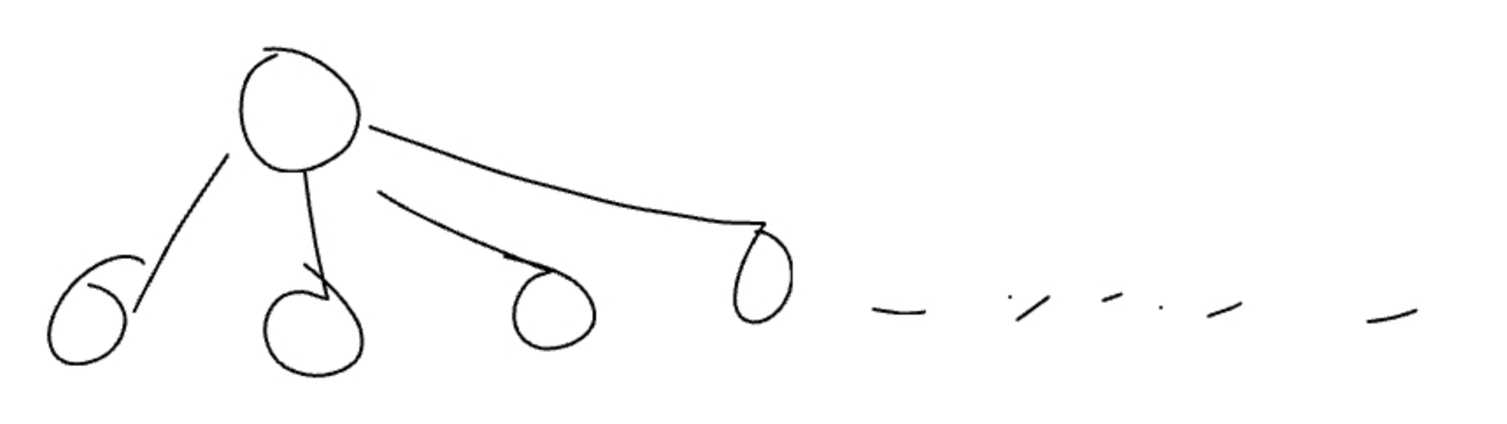
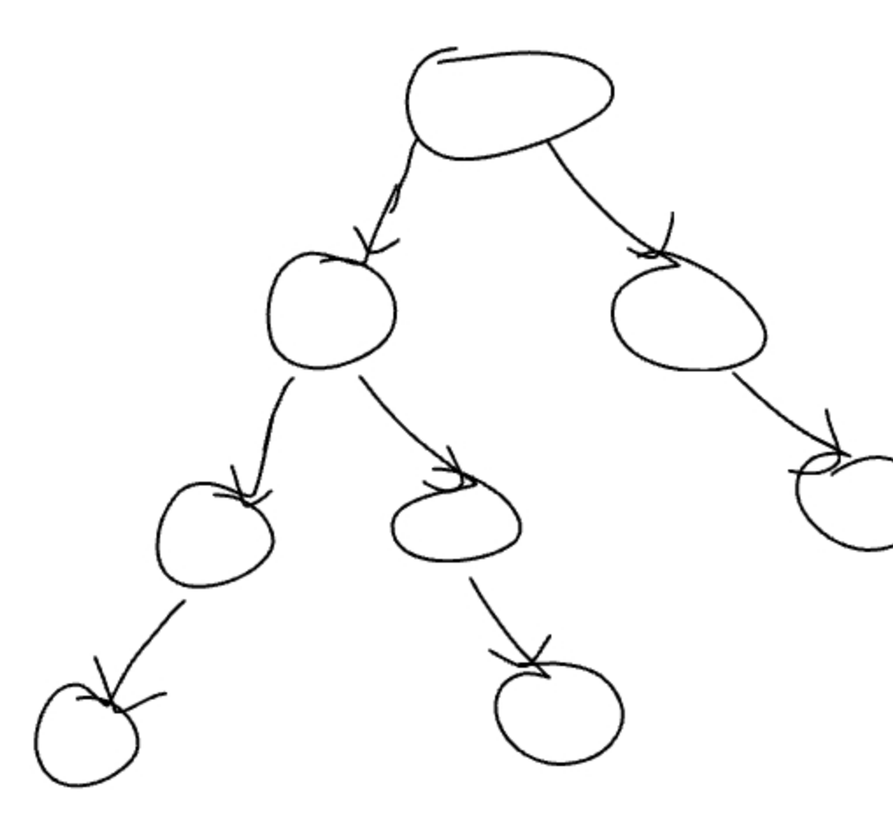




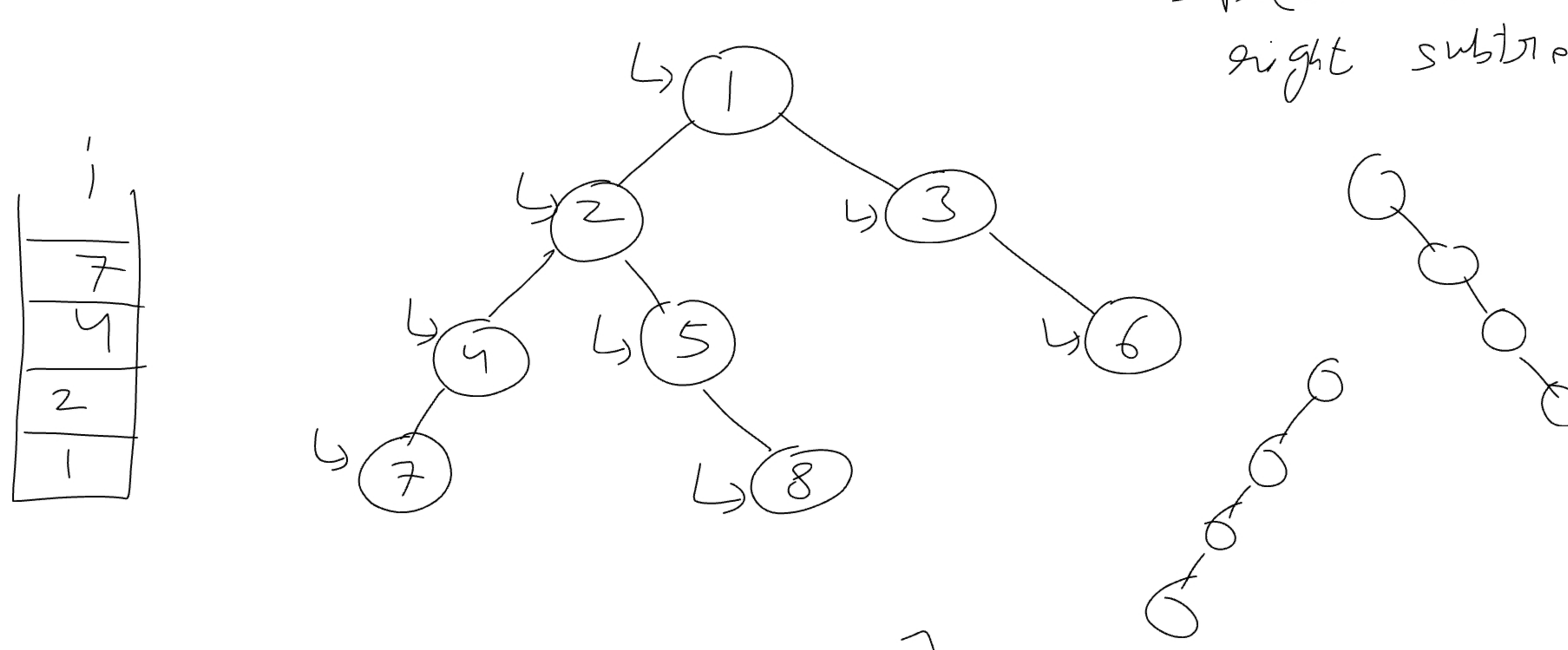
Node
~~GenericTree~~ {
int data,
vector <Node*> children.
}

