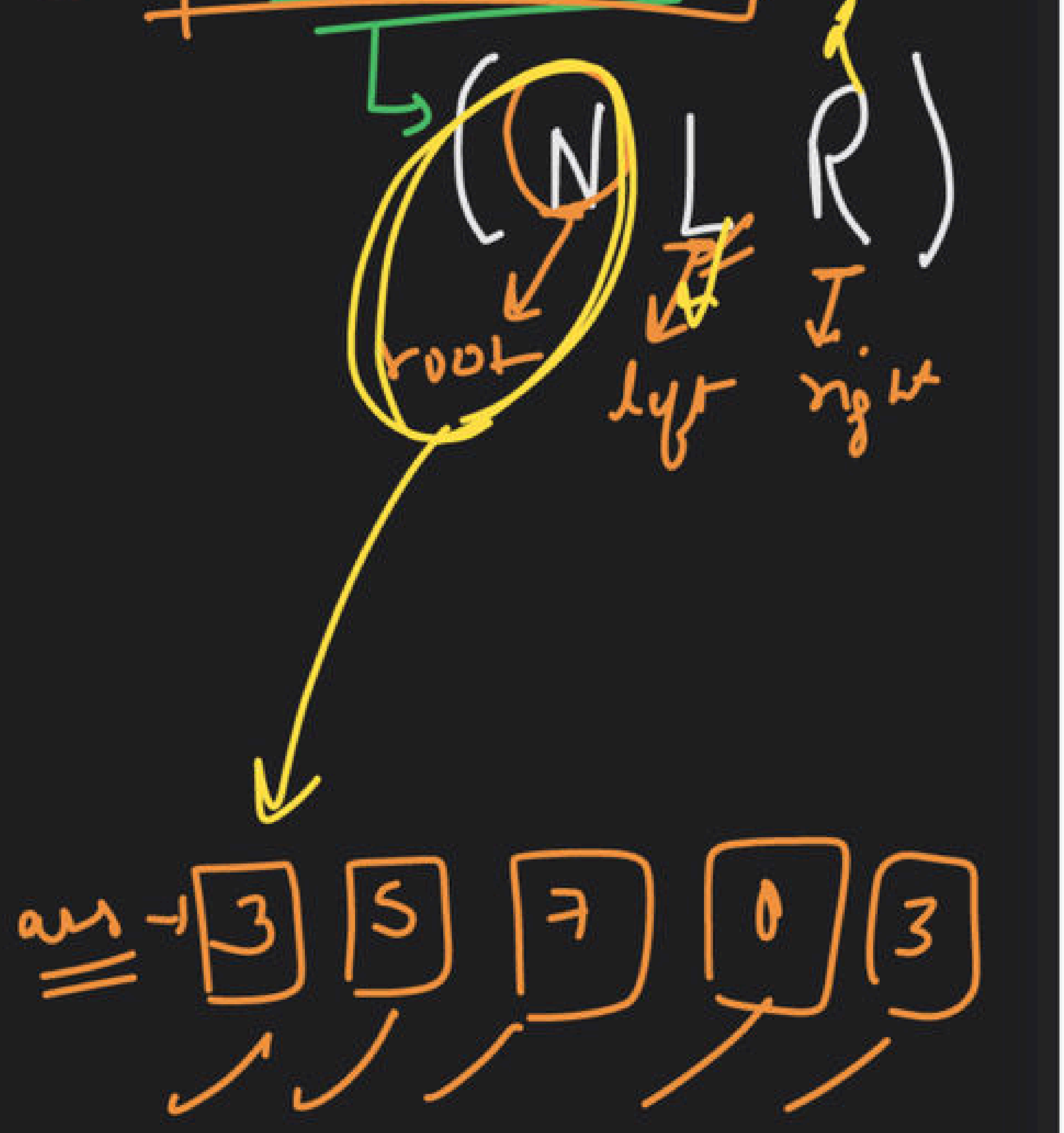

1
3
x
x
1 -1 3 -1 -1


3 5 7 -1 -1 -1 8 3 -1 -1 -1
return
3
5
8
7
x
3
x
x
x
x
x
x

Traversals:



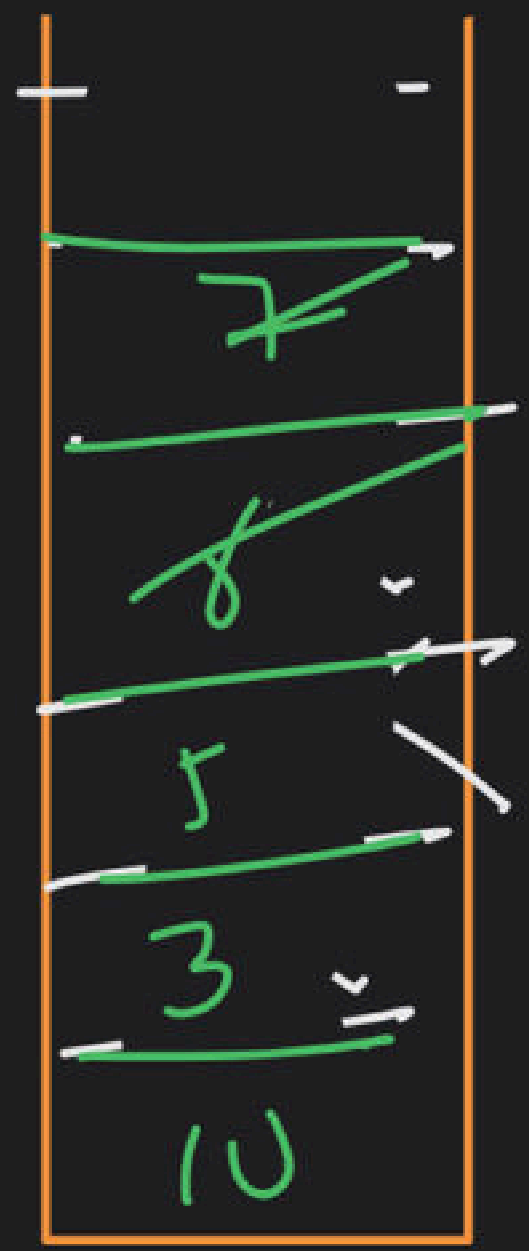
① PreOrder

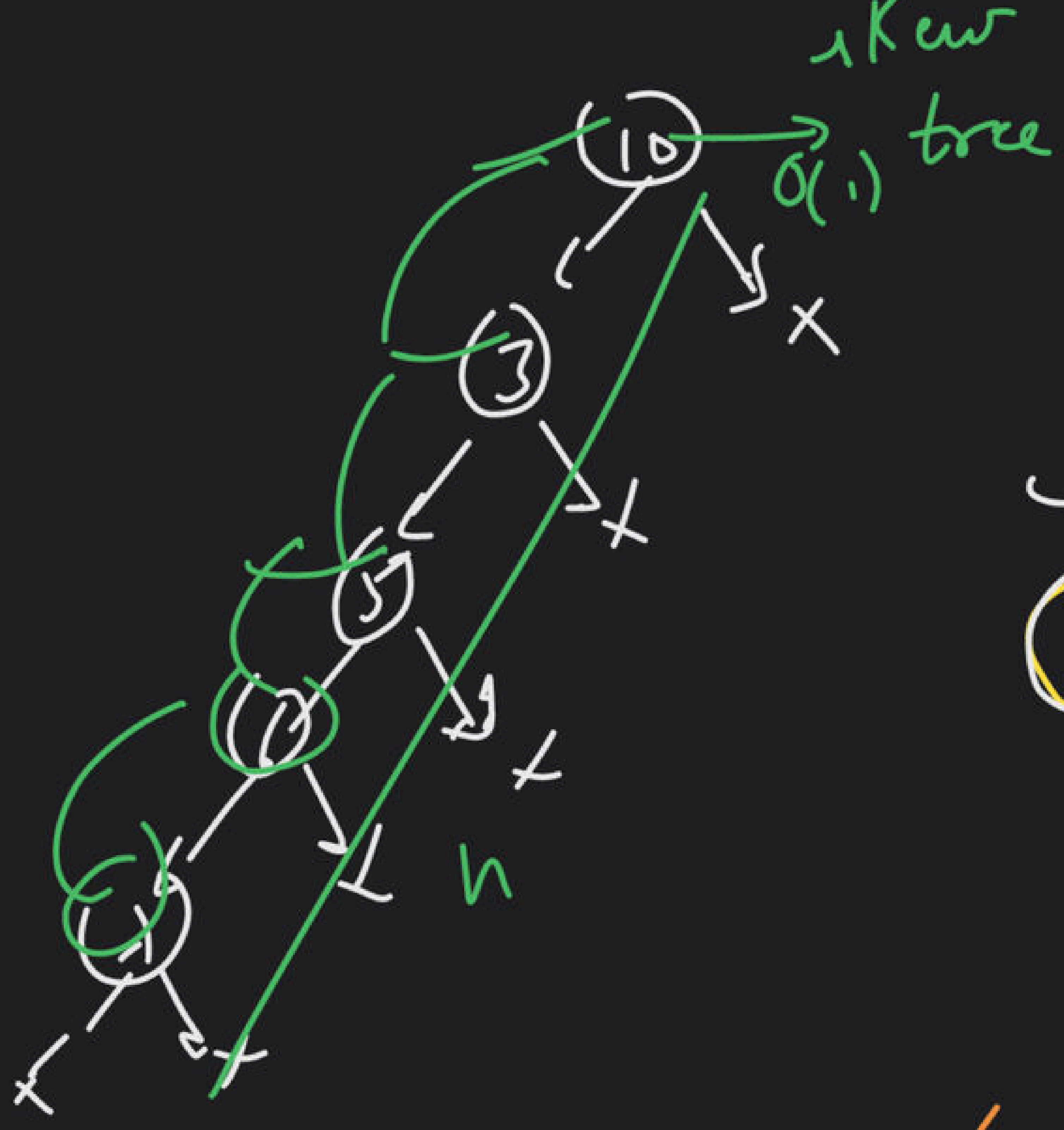


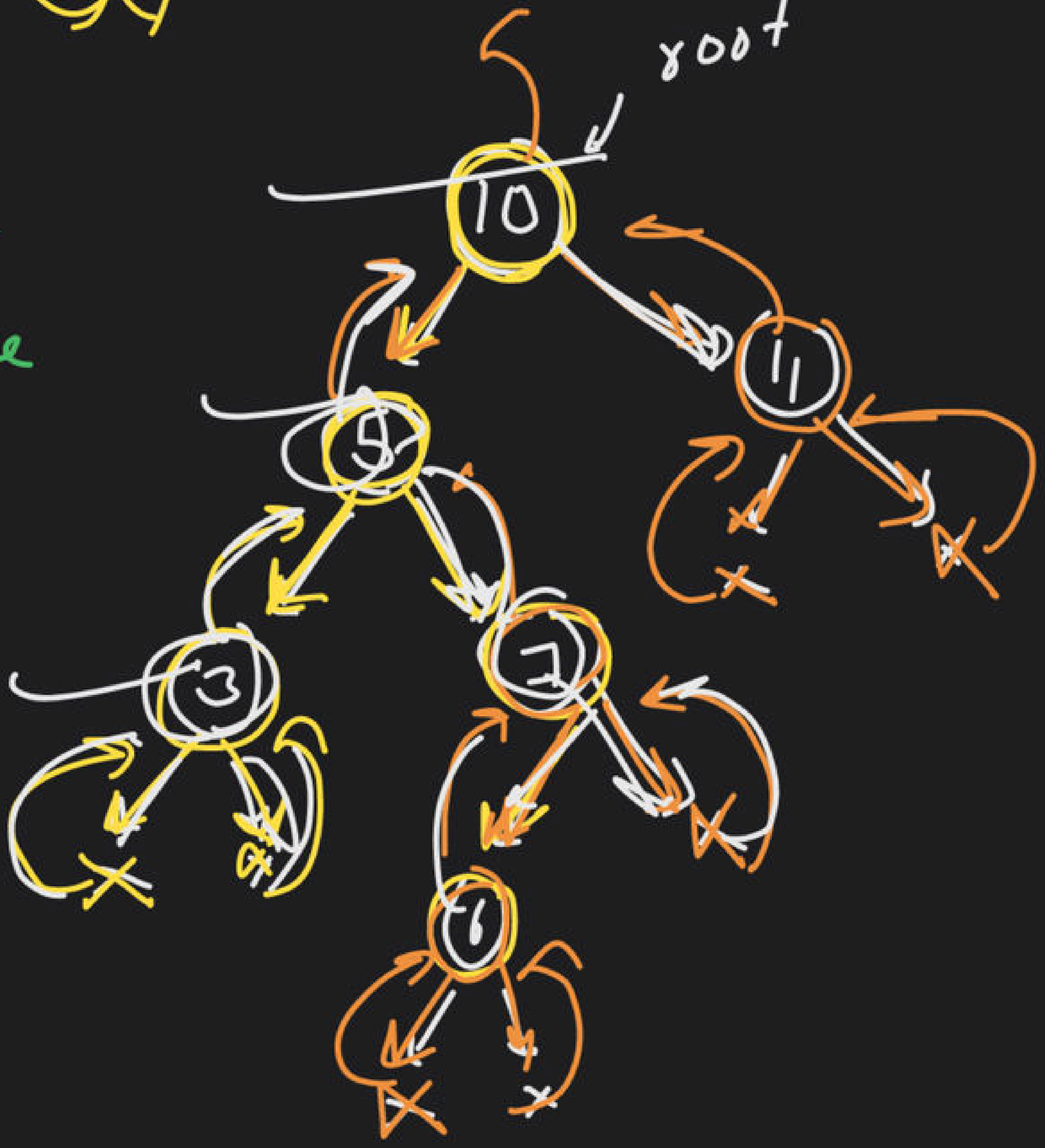
```
fun()
{
    if (root == NULL)
        return;
    cout << root->data;
    fun(left);    fun(ri
}
```

