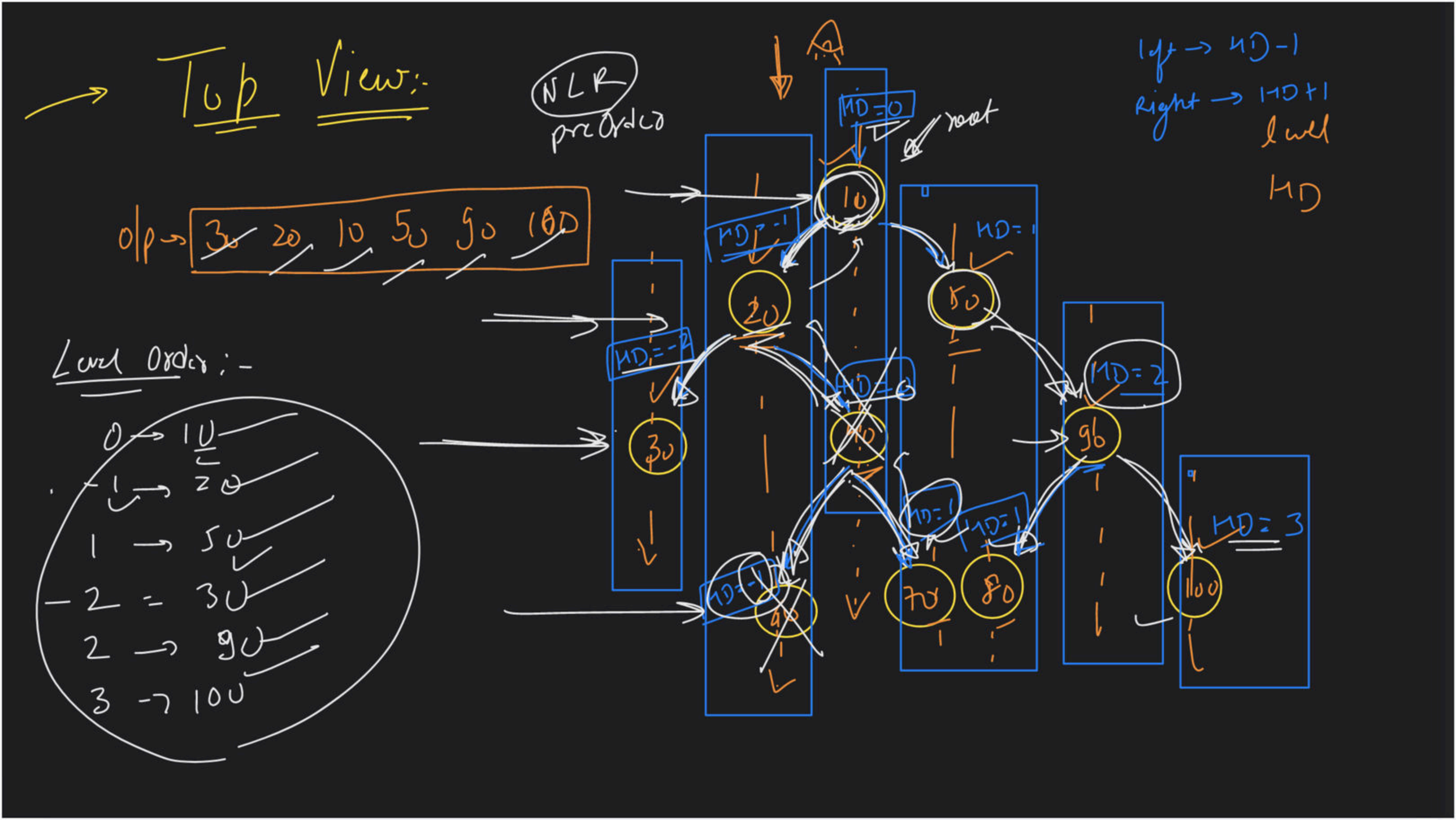
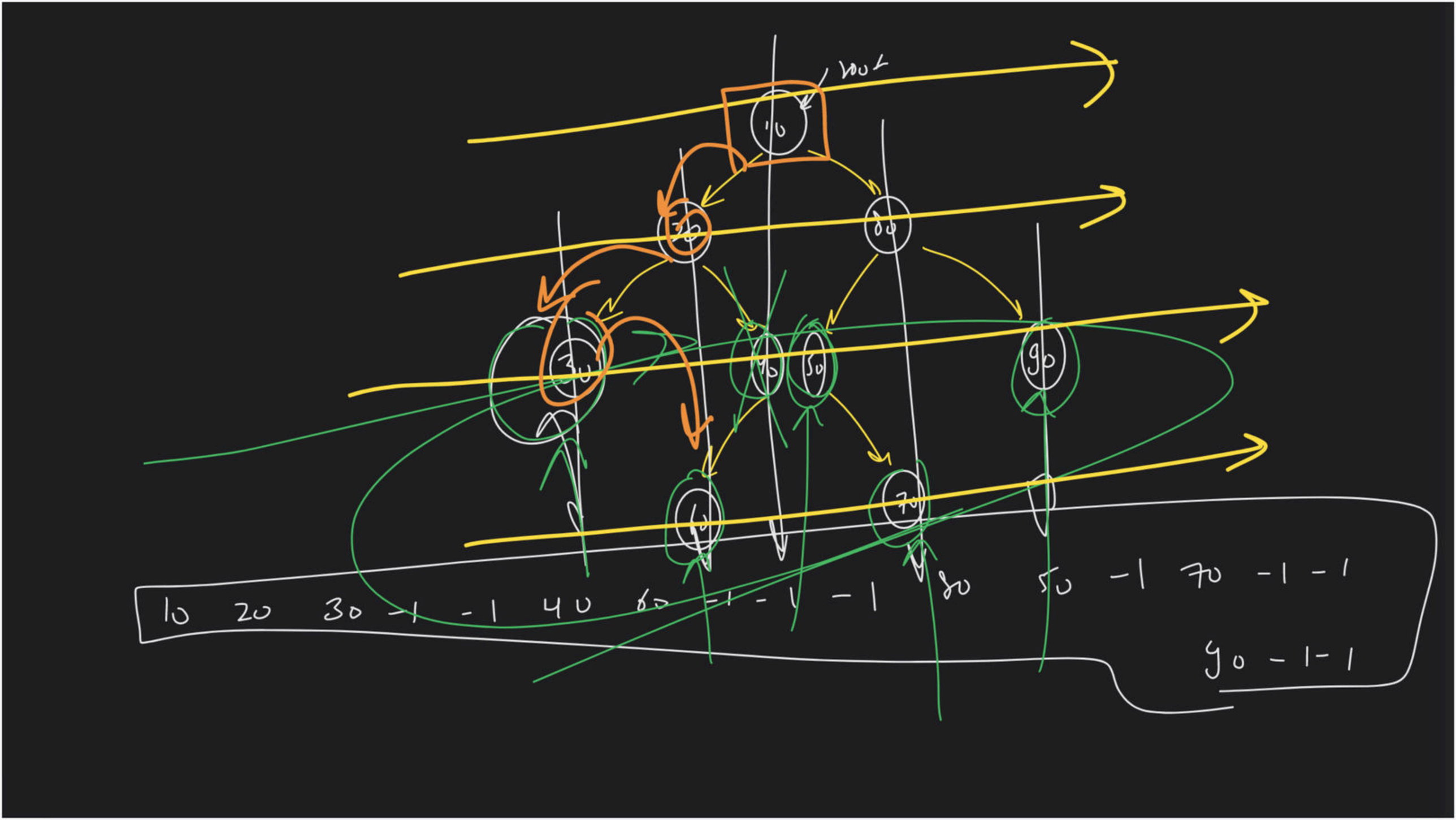