Preorder:

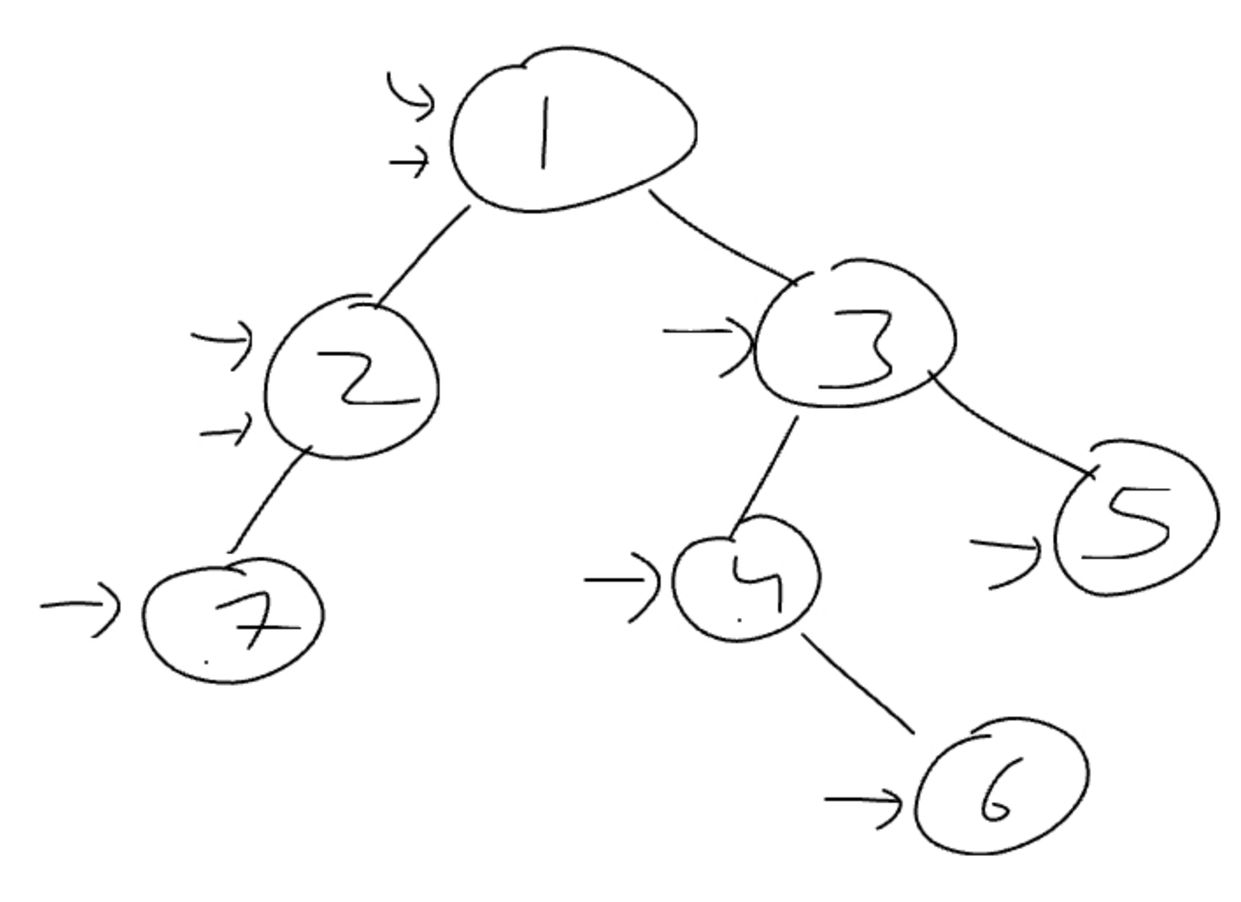
root → (left) → (right)

