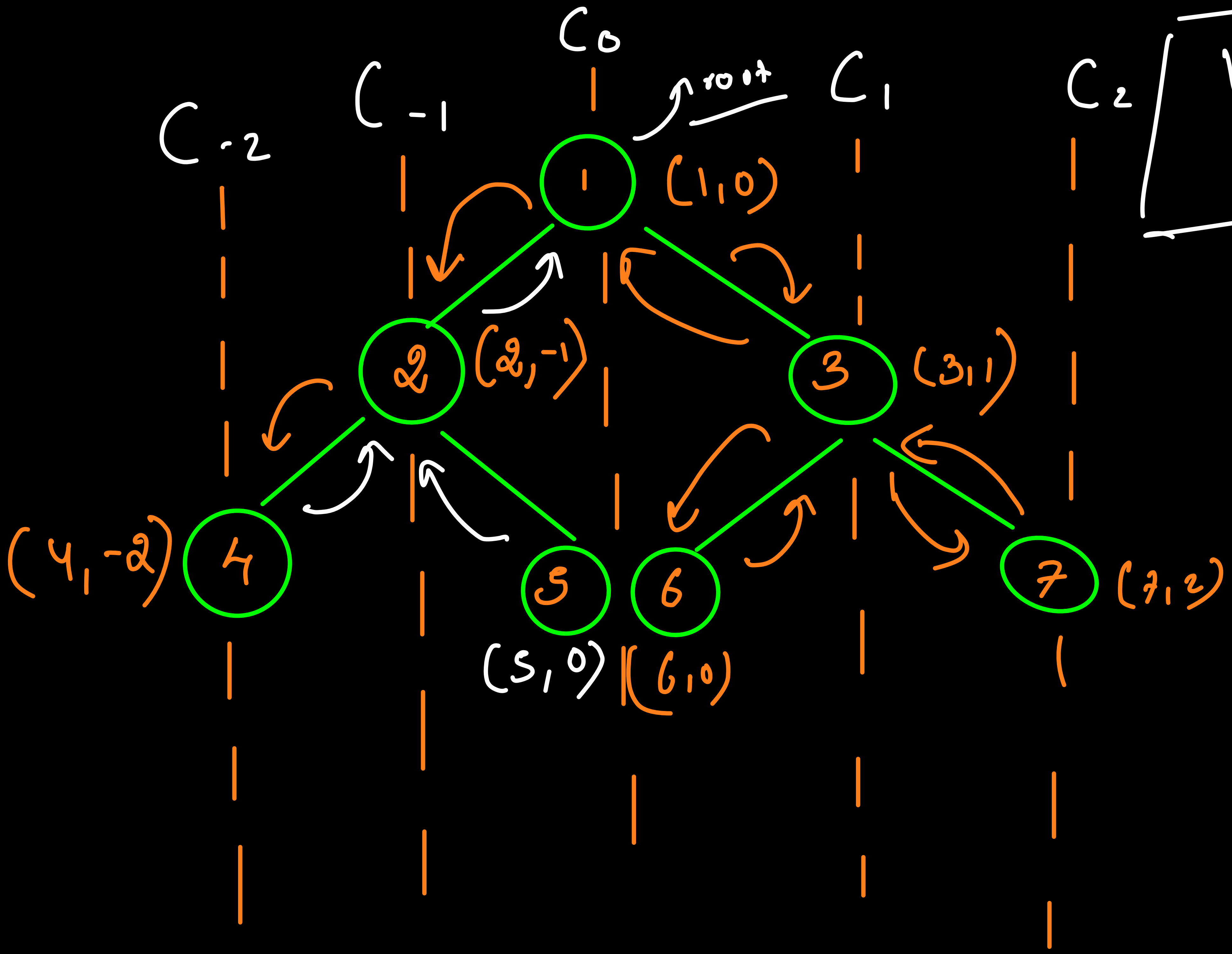
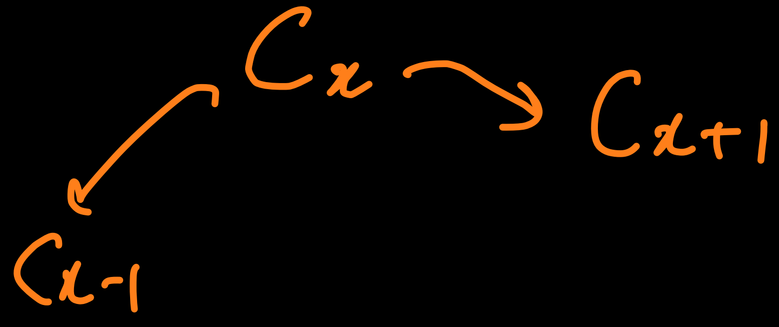
Vertical Order Traversal