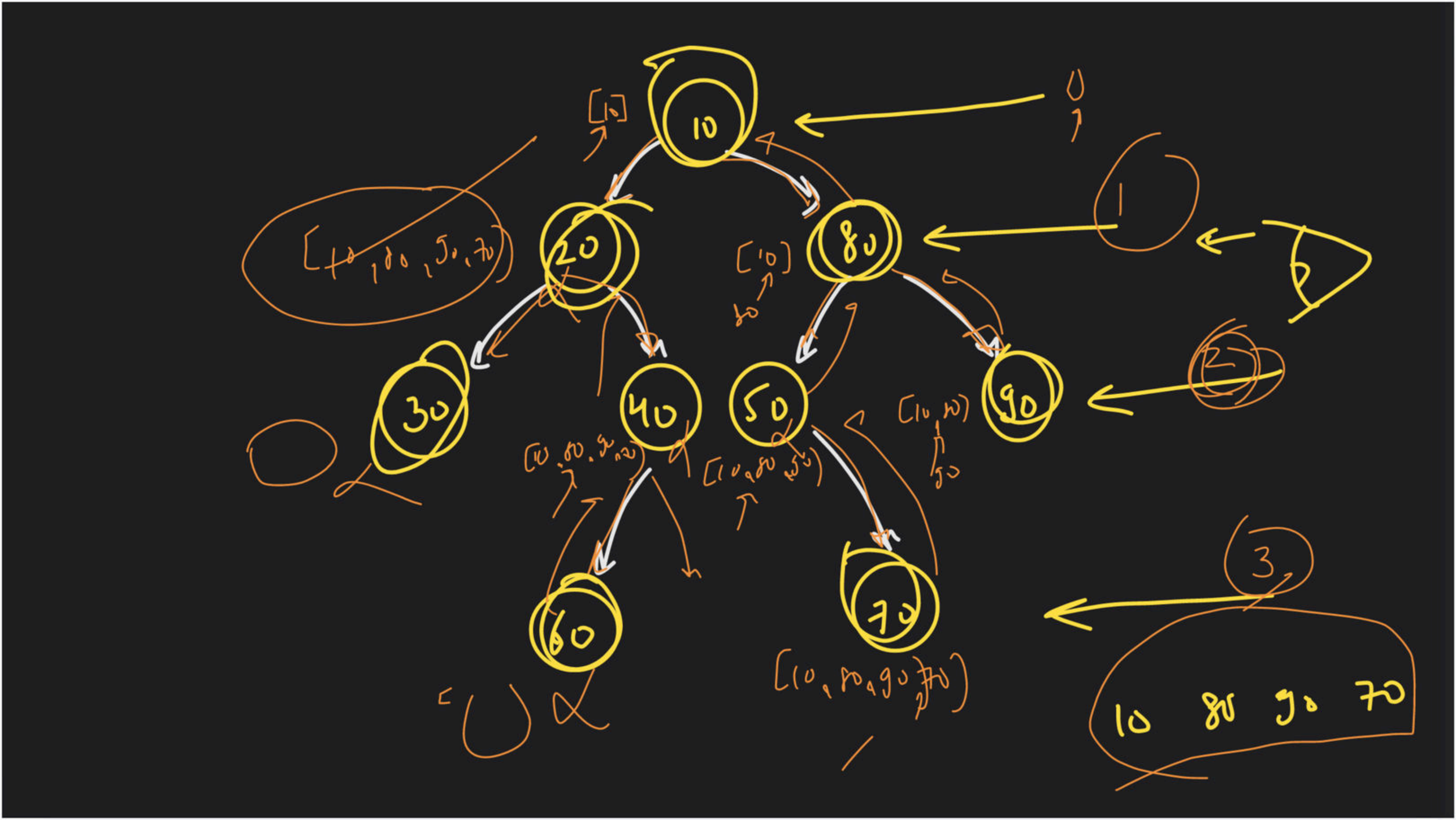
Trees Class - 5