(left) → Preorder trav. of left subtree

(right) → Preorder trav. of right subtree

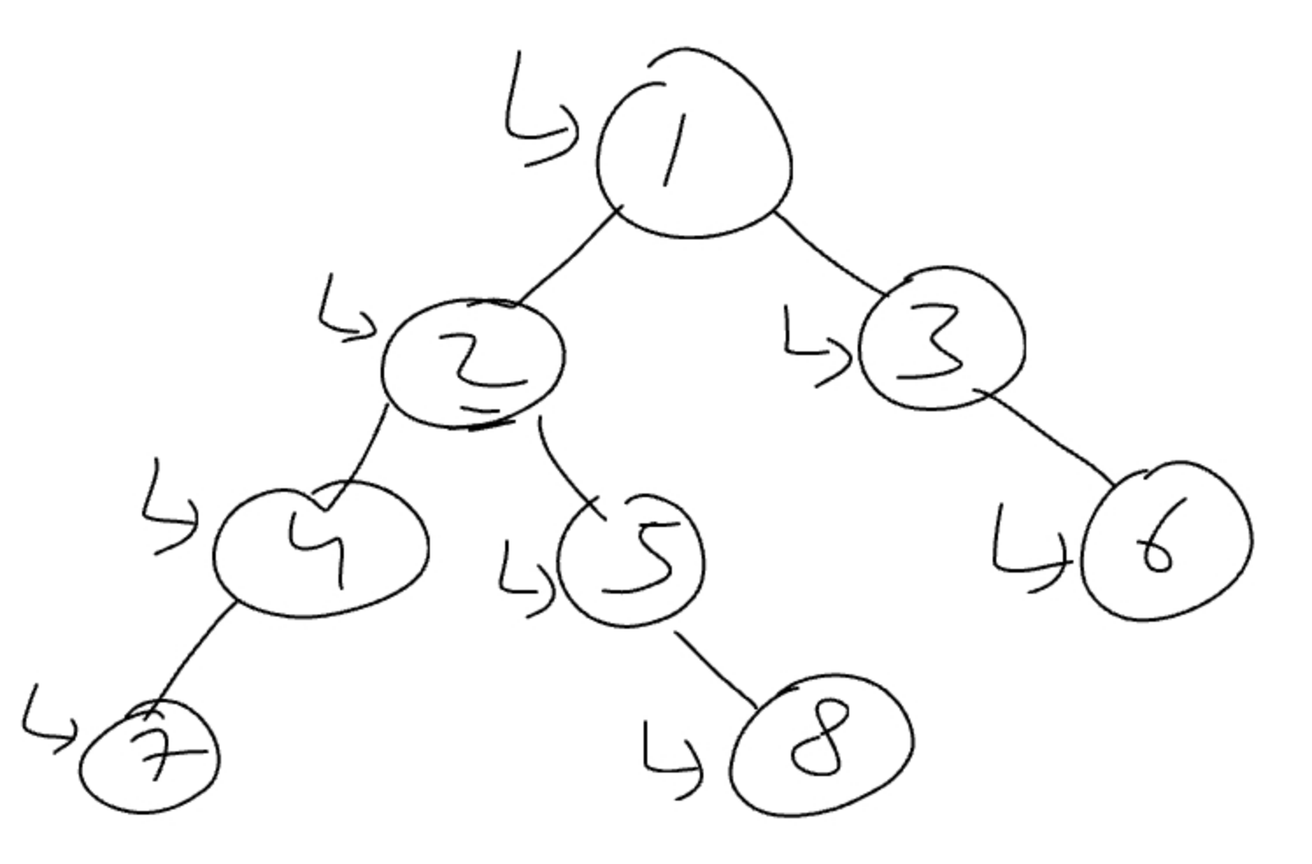


o/p: [1, 2, 4, 7, 5, 8, 3, 6]



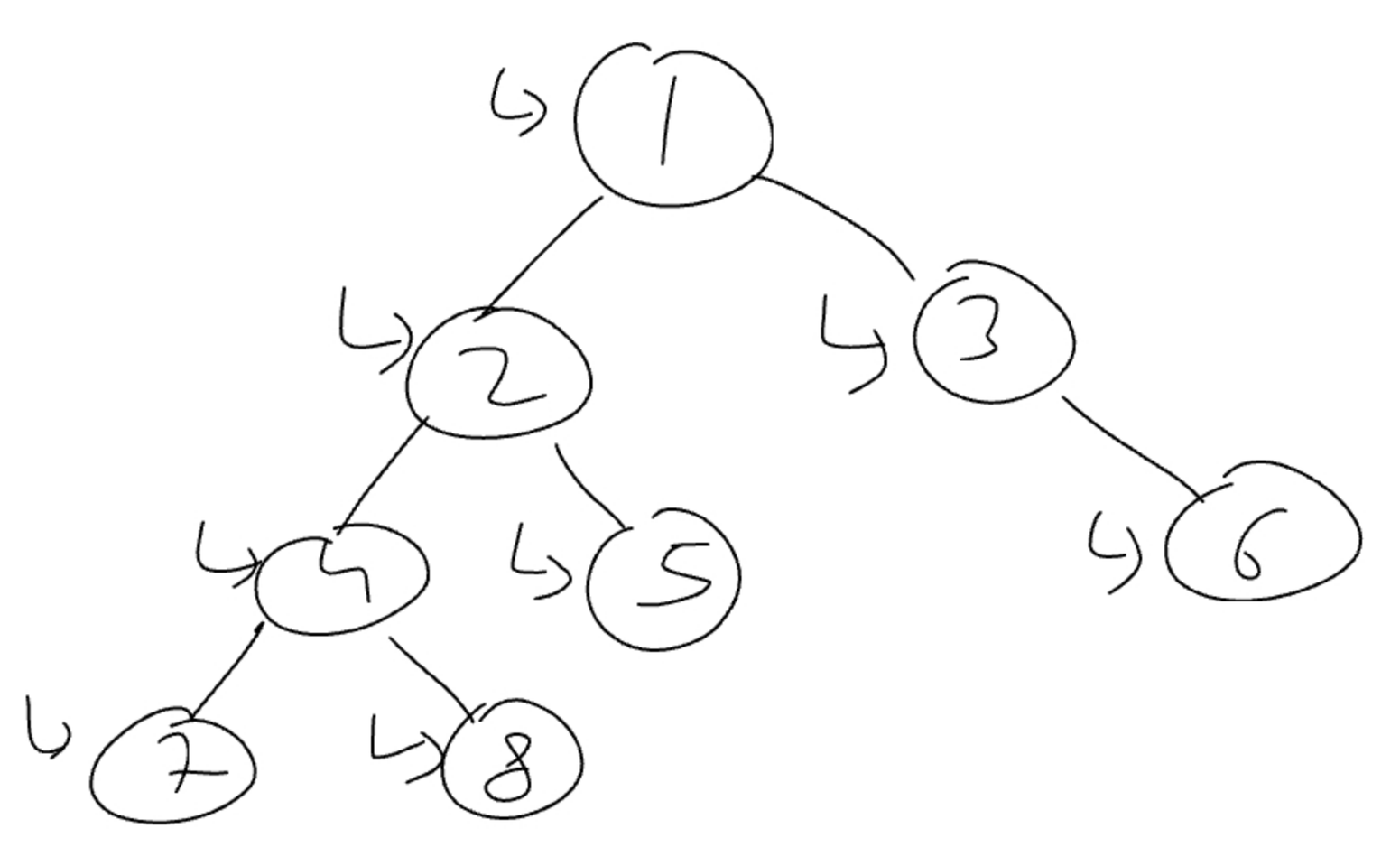
pre: [1, 2, 7, 3, 4, 6, 5]
post: [7, 2, 6, 4, 5, 3, 1]
in: [7, 2, 1, 4, 6, 3, 5]