NLR

```
void preOrder (node* & root)
{     //
```





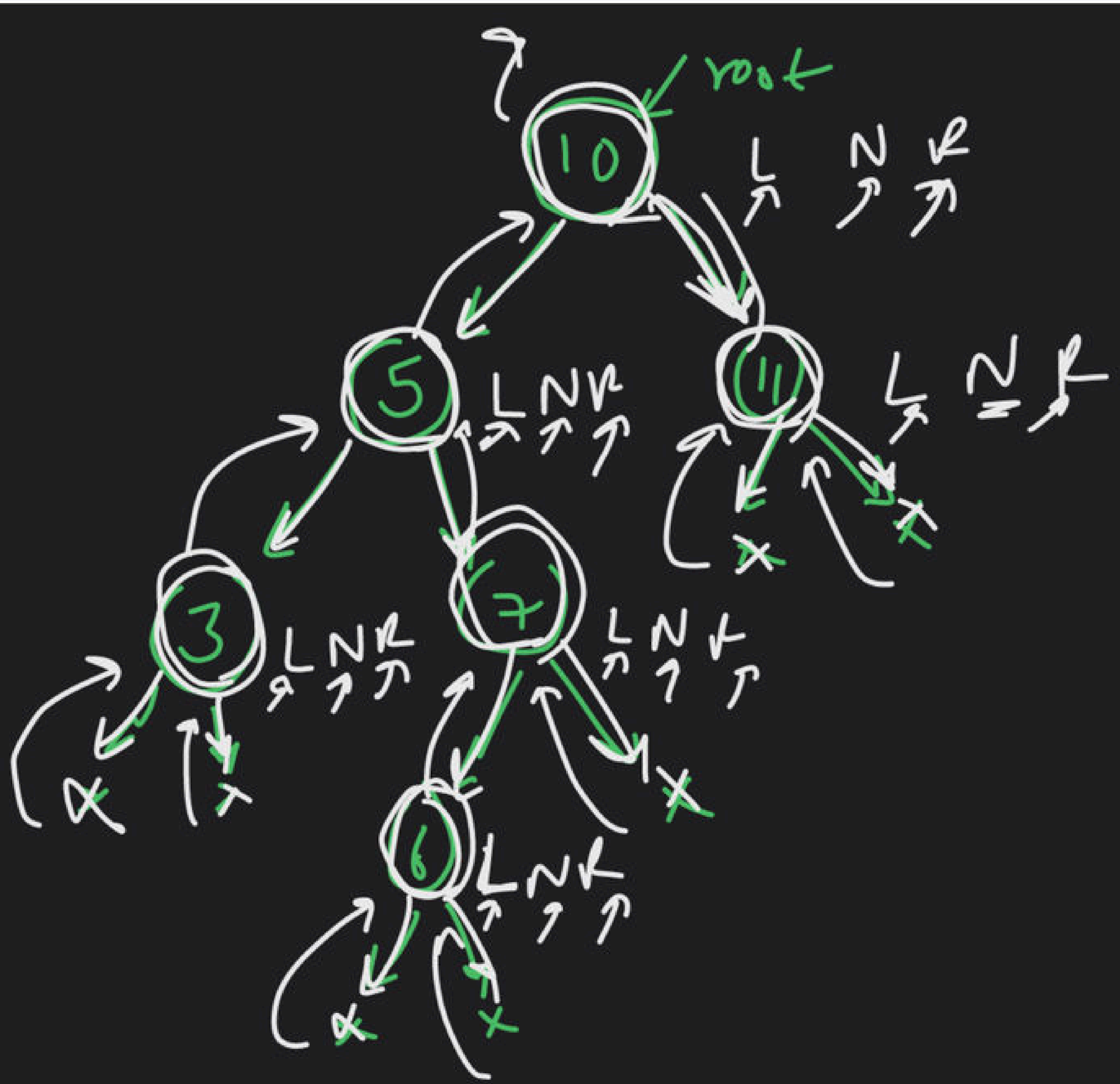


10 5 3 7 6 11

T.C -> O(N)

S.C -> O(N)

Inorder traversal → L N R



Inorder

↳ LNR

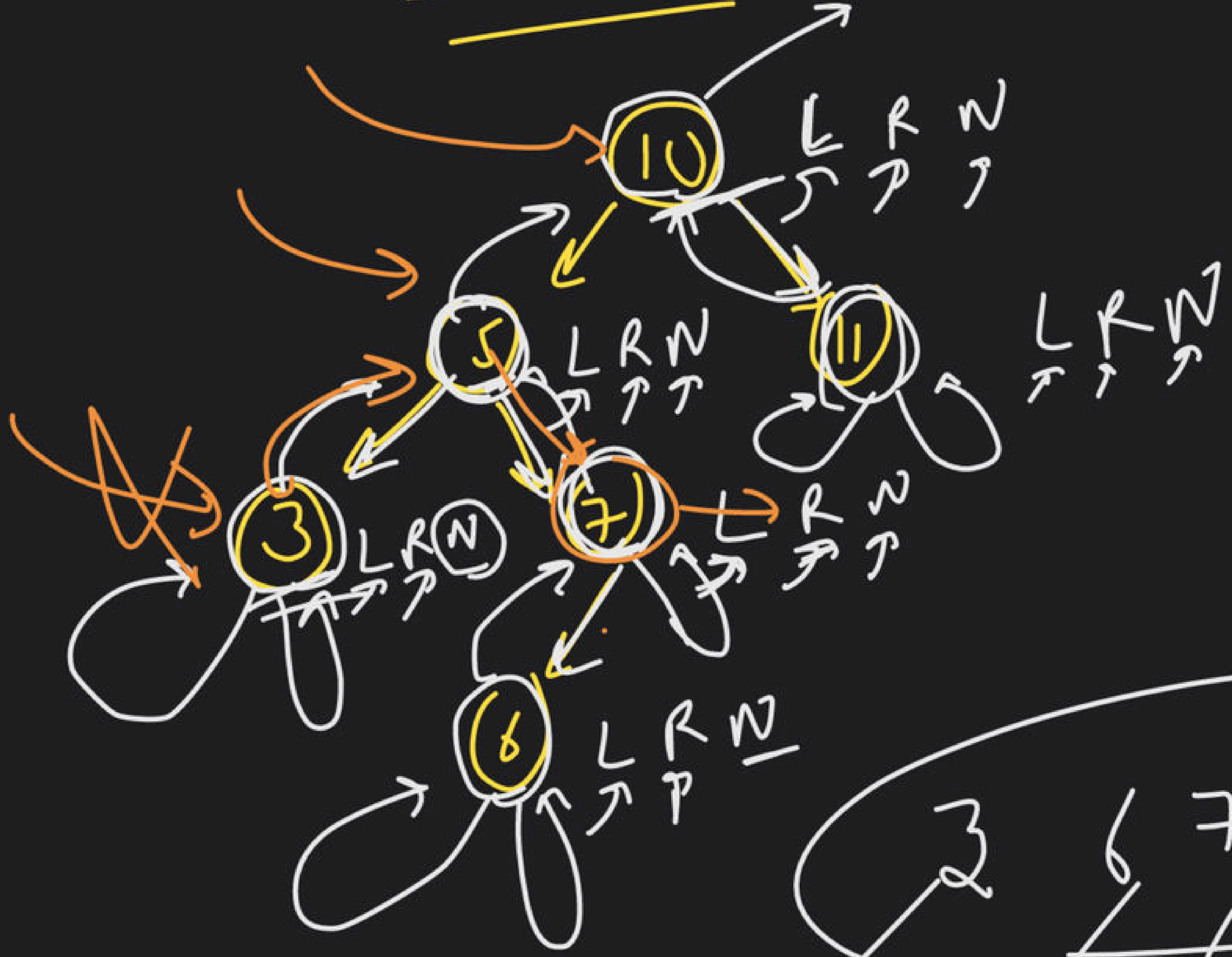
T.C → O(n)

S.C → O(n)