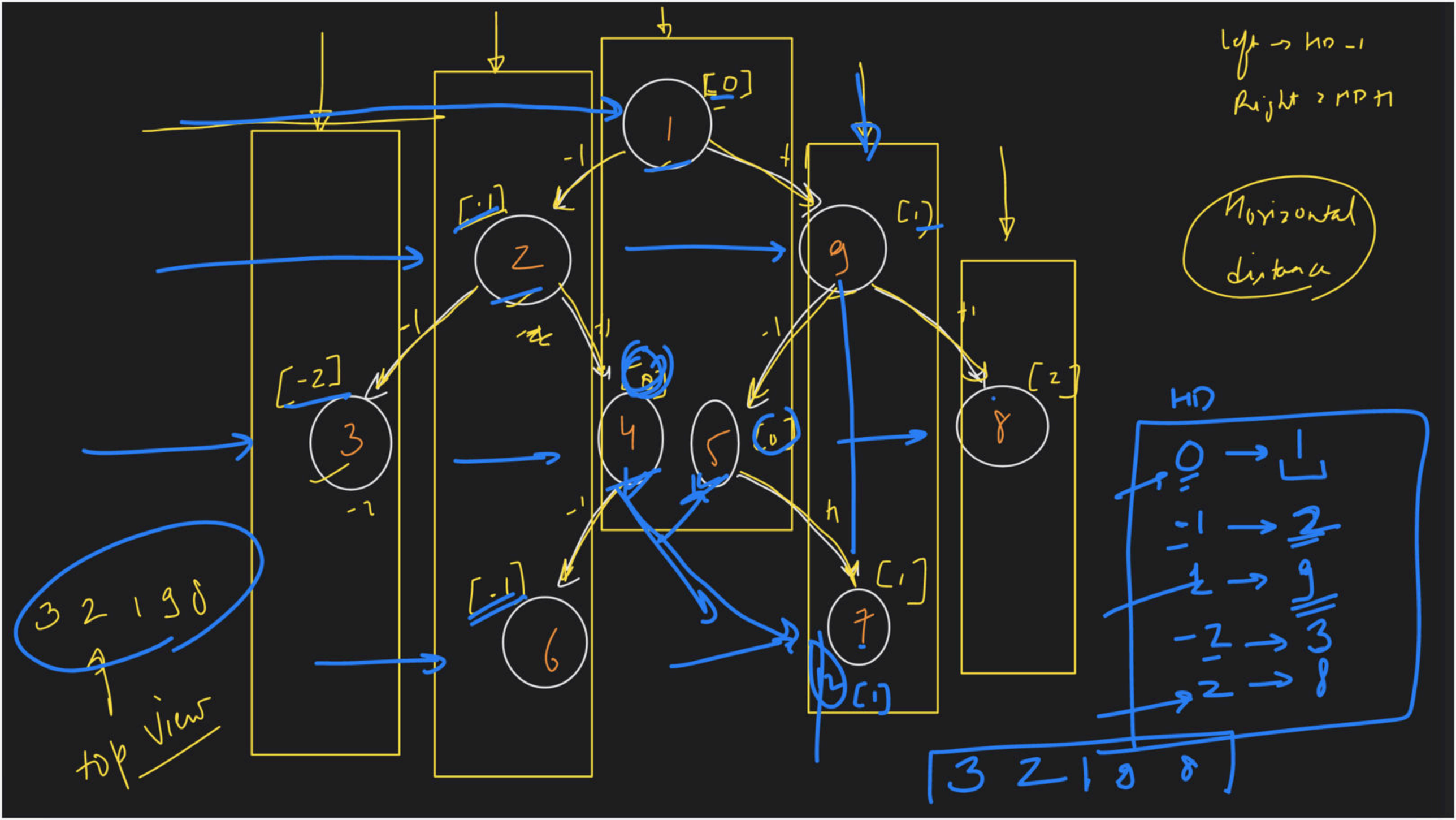
Special class

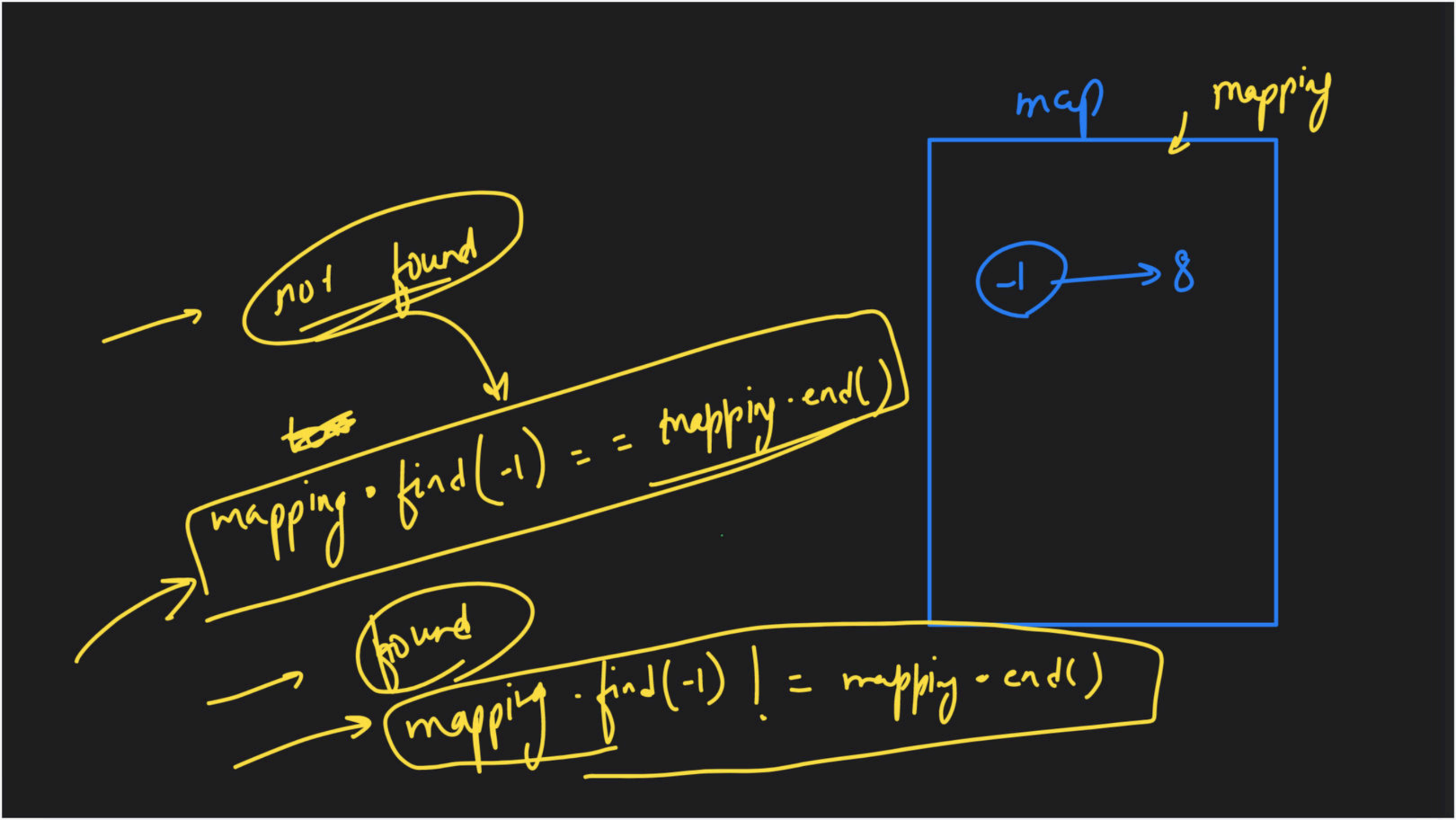
Love Babbar • May 13, 2023


Top View:-
NLR
preOrder
o/p → 30 20 10 50 90 100
Level Order :-
0 → 10
1 → 20
1 → 50
-2 = 30
2 → 90
3 → 100
HD=0
root
HD=-1
HD=1
HD=-2
HD=0
HD=2
HD=-1
HD=1
HD=3
left → HD-1
Right → HD+1
level
HD


10 20 30 -1 -1 40 60 -1 -1 -1 80 50 -1 70 -1 -1
90 -1 -1


10
20
80
30
40
50
90
60
70
1
2
3
10 80 30 70


Left → HD - 1
Right → HD + 1
Horizontal distance
[0]
[-1]
[1]
[-2]
[2]
[-1]
[1]
[0]
1
2
9
3
8
4
5
6
7
-1
+1
-1
+1
-1
+1
-2
-1
+1
+1
3 2 1 9 0
top View
HD
0 → 1
-1 → 2
1 → 9
-2 → 3
2 → 8
3 2 1 9 0 8


map
mapping
not found
mapping.find(-1) == mapping.end()
-1
8
found
mapping.find(-1) != mapping.end()

true / false

Create

search → mappij.find(-1) → exist → != mappij.end()

not exist → == mappij.end()