Postorder: (left) → (right) → root



o/p: [7, 4, 8, 5, 2, 6, 3, 1] ✓

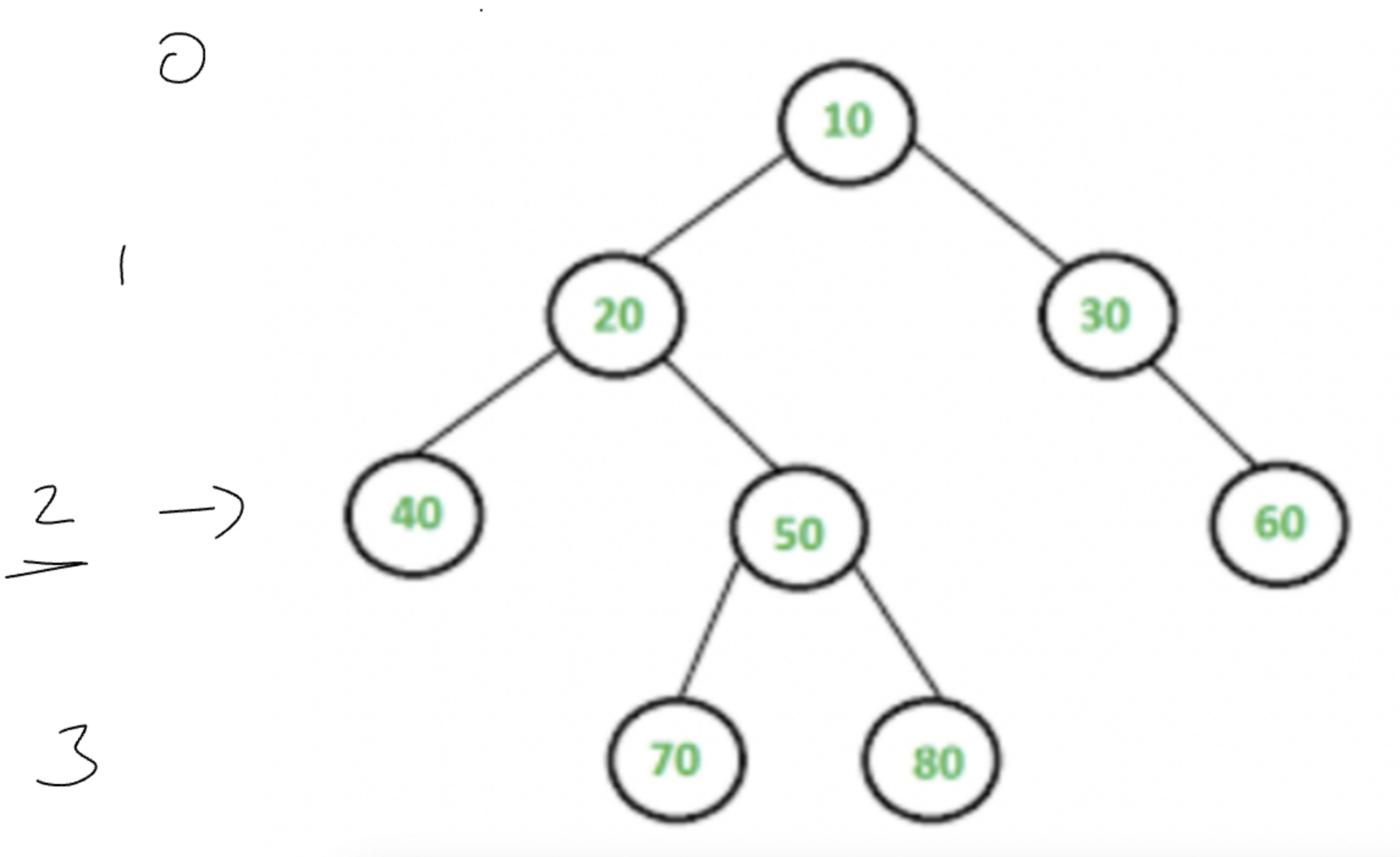
Inorder: (left) → root → (right)



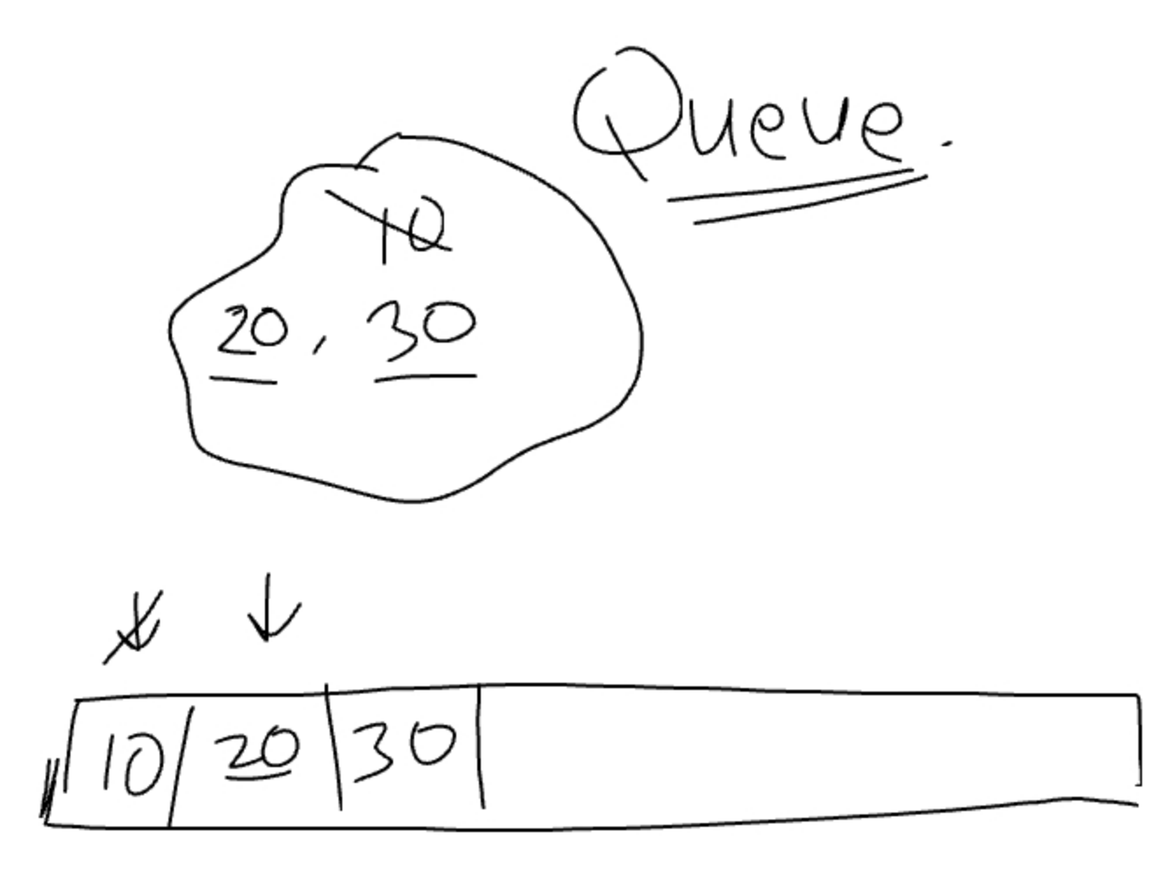
o/p: [7, 4, 8, 2, 5, 1, 3, 6]