3 5 6 7 10 11

Post Order

→ LRN



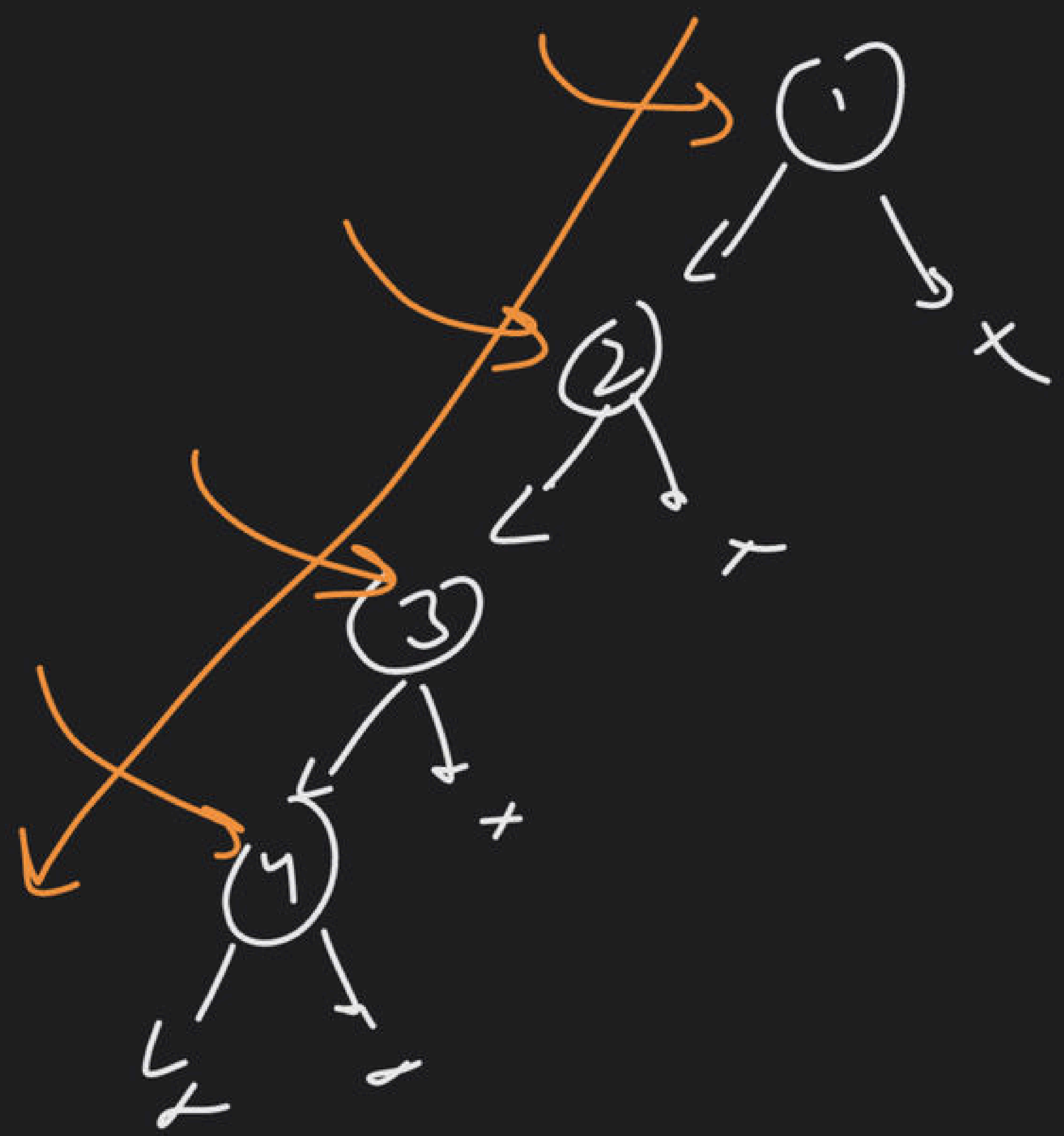
Post Ordourrr

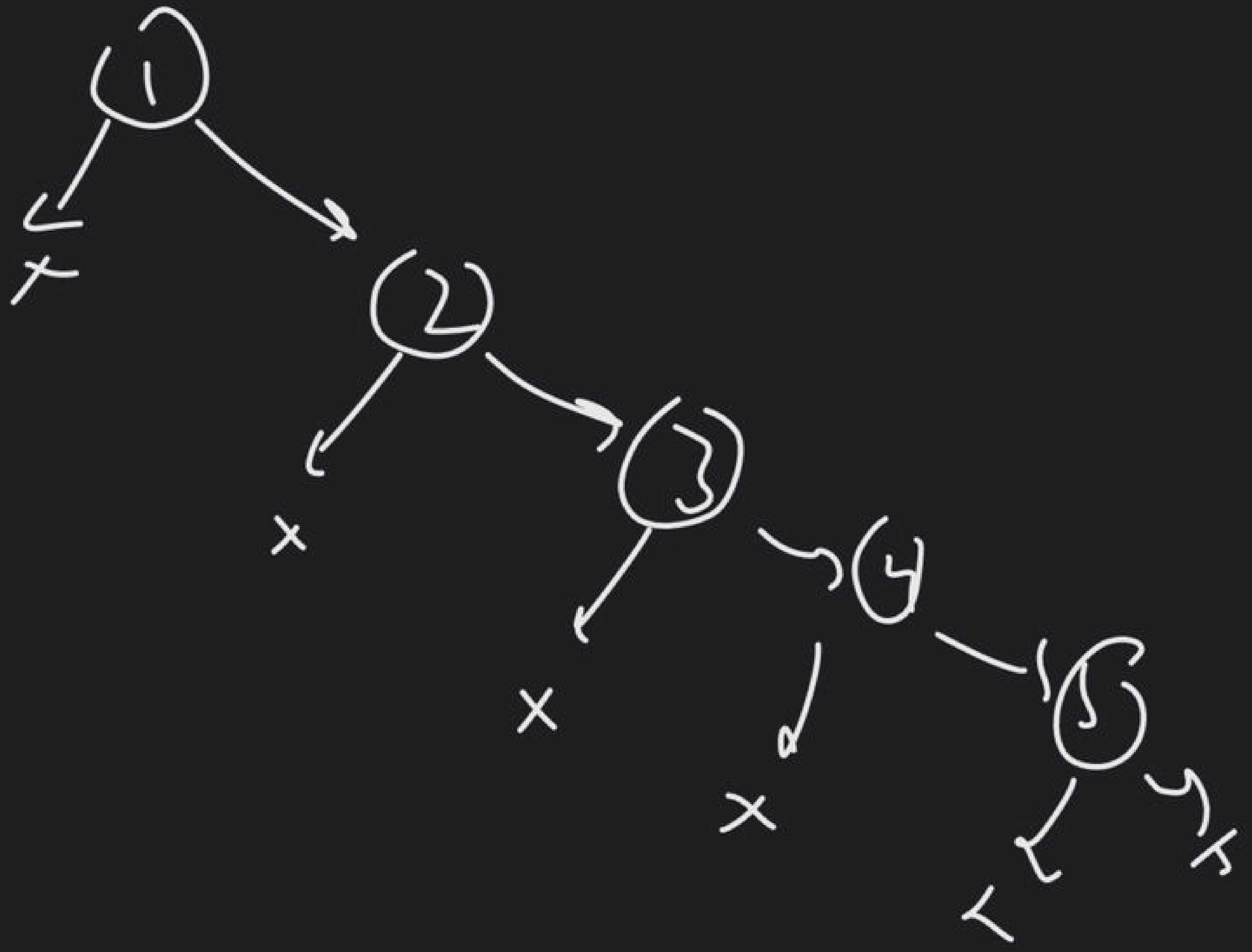
T.C → O(n)

S.C → O(n)

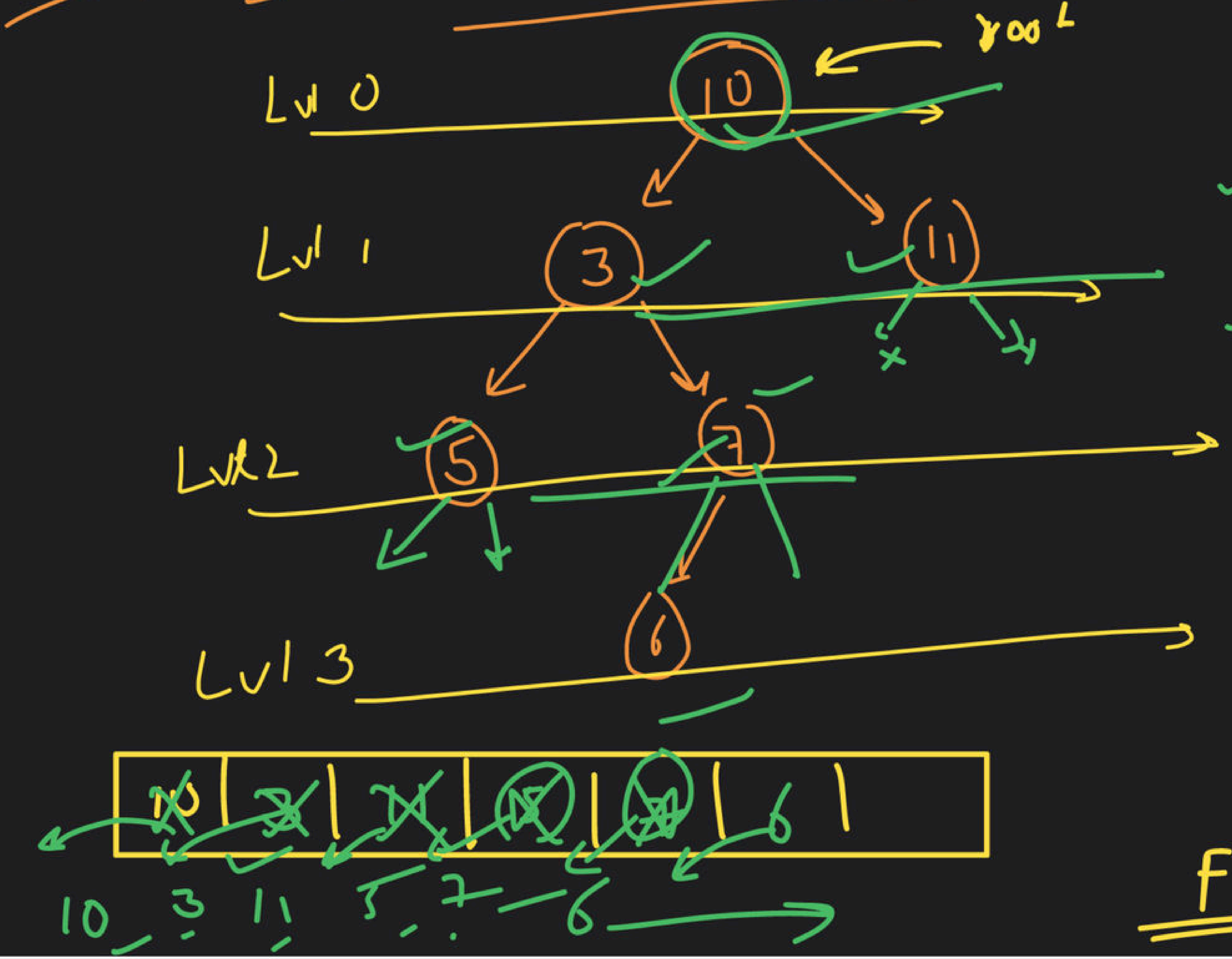
3 6 7 5 11 10

skew tree:





→ Level Order traversal:-



algo:- sun
le nikal
&
bacche chor
jaiyo

→ 10

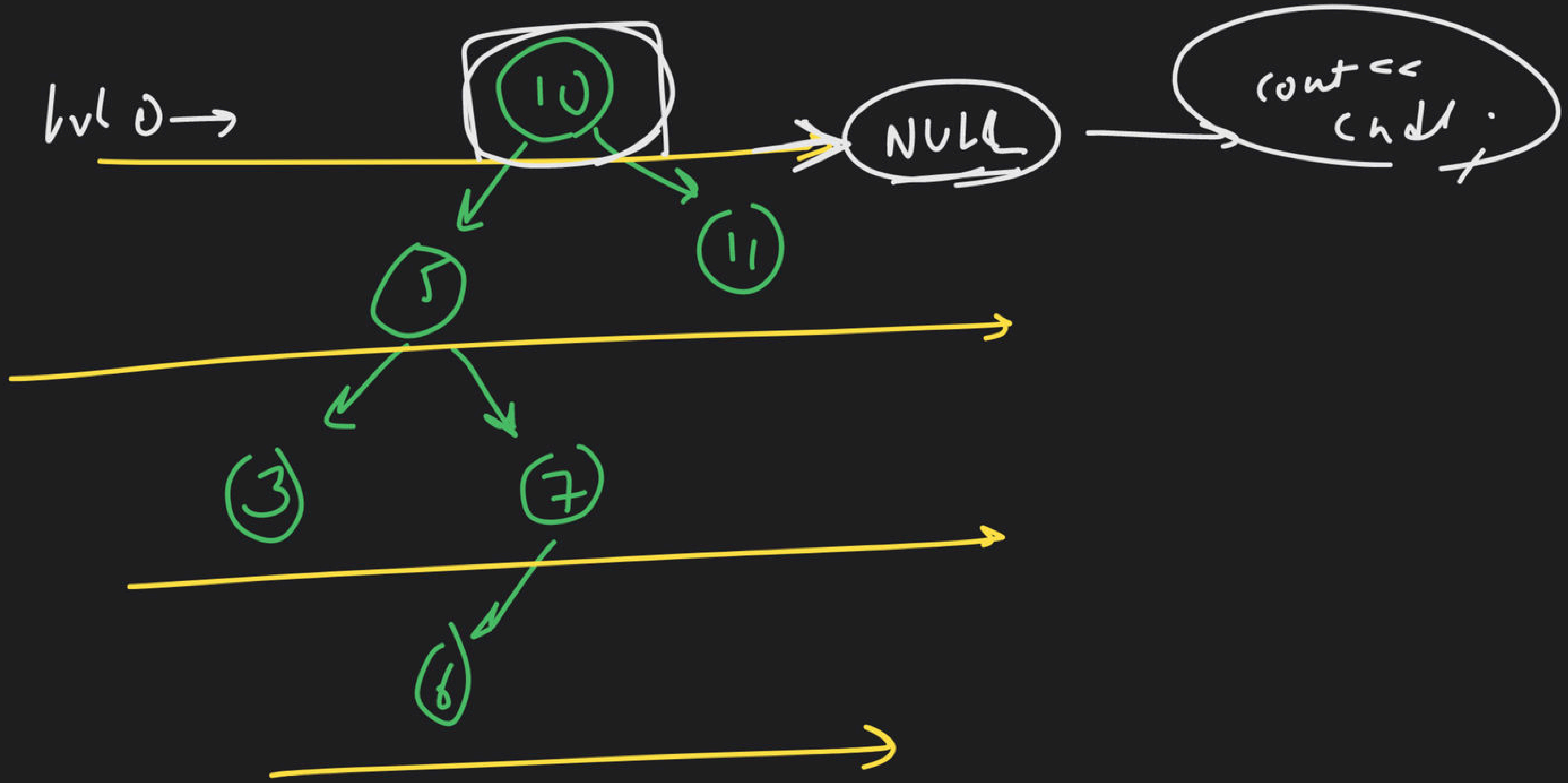
→ 3 11

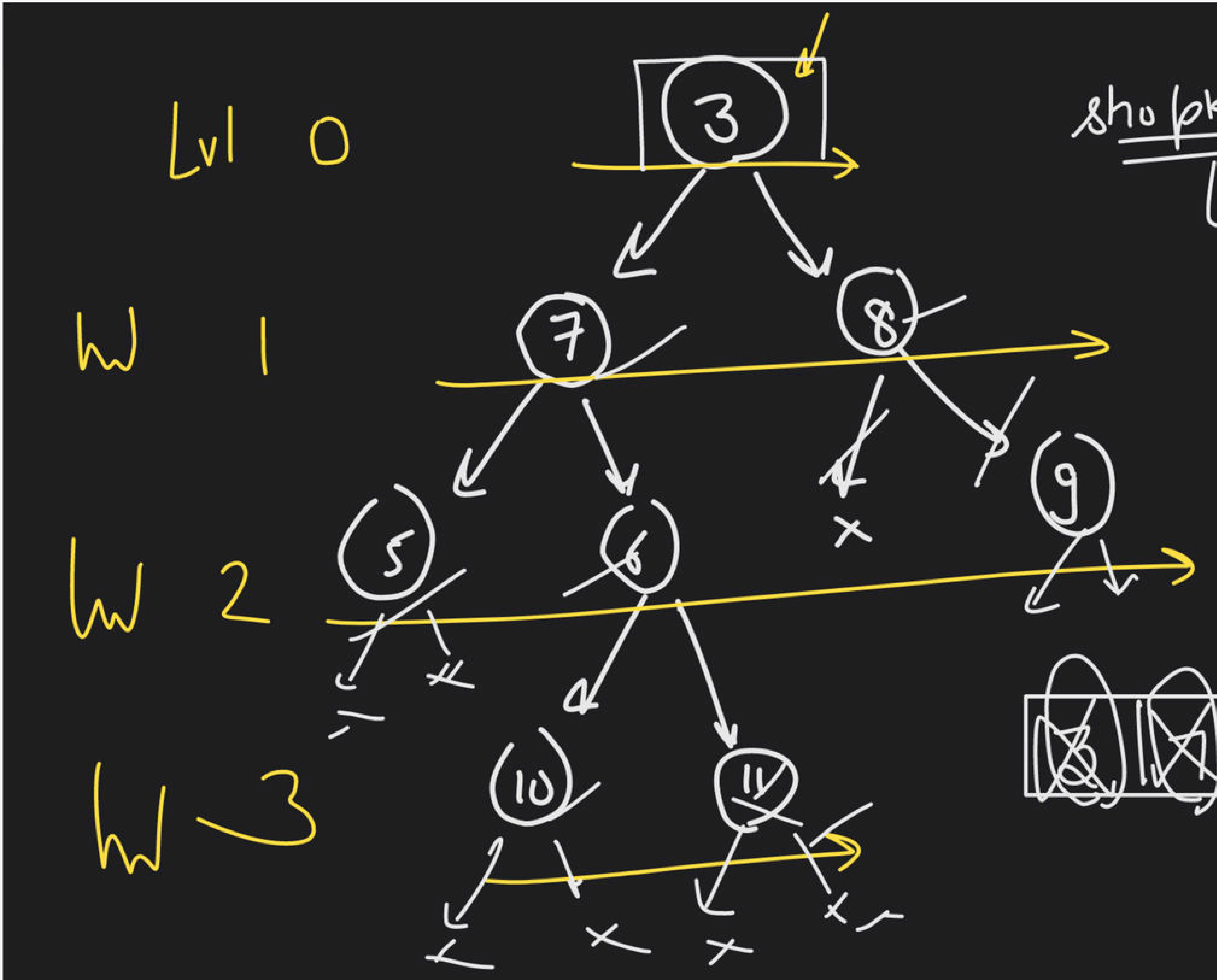
→ 5 7

→ 6

Queue → ?

FIFO

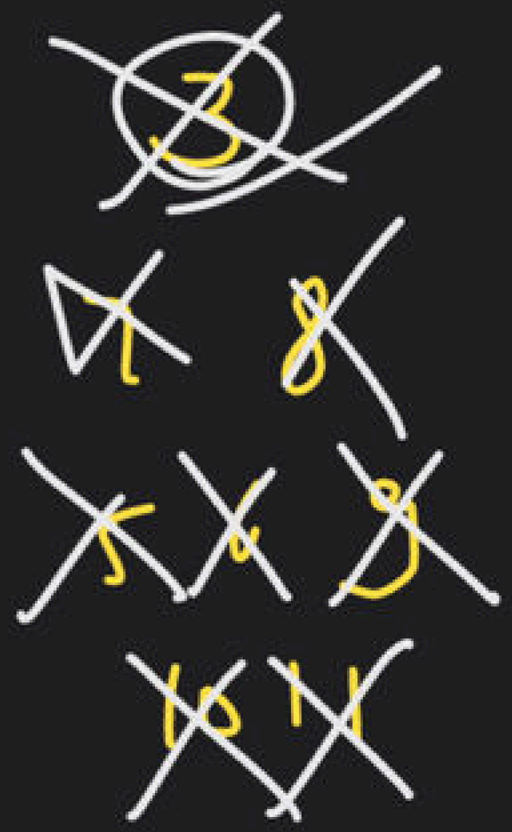

lvl 0 →
10
5
11
3
7
6
NULL
cout << endl;

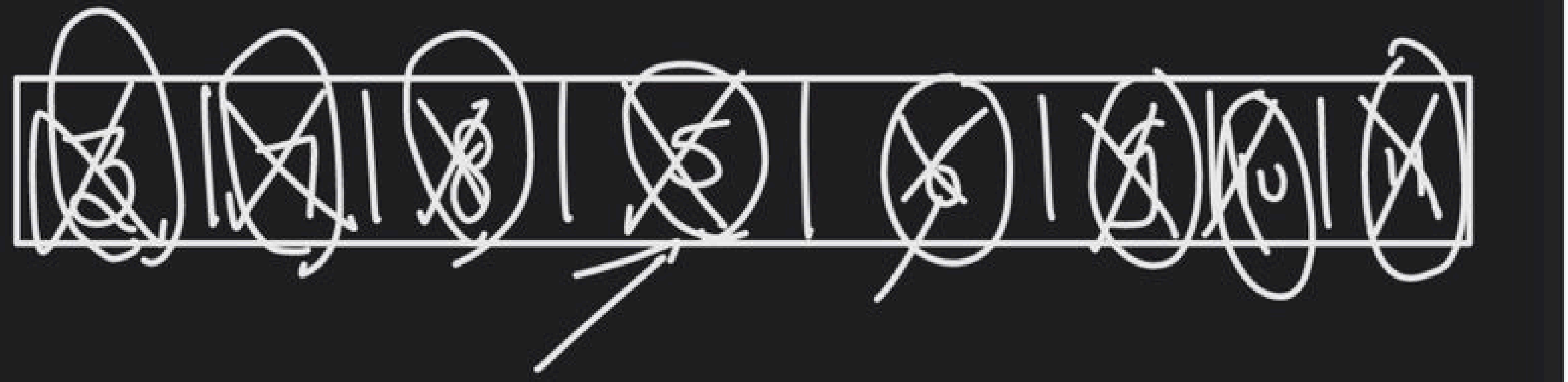


shopkeeper

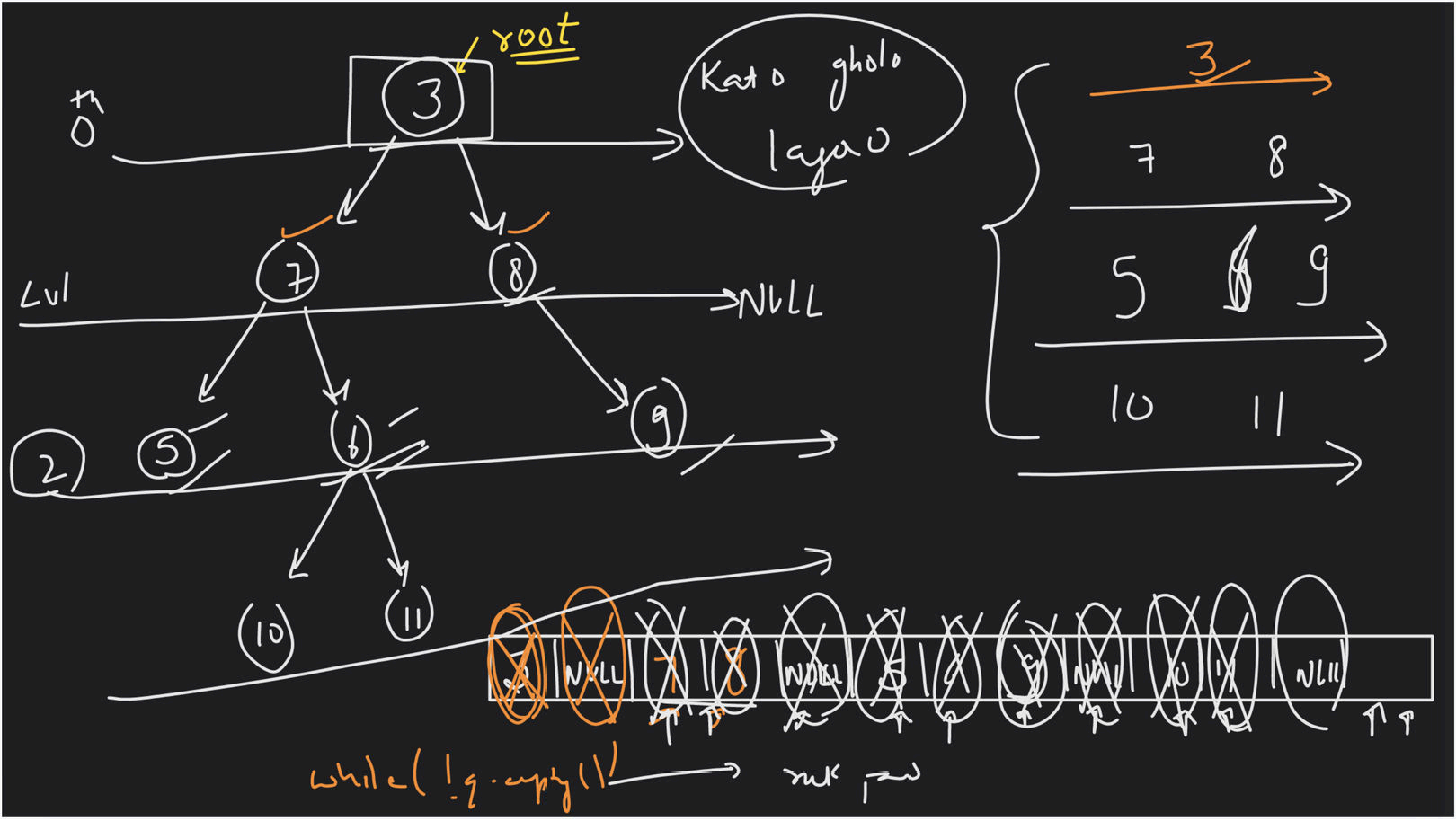
Customer

Satisfaction





q.empty() → stop


root
0th
3
Kat o gholo layao
Lvl
7
8
NULL
2
5
6
9
10
11
3
7 8
5 6 9
10 11
3
NULL
7
8
NULL
5
6
9
NULL
10
11
NULL
while(!q.empty())