map < int , int > mapping

Key

Value

map < char , int >

< int , bool >

< Node* , int >

< Node* , bool >

insert

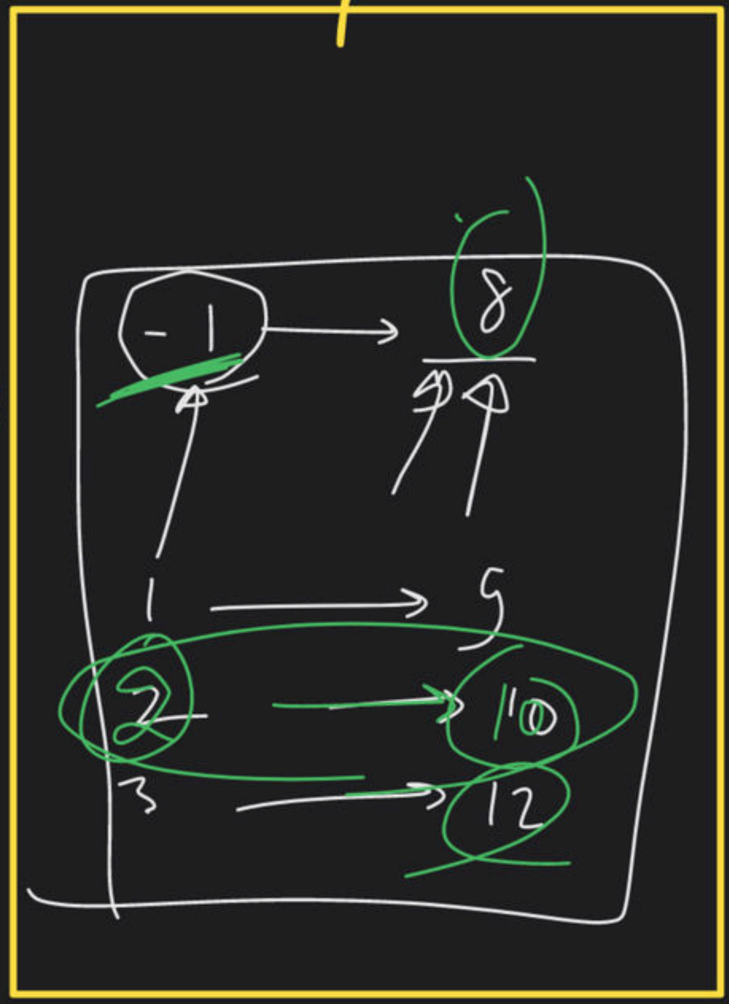