Levelorder Traversal

[10, 20, 30, 40, 50, 60, 70, 80]

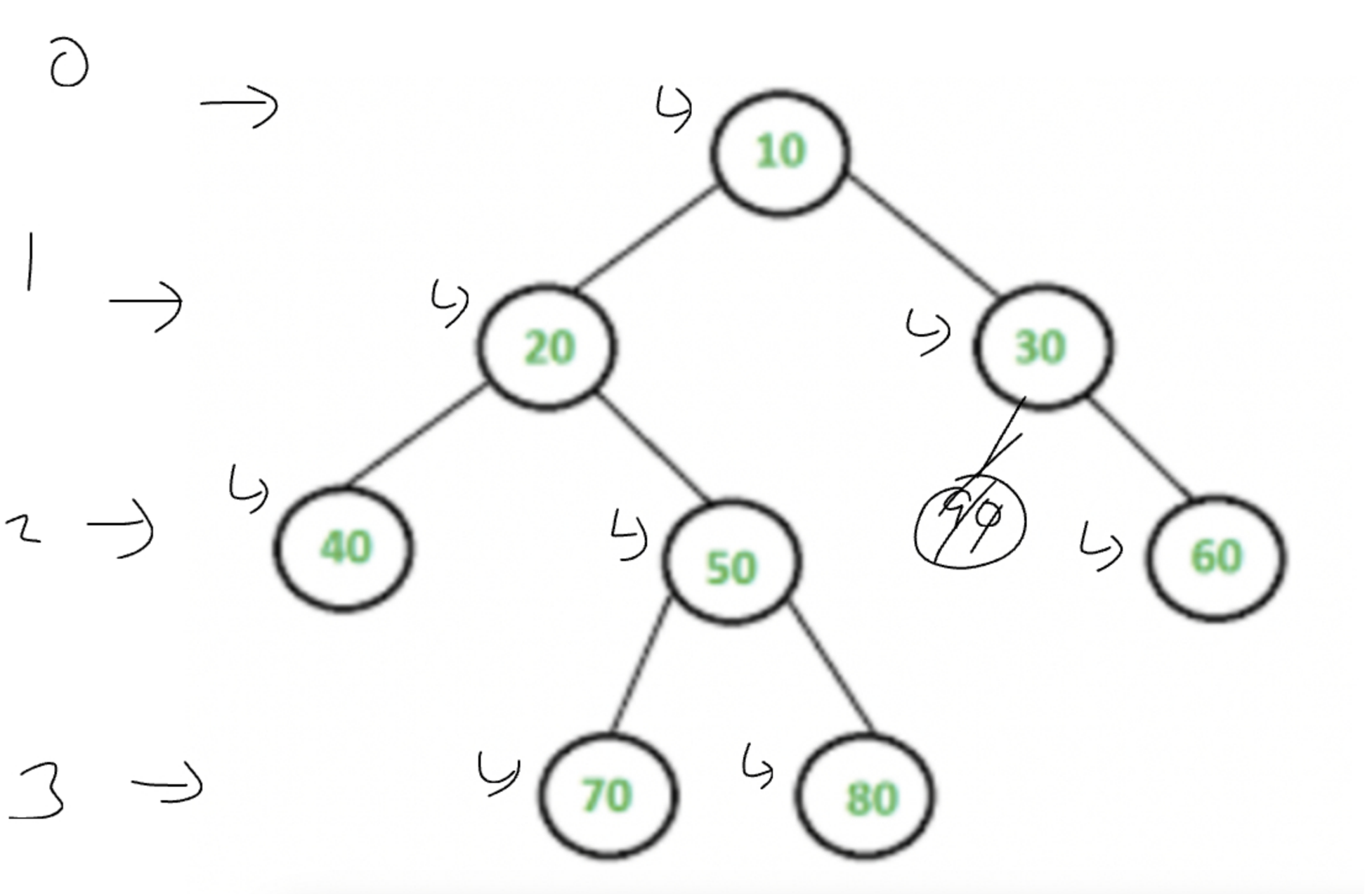


[~~20, 30~~].