Maths → 100/100

MS

Math → Boards

1 hr 40 min

LB → Math

12th → P.B → S → 100

M → 95

LB → 75

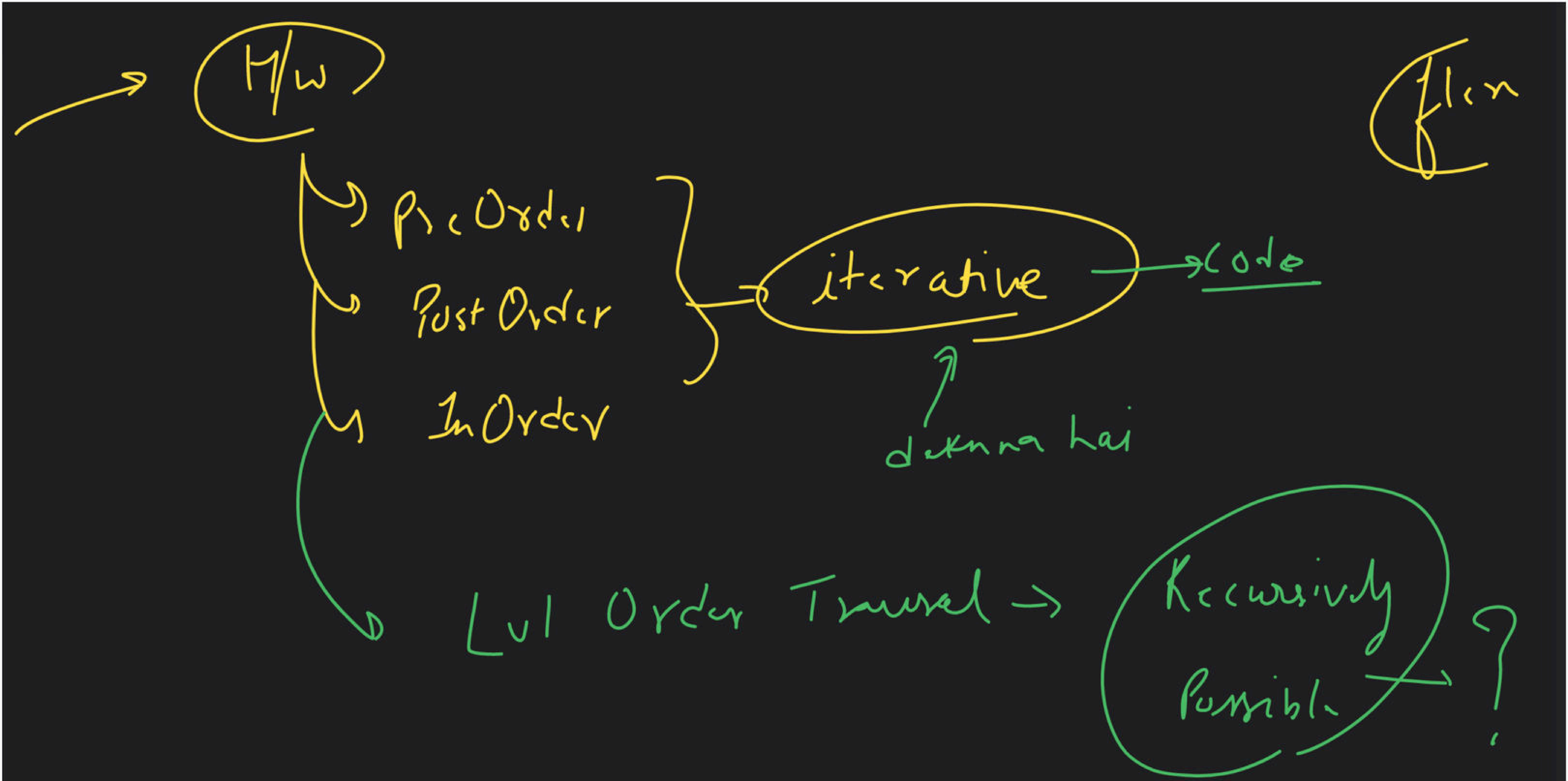
M

7 days

P.B → Lvl 2 → SM → 30 → fail

MS → 95

L.B → 100



H/W

Flex

PreOrder

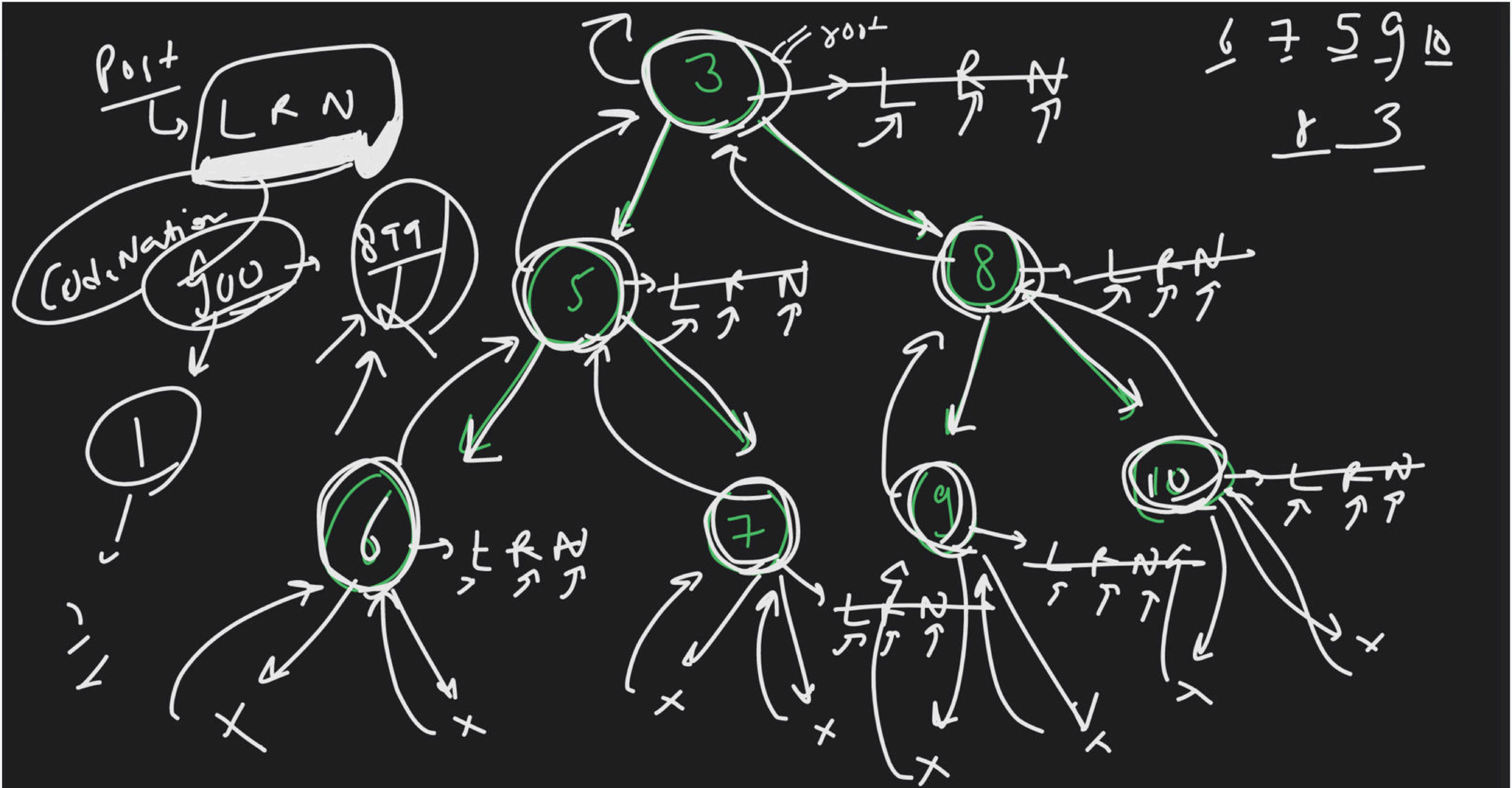
PostOrder

InOrder

iterative → code

dekhna hai

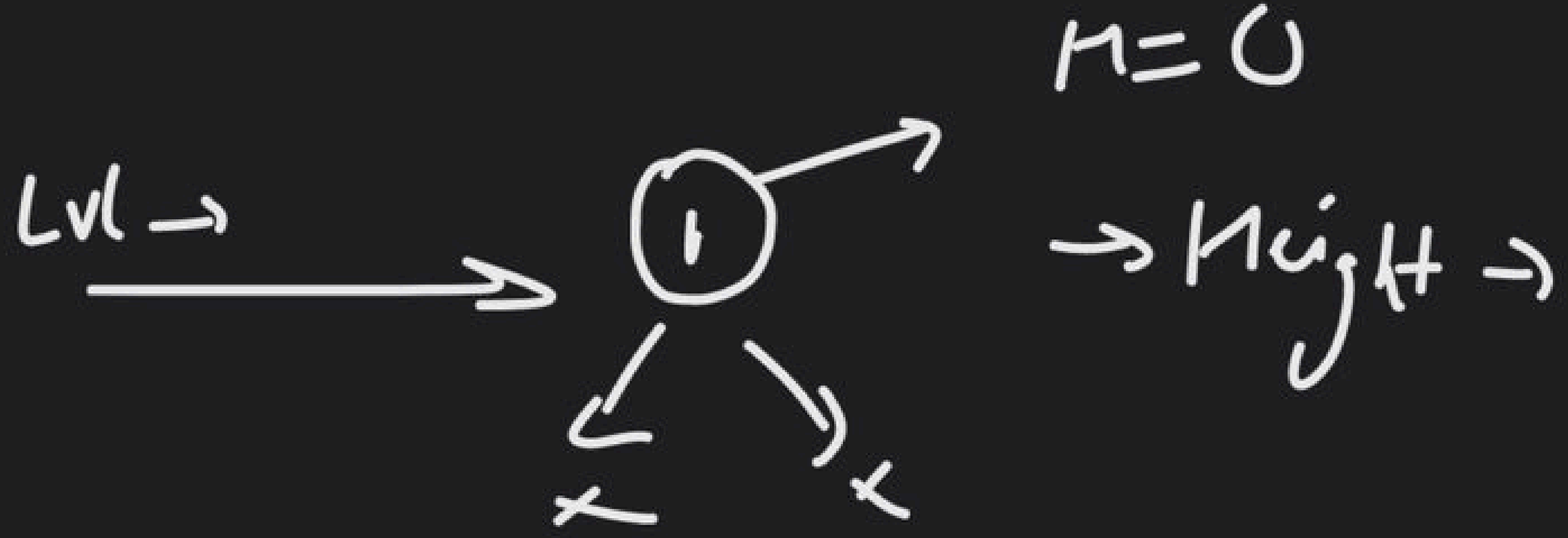
Lvl Order Traversal → Recursivly Possible → ?

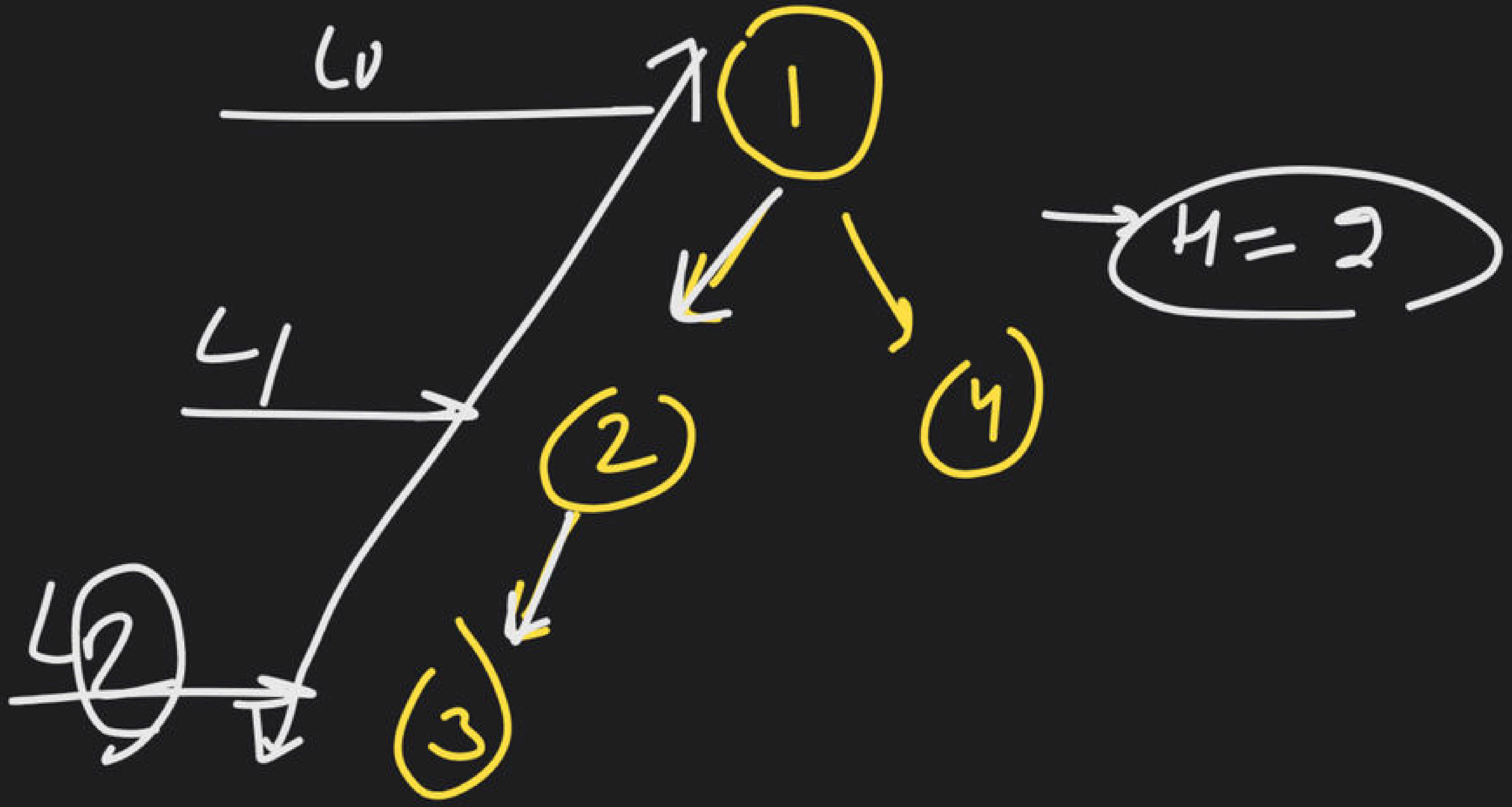

Post
LRN
Codination
f00
1
899
root
3
5
8
6
7
9
10
L R N
L R N
L R N
L R N
L R N
L R N
L R N
6 7 5 9 10
8 3