mappij[2] = 10

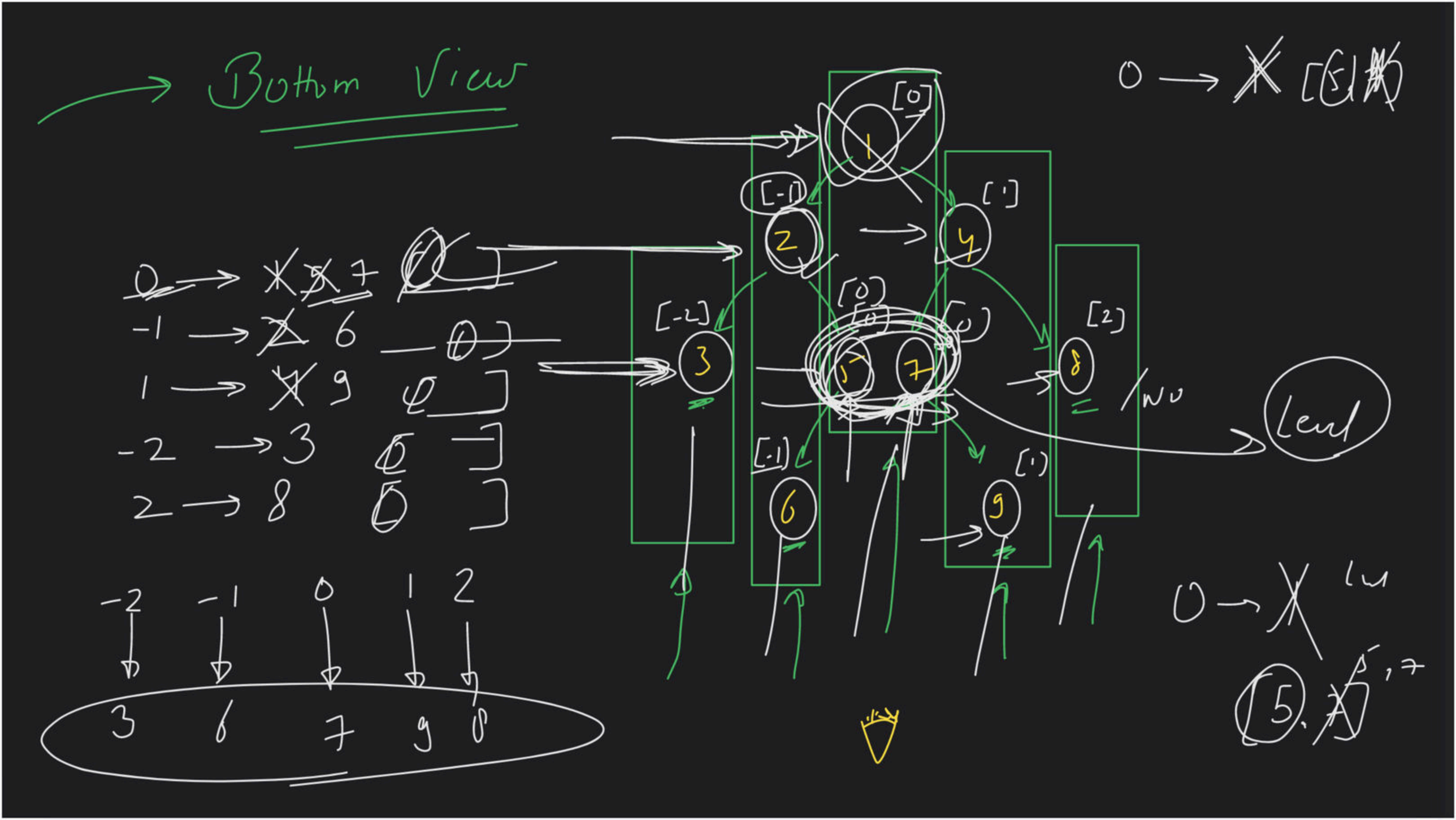
operator[]

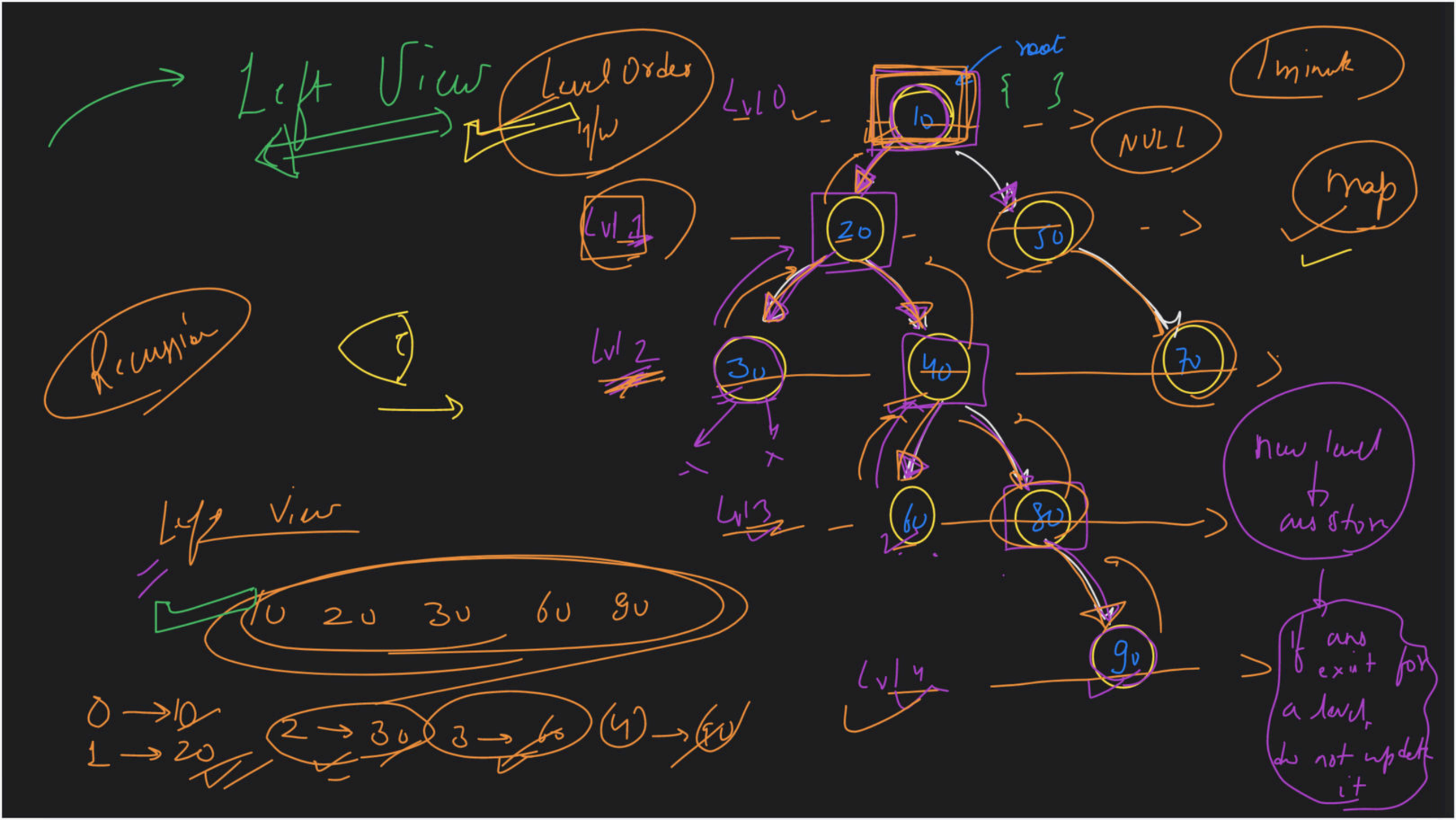
cout << mappij[3]