[10, 20, 30

[10,



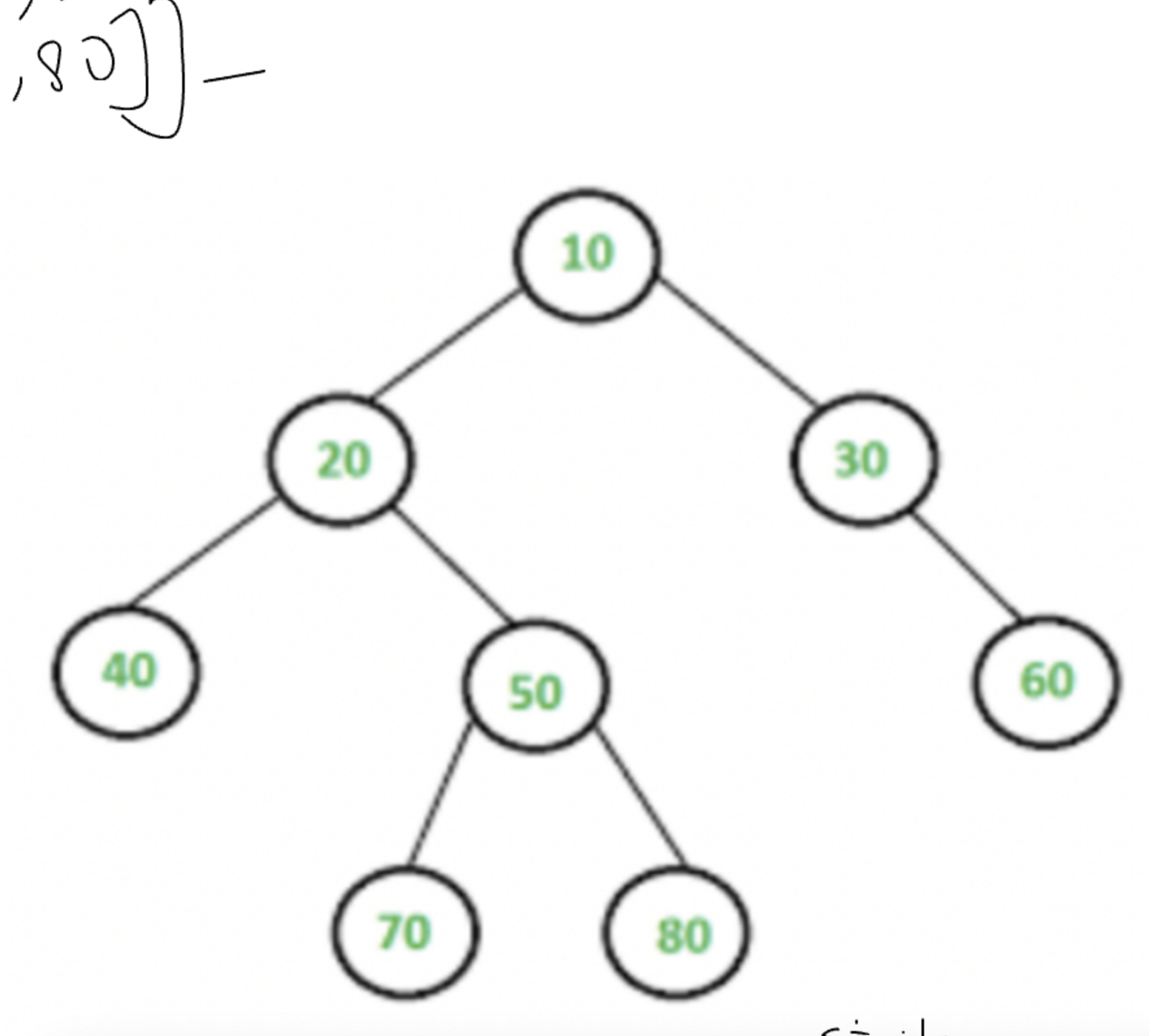
Queue.
~~10~~ ~~20~~, ~~30~~, ~~40~~, ~~50~~
~~60~~, ~~70~~, ~~80~~

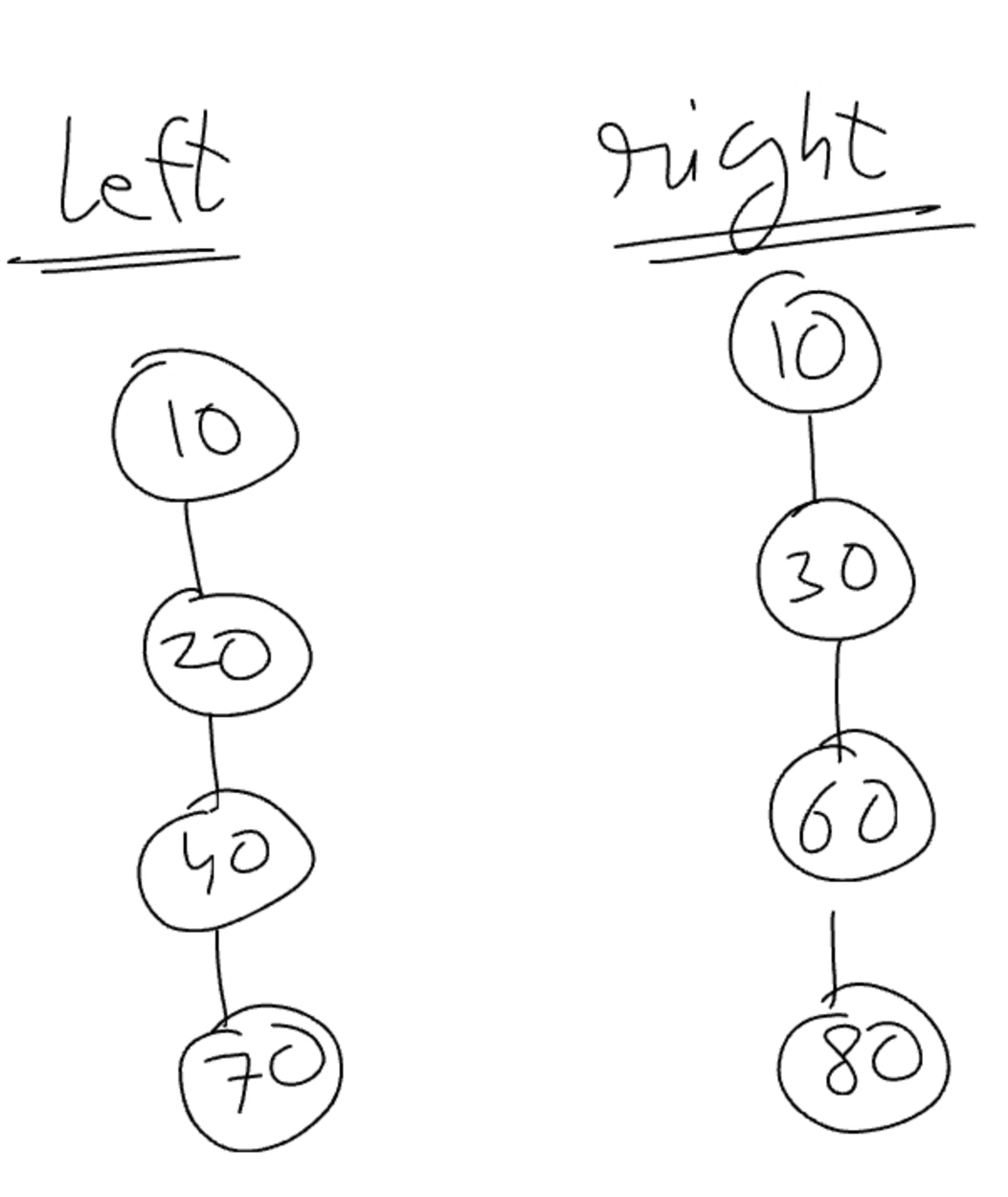
o/p: [10, 20, 30, 40, 50, 60,
70, 80]

[~~10~~, ~~20~~, ~~30~~, 40, 50, 60, ...