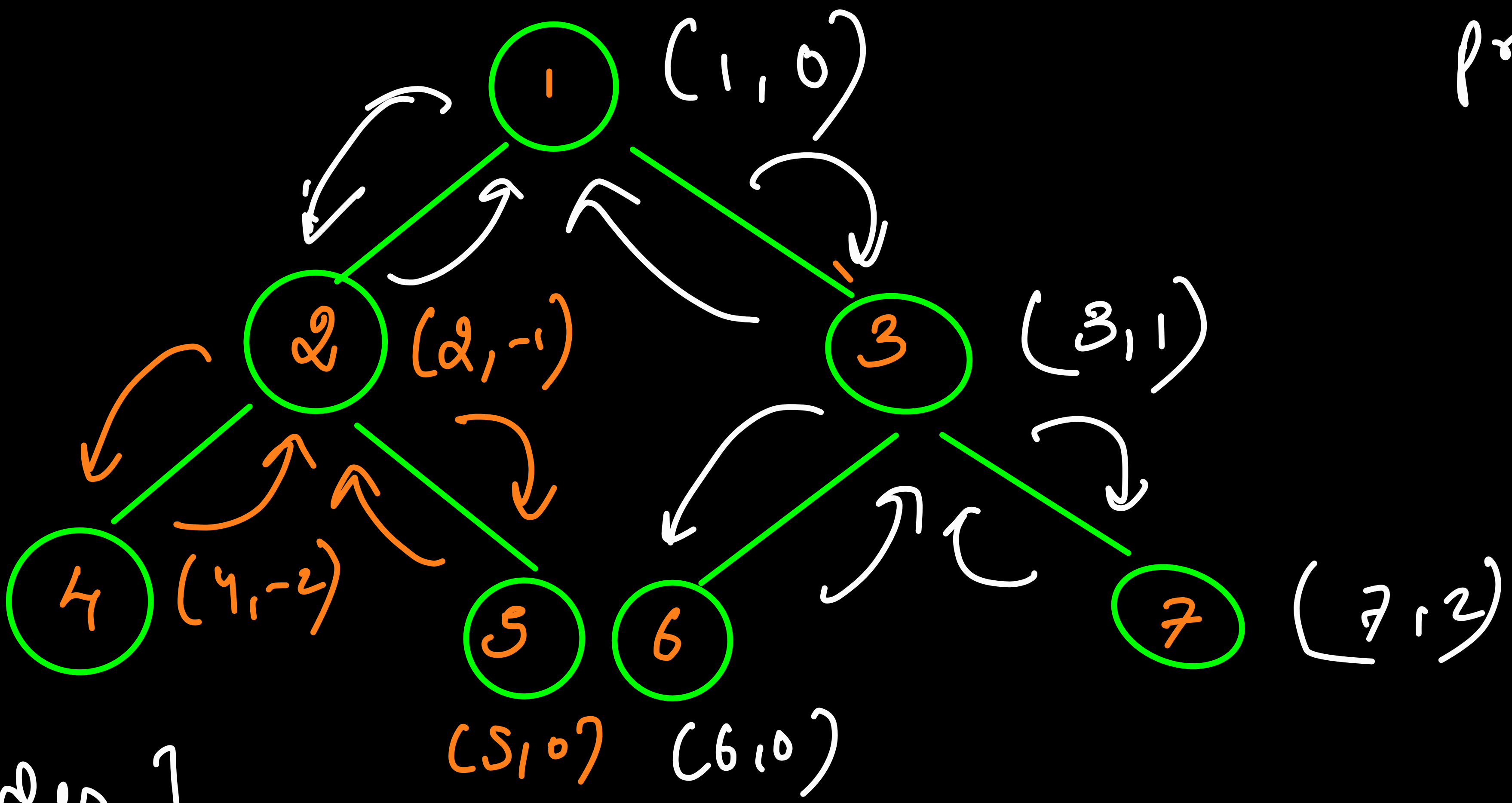


pre(root, col)

what if I start collecting all nodes with same col value.