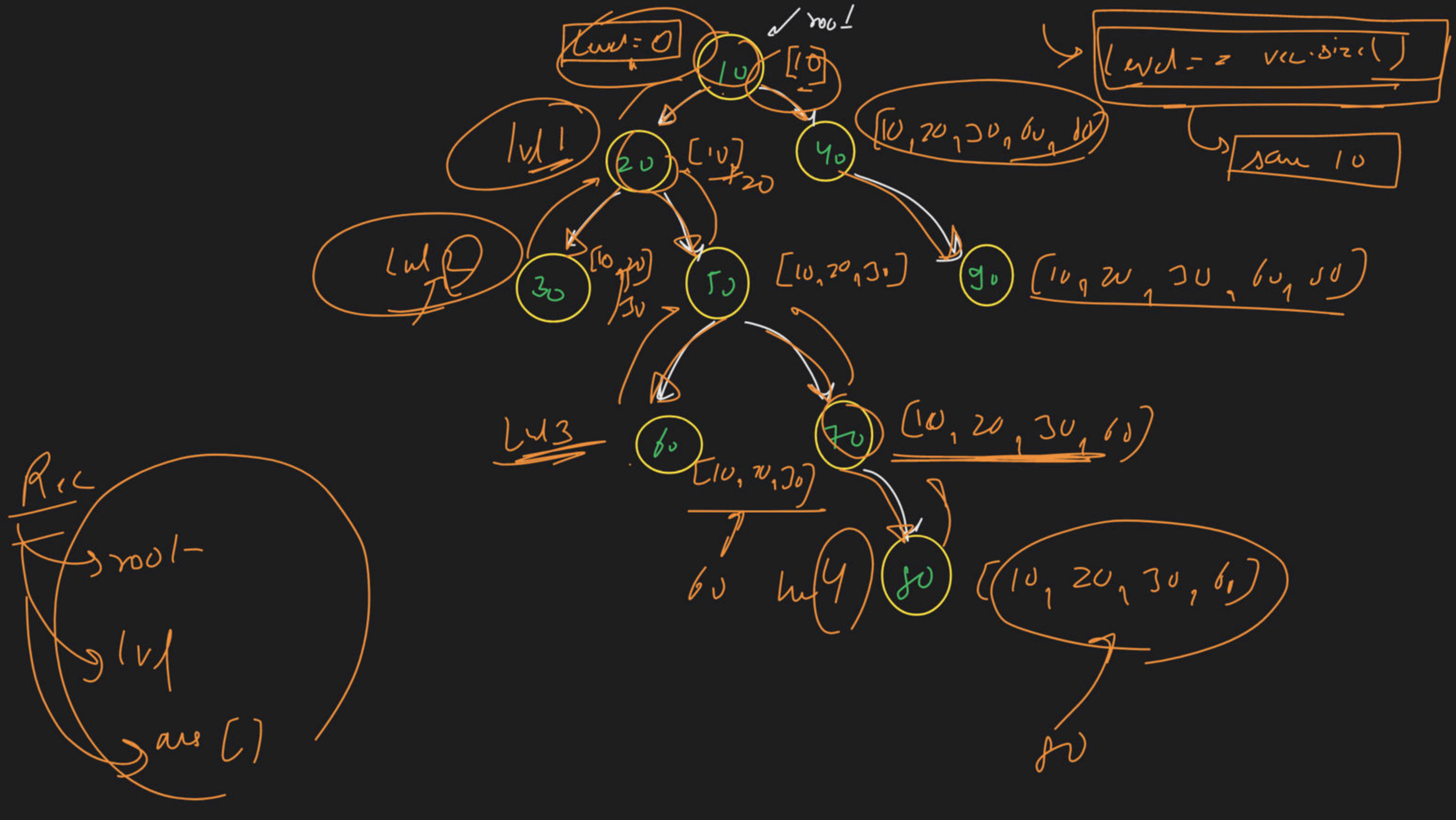
12

twoPNode

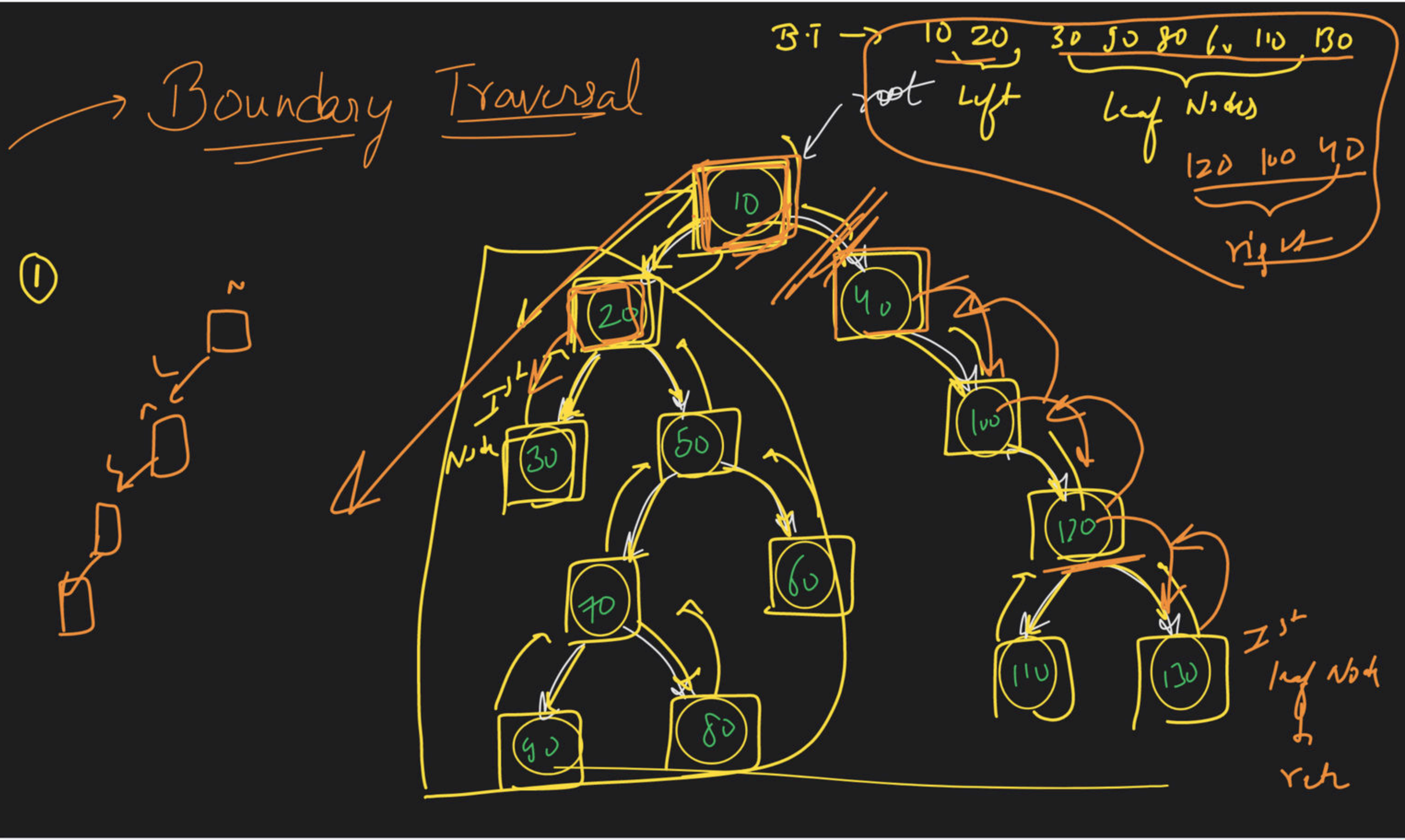



Bottom View
0 → 7
-1 → 6
1 → 9
-2 → 3
2 → 8
[0]
[-1]
[1]
[-2]
[2]
[-1]
[1]
[0]
1
2
4
3
5
7
8
6
9
Level
-2 -1 0 1 2
3 6 7 9 8

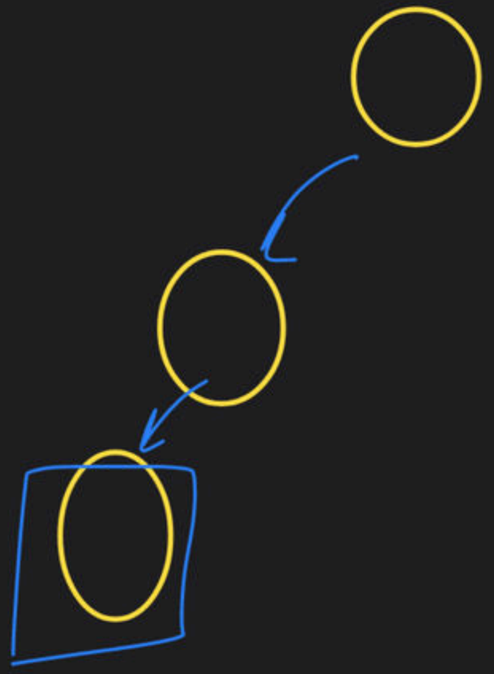

Left View
Level Order
l/w
Lvl 0
root
10
{ }
NULL
Lvl 1
20
50
Lvl 2
30
40
70
Recursion
Lvl 3
60
80
Lvl 4
90
Left View
10 20 30 60 90
0 → 10
1 → 20
2 → 30
3 → 60
(4) → (90)
Hint
Map
new level
→