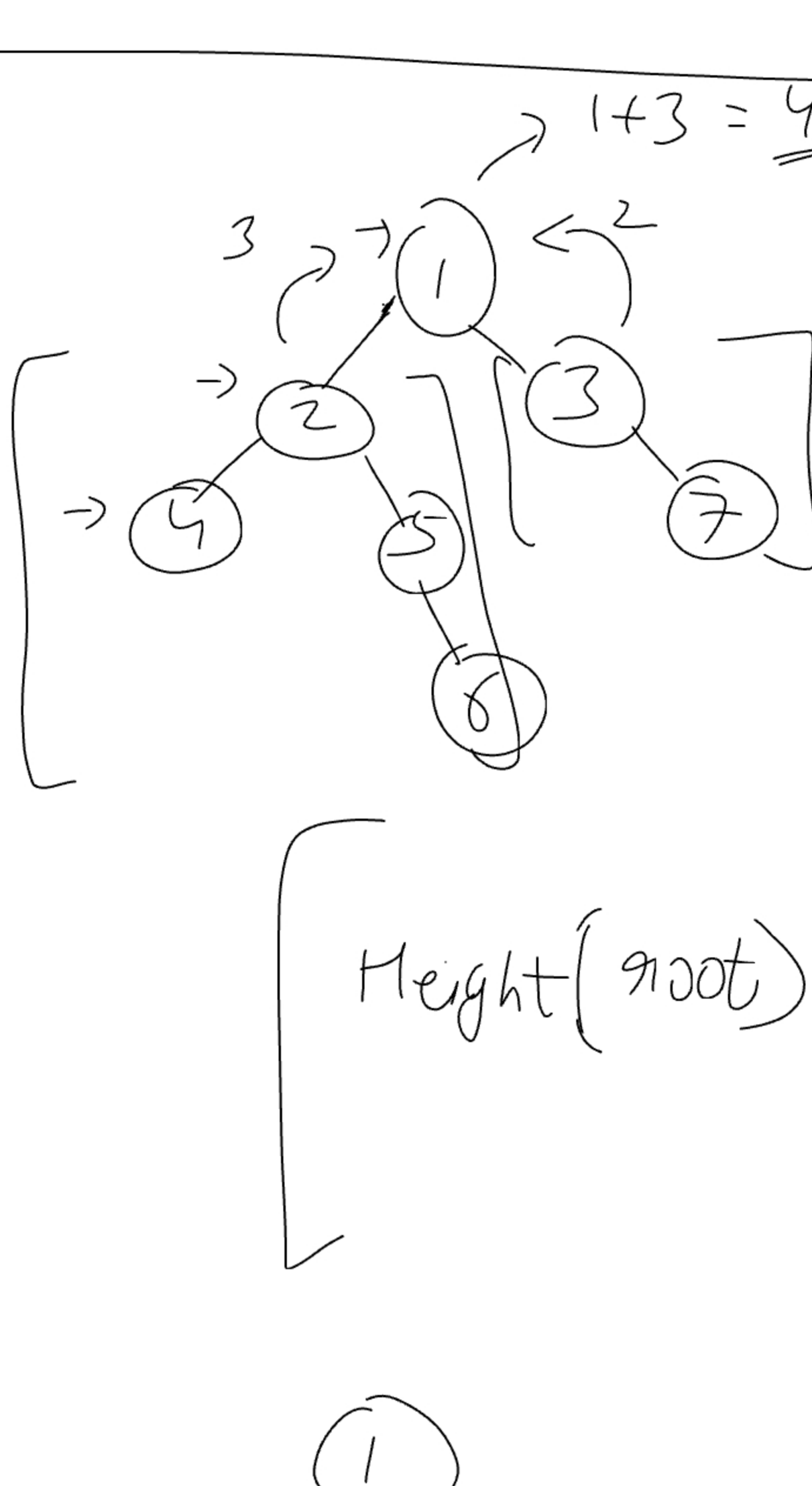
TC: O(n)
AS: ~~O(n)~~. O(n) → O(max no. of nodes in any level)