→ Height of tree

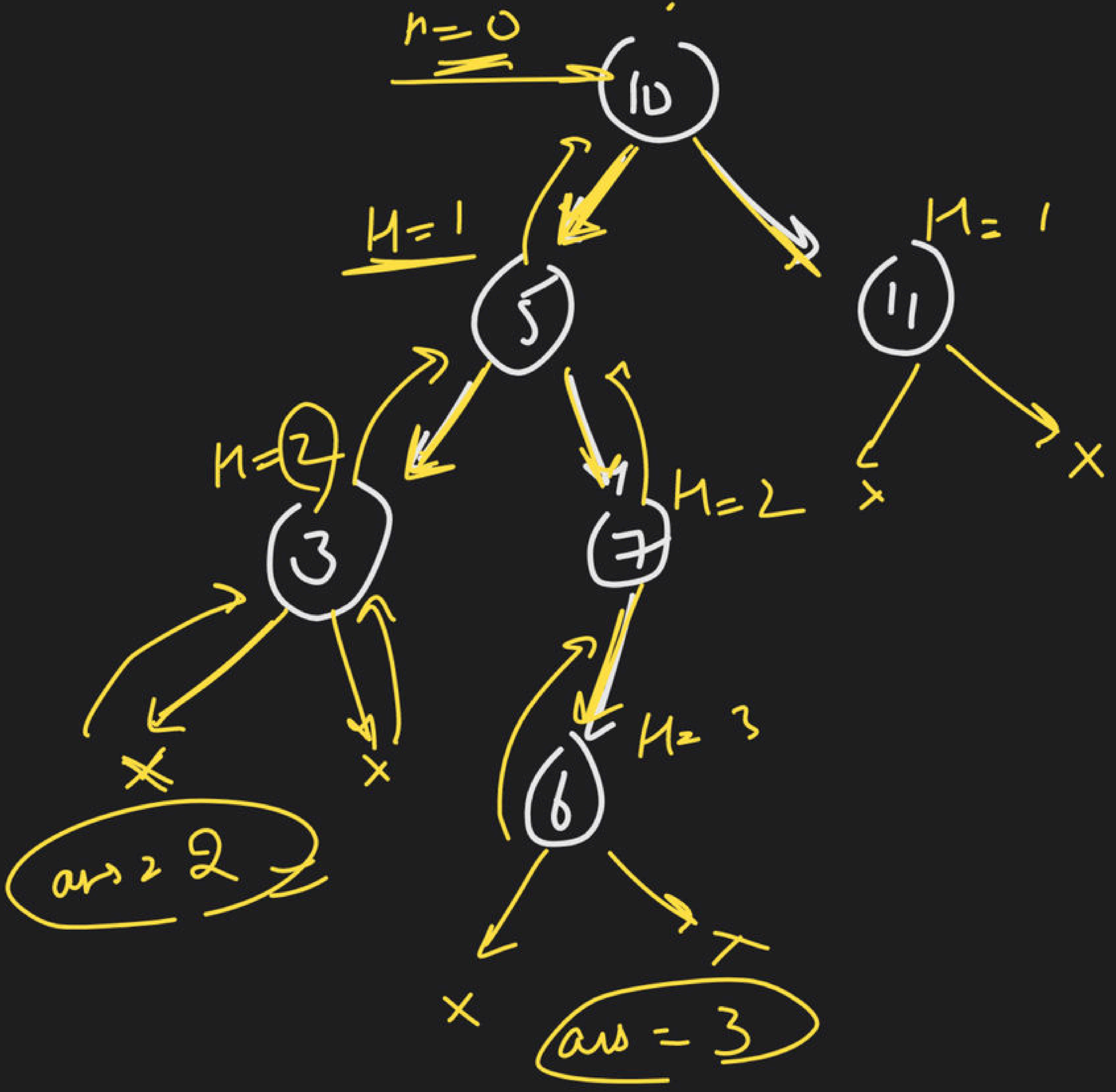
Max Depth

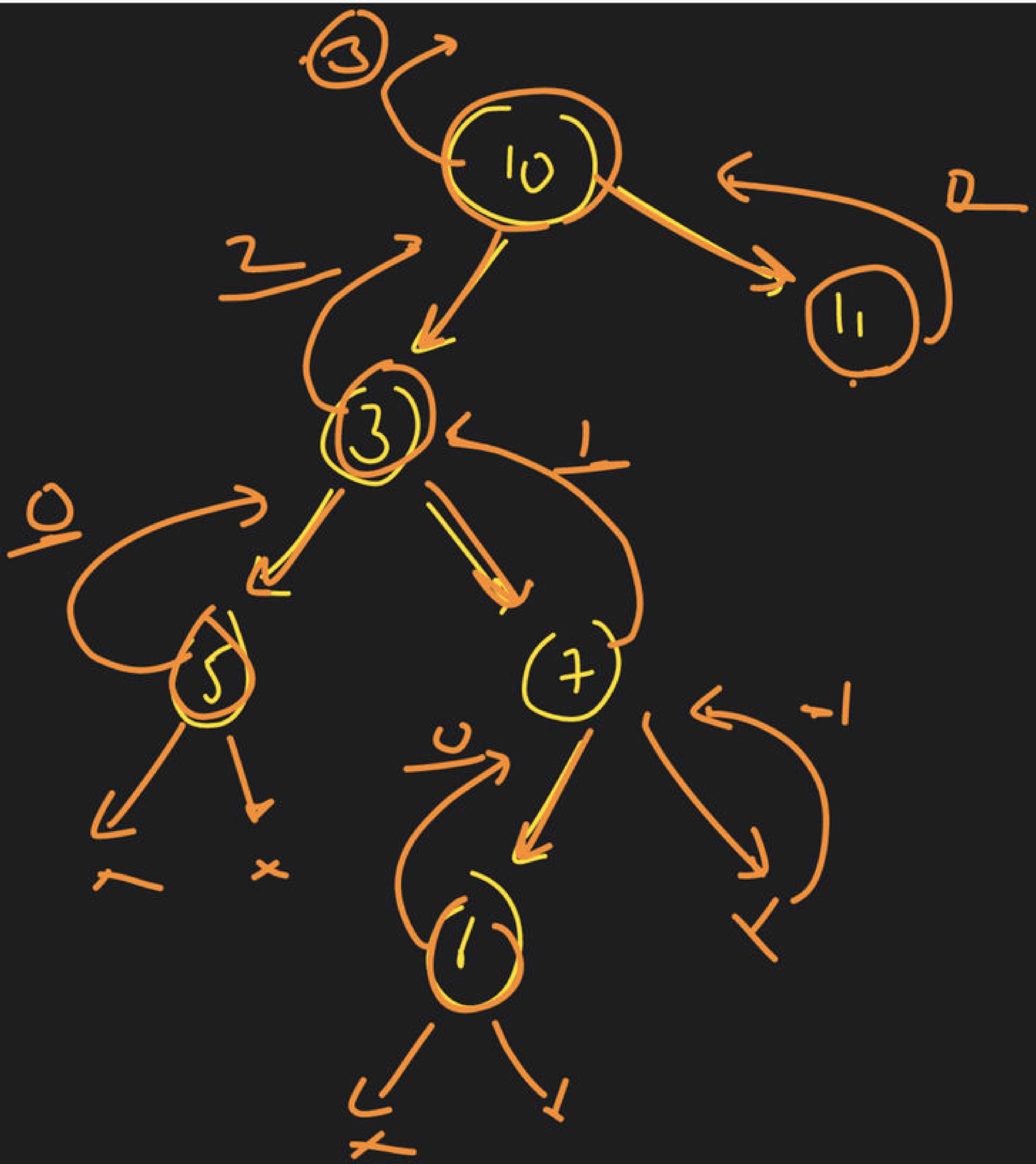
Height → ?