action
if ans exist for a level, do not update it


root
level = 0
10
[10]
lvl 1
20
[10]
40
[10, 20, 30, 60, 80]
level = 2 vec.size()
sum 10
lvl 2
30
[10, 20]
50
[10, 20, 30]
90
[10, 20, 30, 60, 80]
Lvl 3
60
[10, 20, 30]
70
[10, 20, 30, 60]
60
lvl 4
80
[10, 20, 30, 60]
80
Rec
root-
lvl
ans []

Boundary Traversal



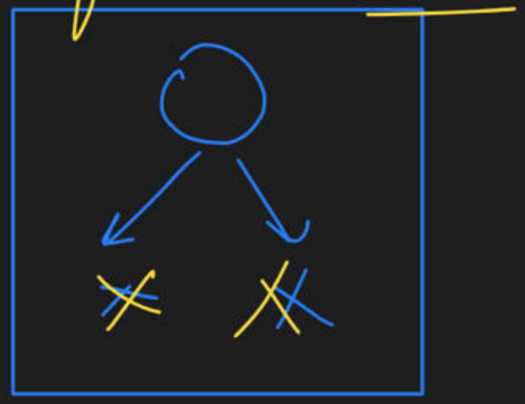
Left Boundary



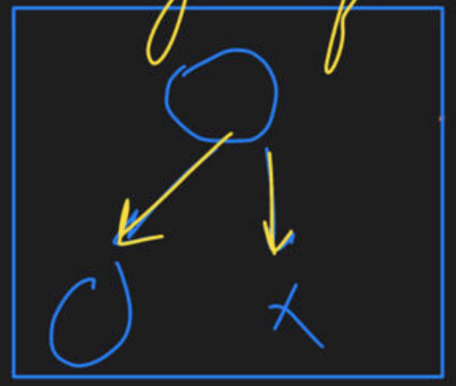
if (fun(root → left))

else fun(root → right)