[[10] –
[20, 30] –
[40, 50, 60] –
[70, 80]] –



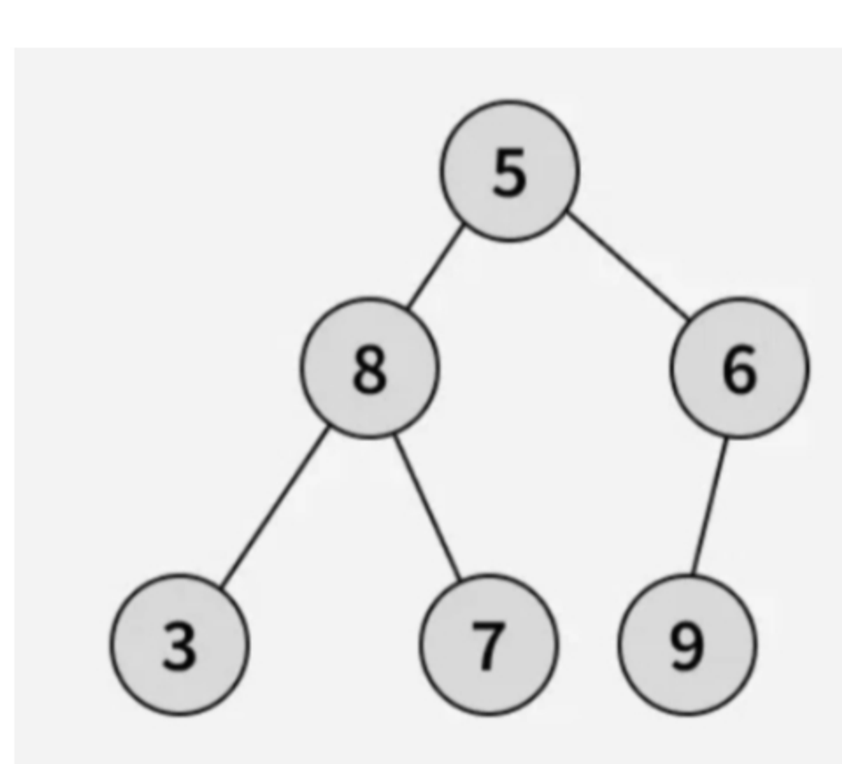


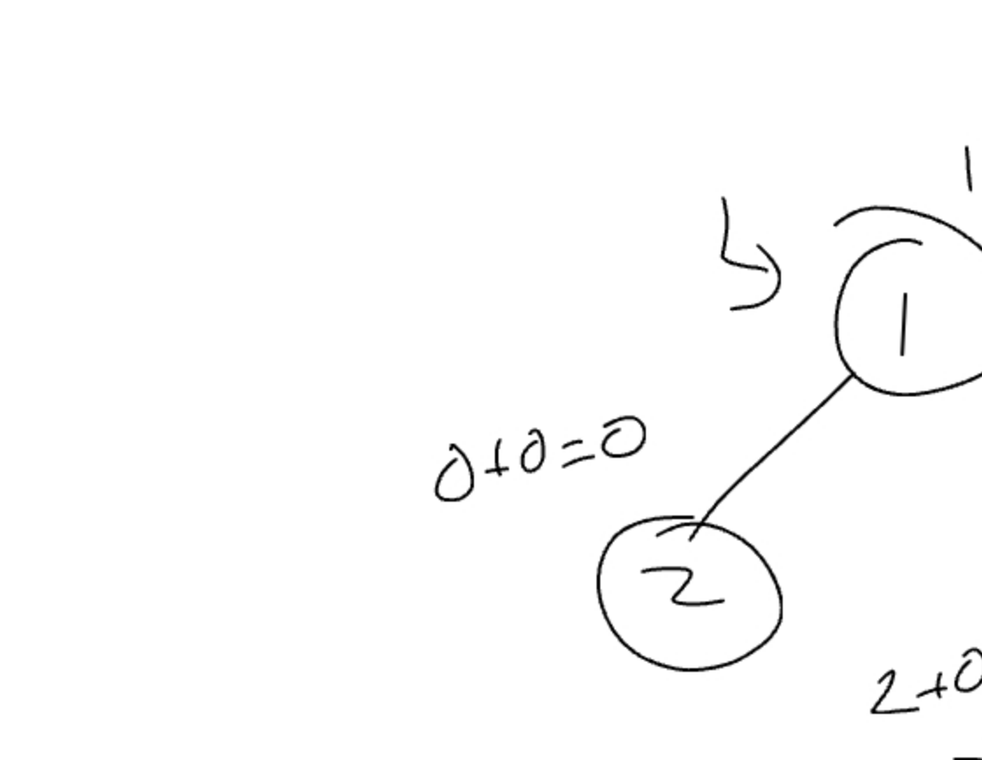
1st itr. queue: 10
2nd itr. queue: 20 30
3rd itr : 40, 50, 60.
4th itr : 70, 80



temp = ~~X~~ ~~2~~ ~~3~~ 2

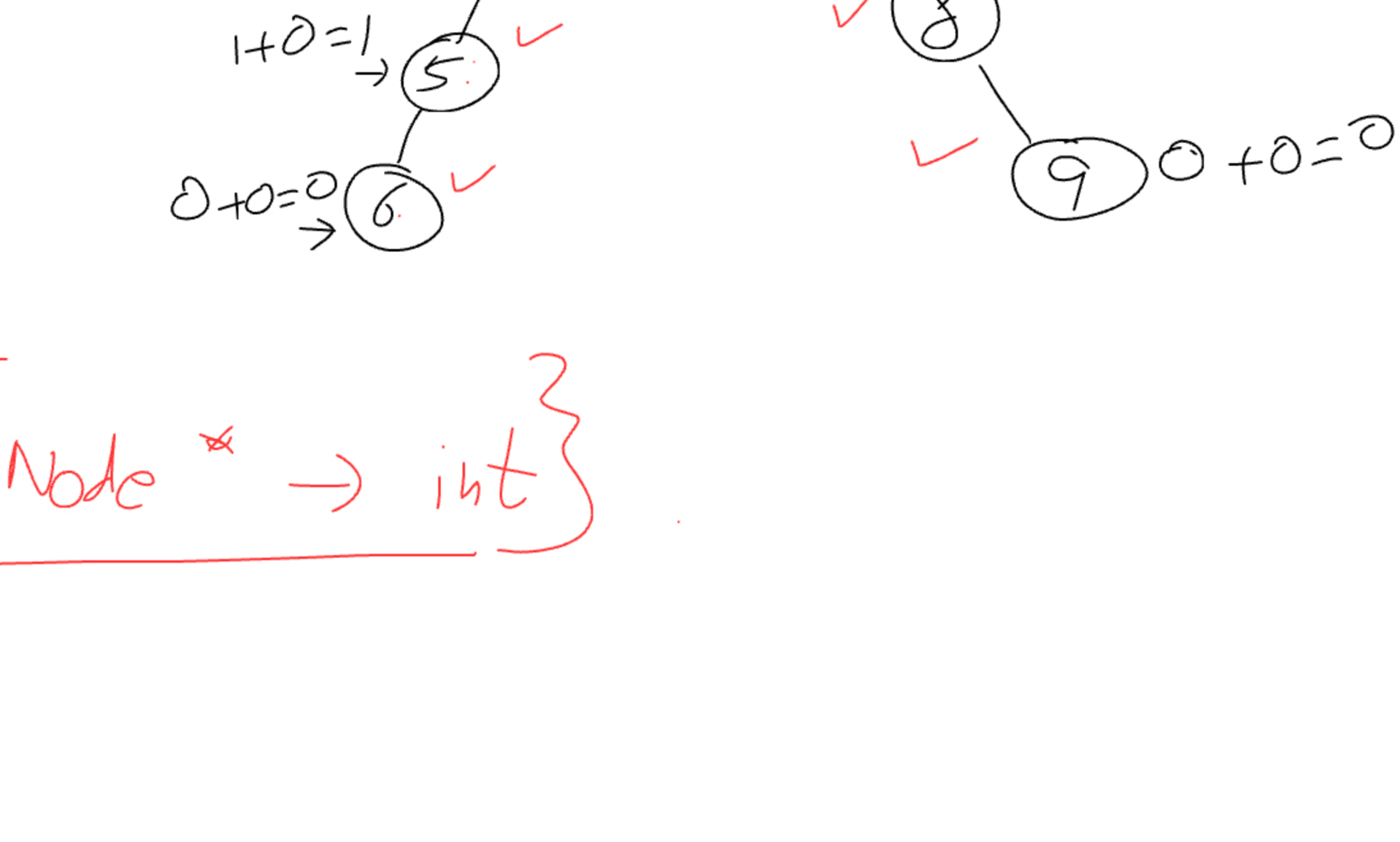
Height(root) = 1 + max (Height(root → left), Height(root → right))





3 → 7 : 3 → 8 → 7 : 2
3 → 9 : 3 → 8 → 5 → 6 → 9 : 4
7 → 9 : 7 → 8 → 5 → 6 → 9 : 4

max = 4
diam.



lh = 1
rh = 4

5

Map: { Node* → int }