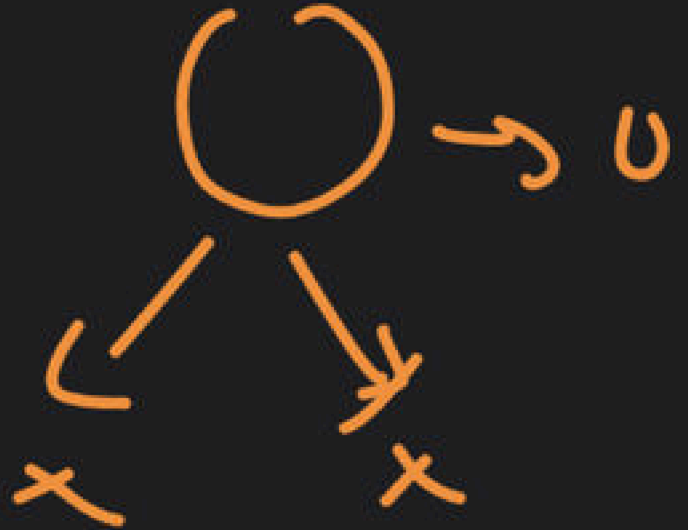


ans = INT_MIN

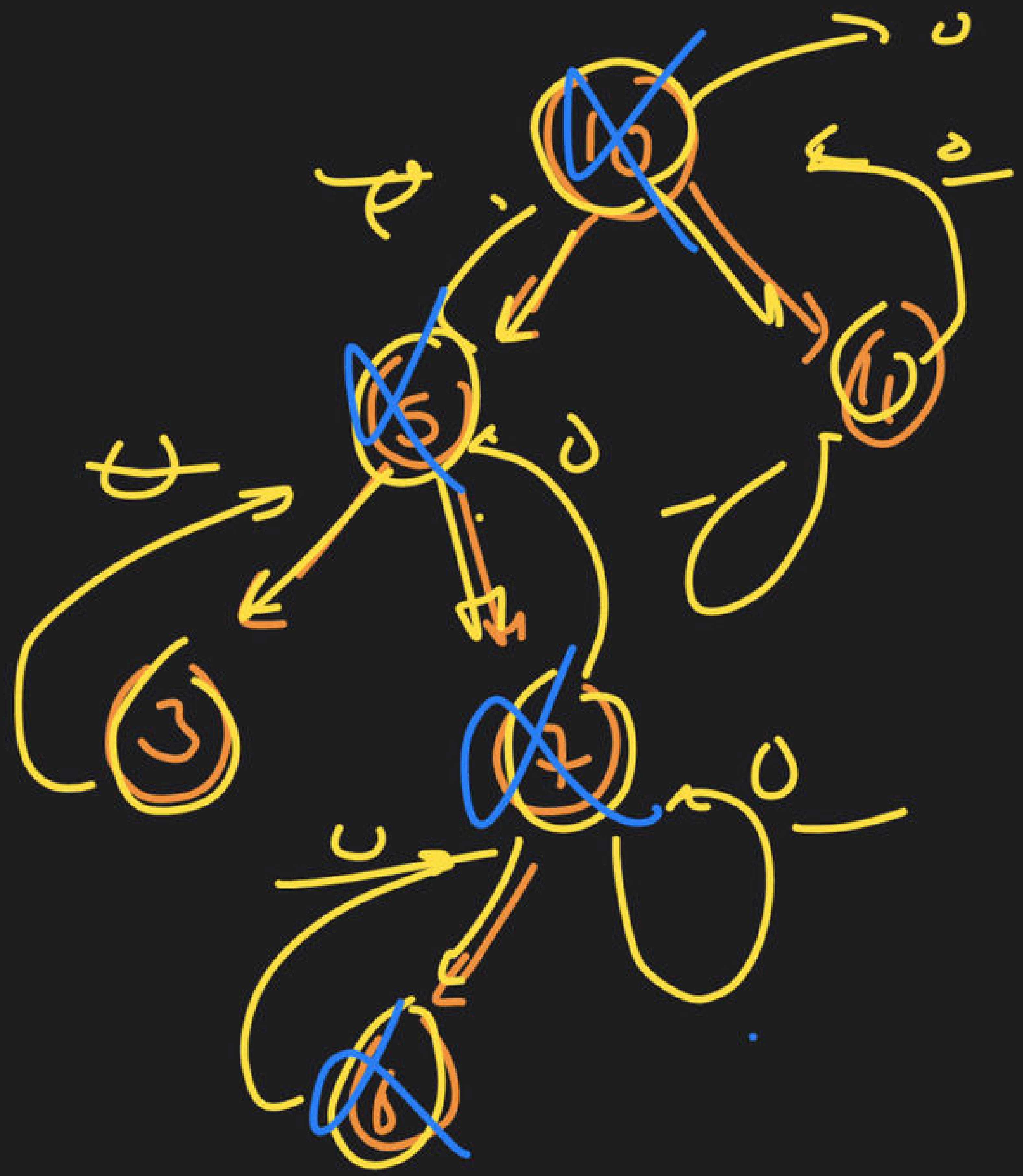


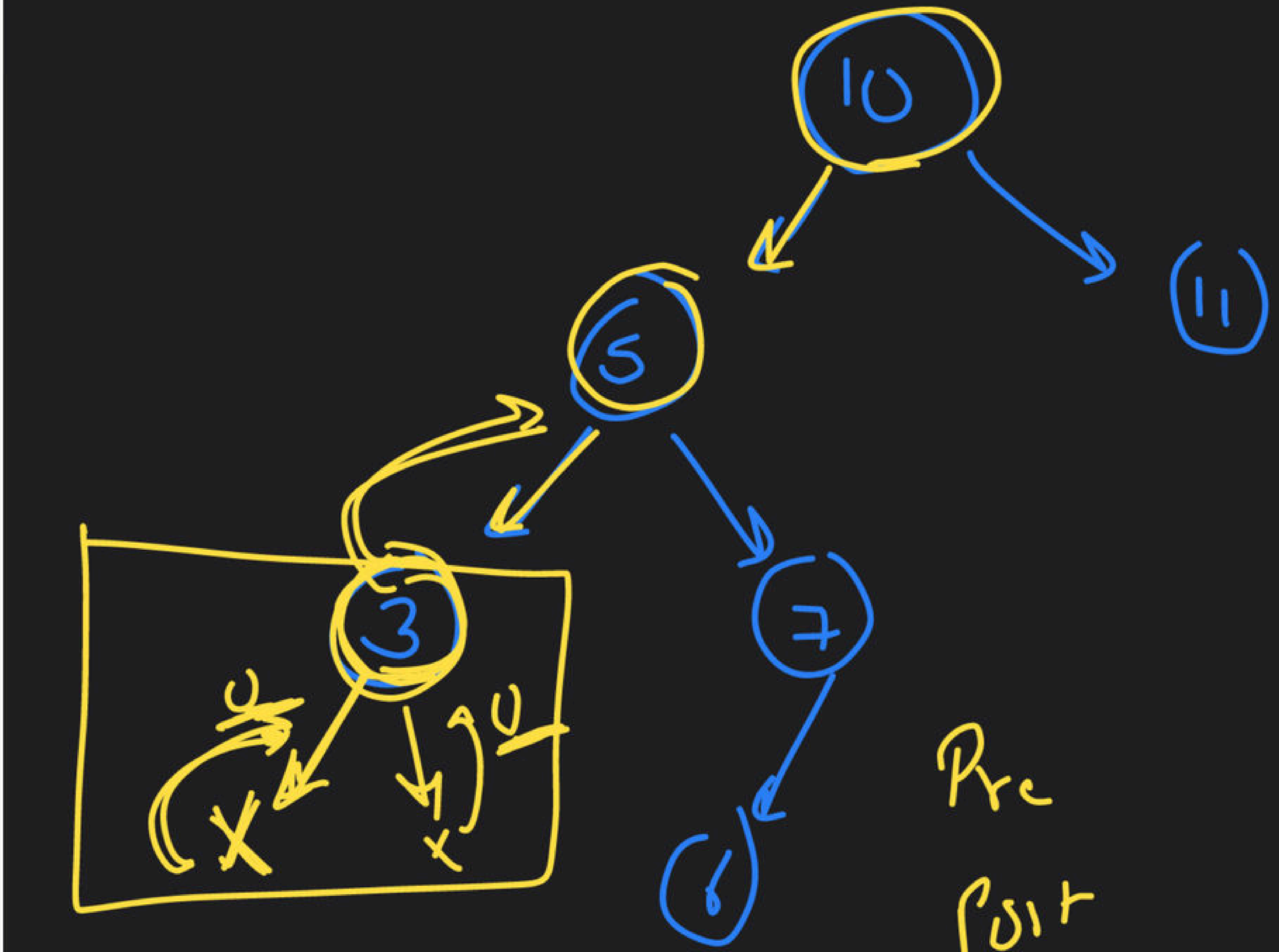


3
10
0
2
11
3
1
0
5
7
-1
0
1
x
x



0
x
x

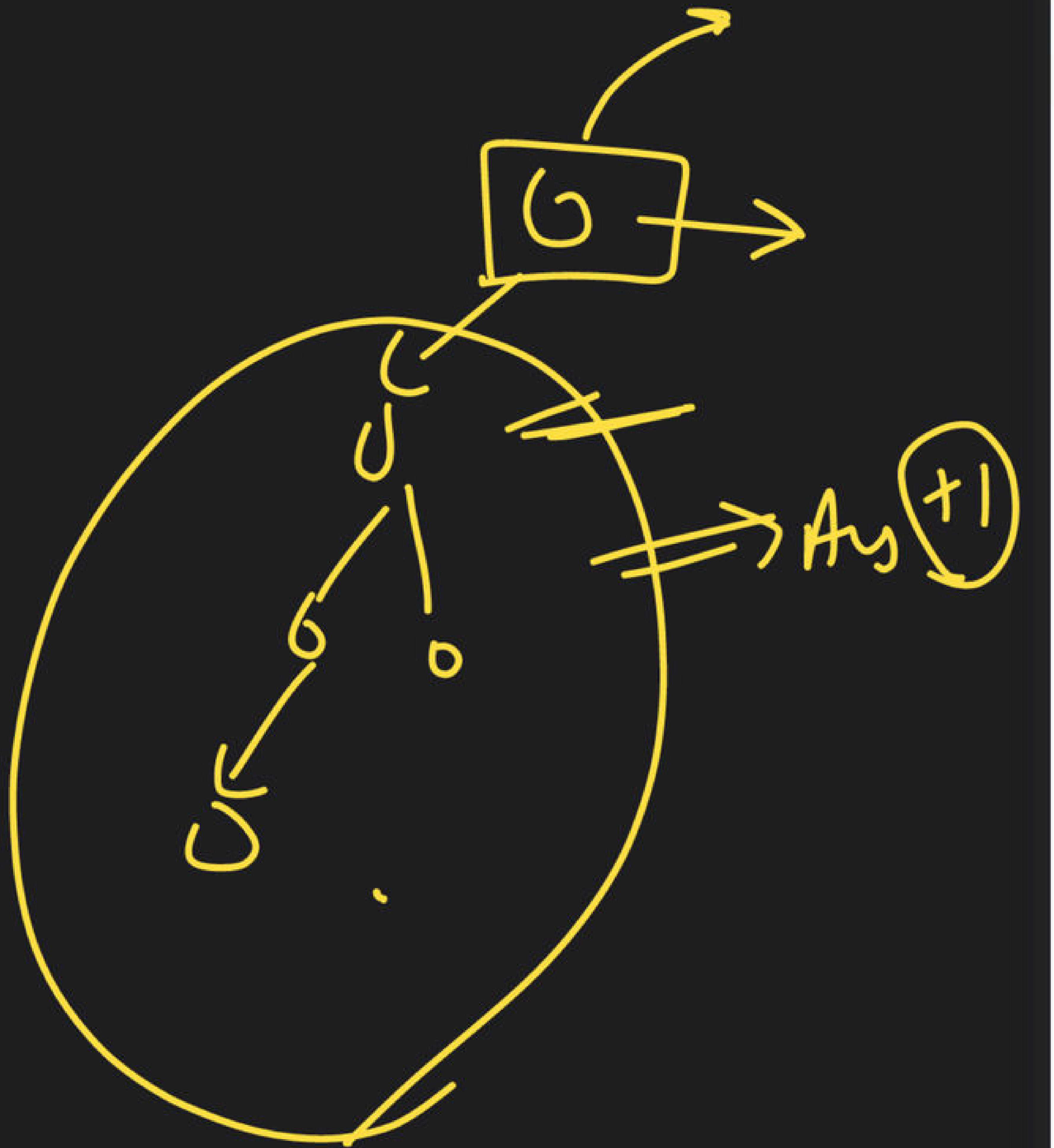

10
5
11
3
7
6
0
0
0
0
0
0
-1



Pre

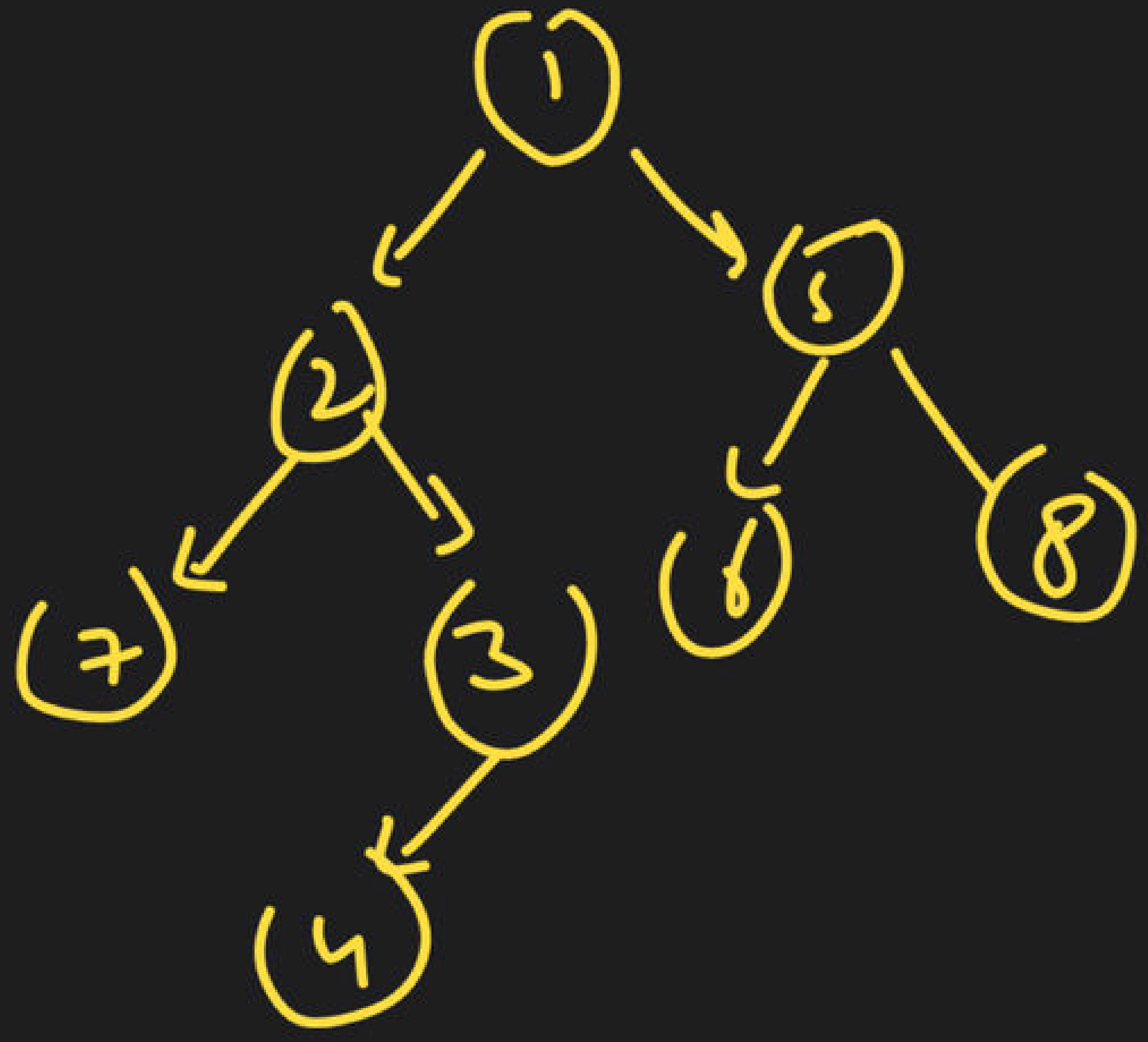
Post

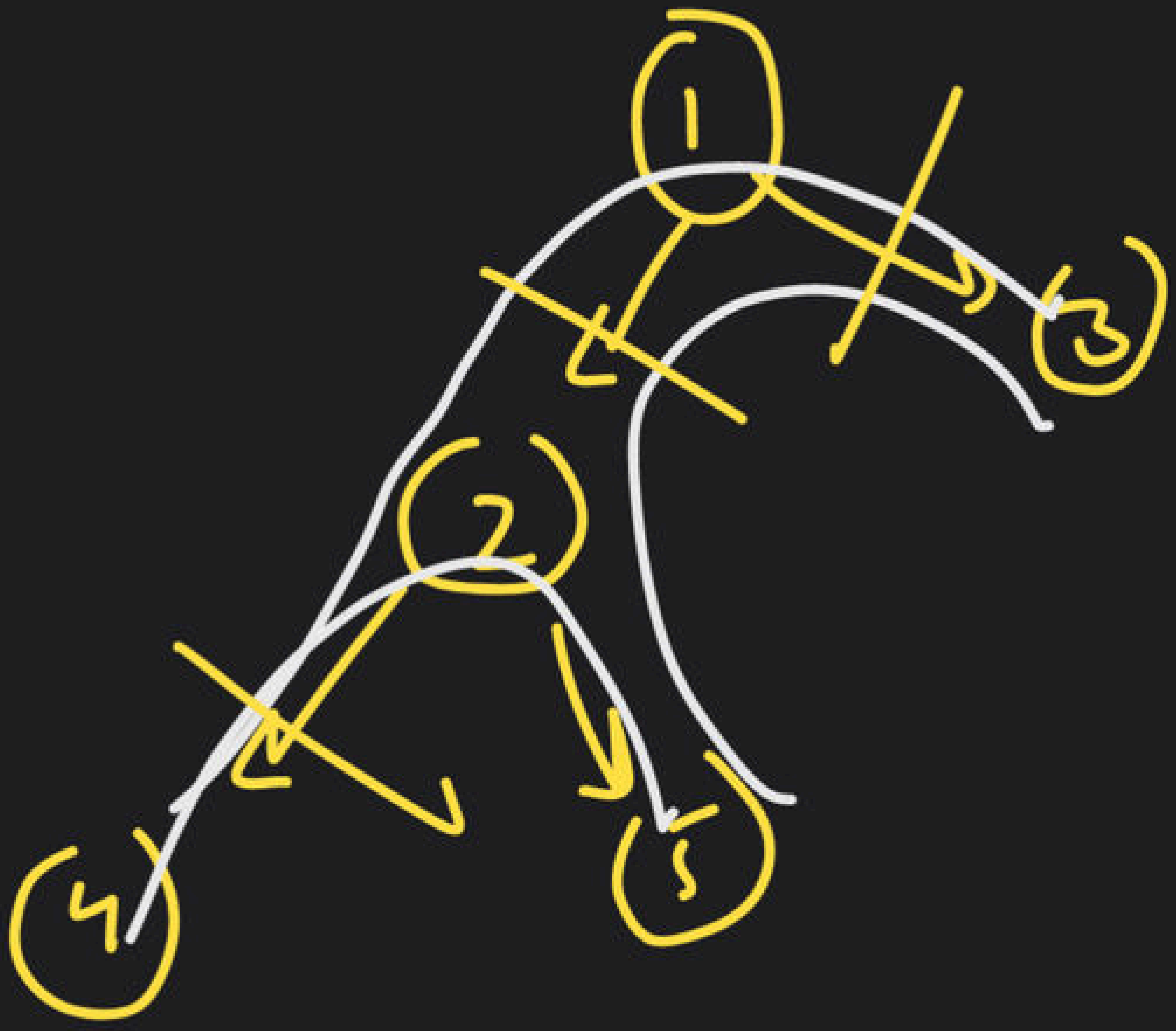