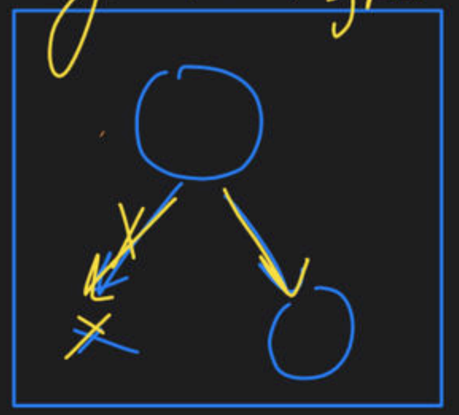
If left node → return



go to left

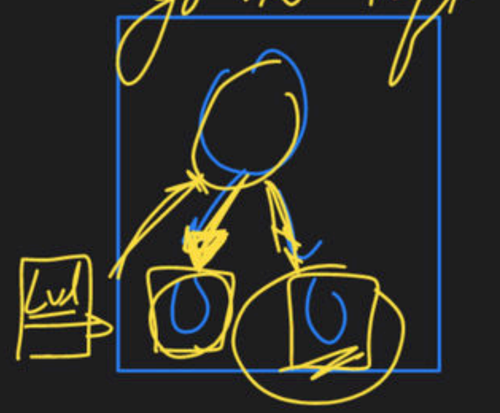


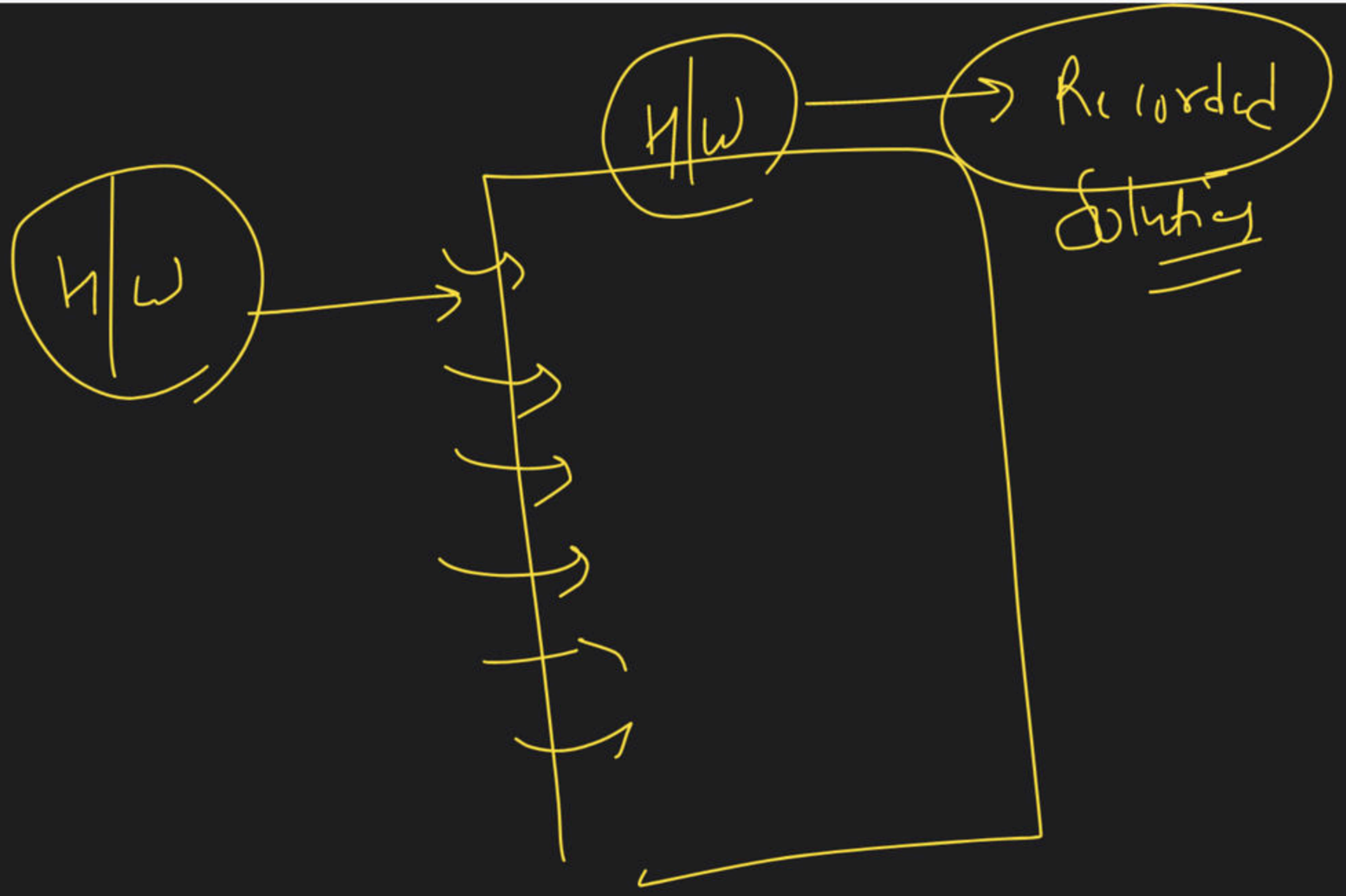
go to right



Lvl → 0

go to left




H/W
H/W
Recorded
Solution