Inord

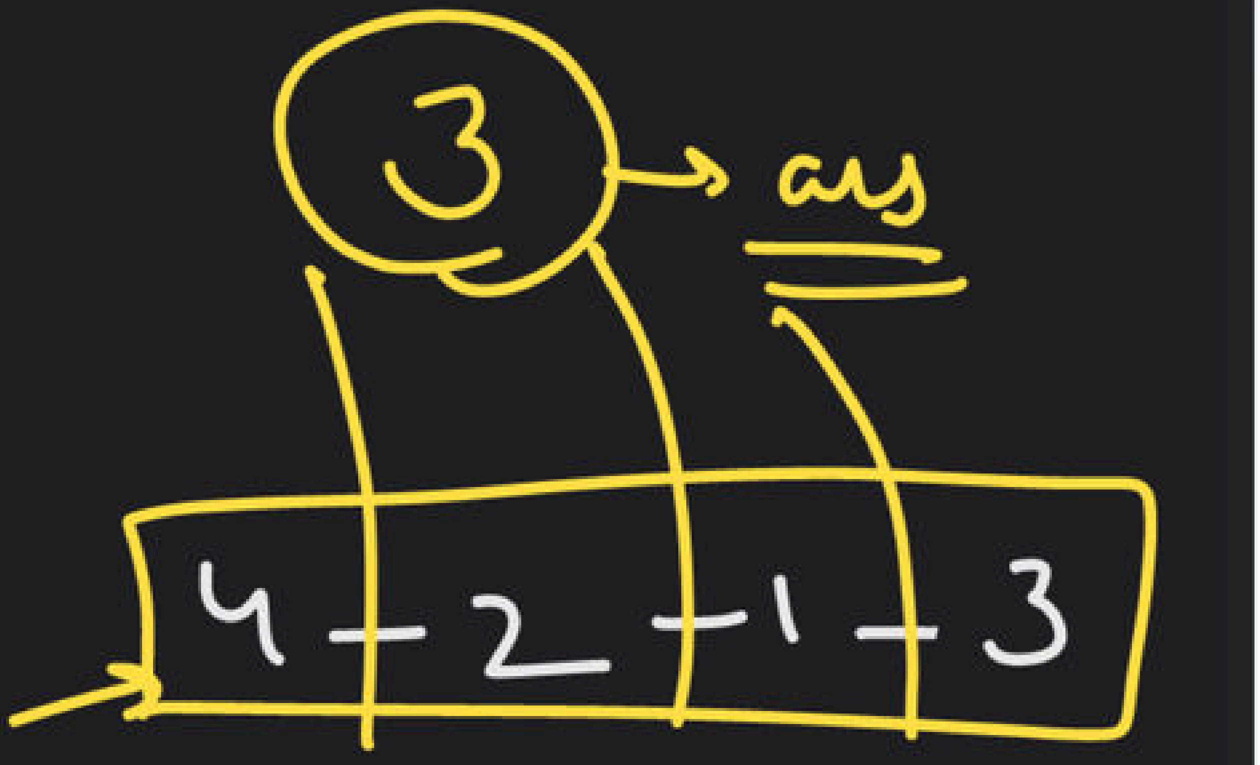


Diameter of tree :-

Diameter :-





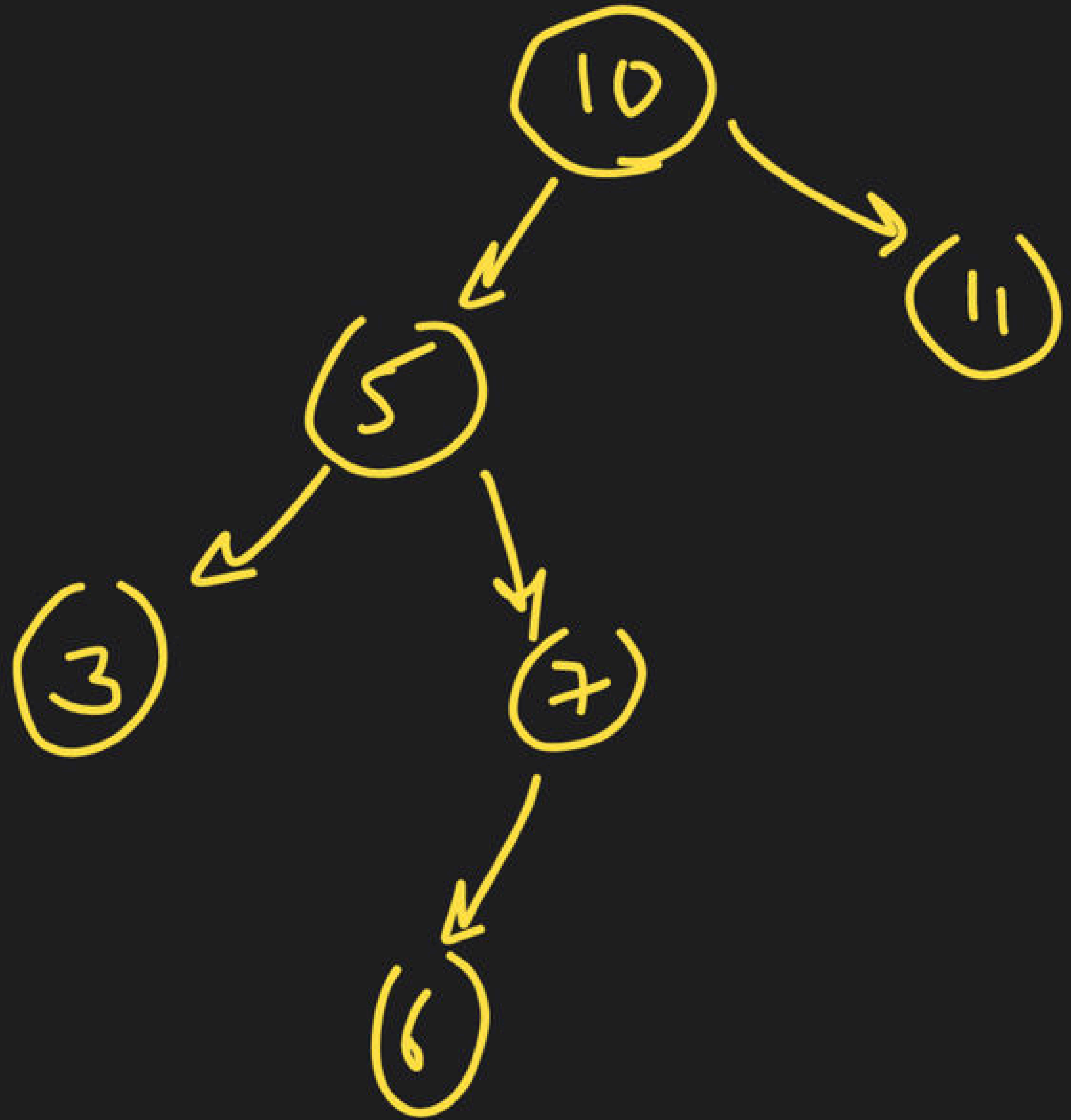


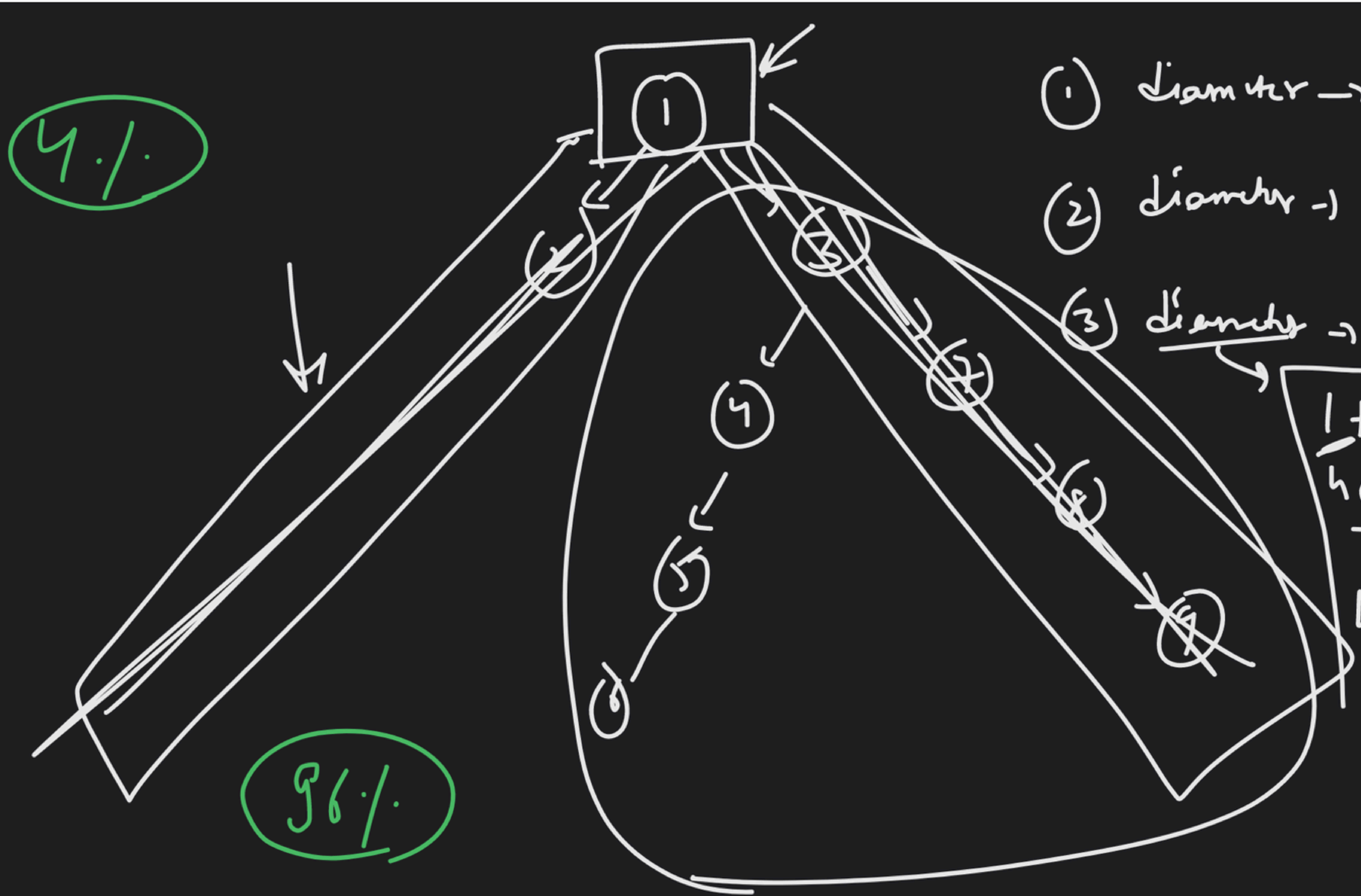
4 - 2 - 1 - 3

4 - 2 - 5

5 - 2 - 1 - 3

Logic





1. diameter → left
2. diameter → right
3. diameter → root

$$\boxed{1 + h(\text{left}) + h(\text{right})}$$

KHUSHI