from any node if you move on step left (left child), you went (col-1), if you move on step right (right child) you went (col+1) min_col = -2

max_col = 2



pre(root, 0)

O(n)

O(n)

col → [nodes]

↑

Key-value

{

0 : [1,5,6]

-1 : [2]

-2 : [4]

1 : [3]

2 : [7]

}

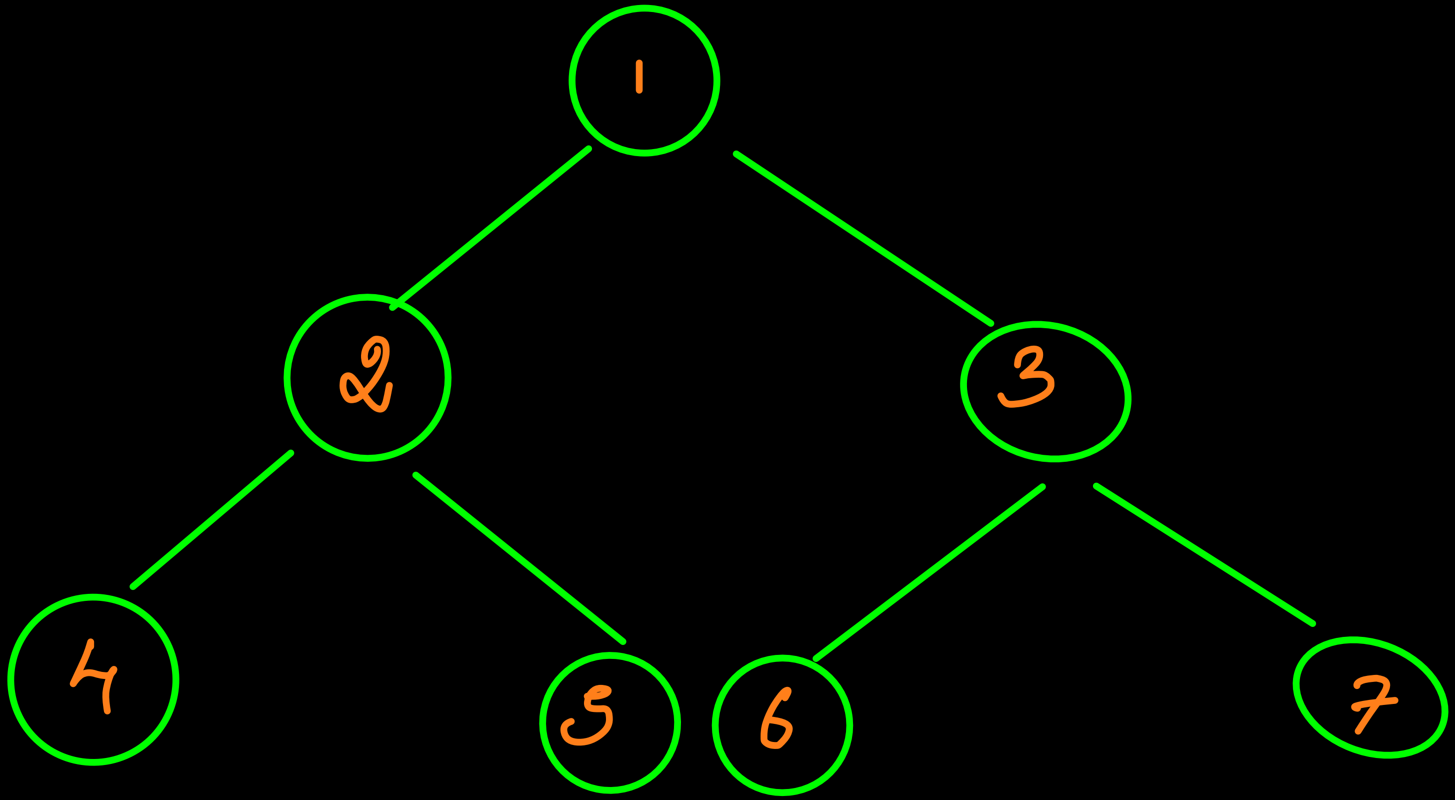
```
for( i = min_col ; i ≤ max_col ; i++)
    res.push( mp[i])
```

[[4], [2], [1,5,6], [3], [7]]

→ O(nlogn)

```
mp = { }
min-col
max-~

f(root, col)
        if( root==null) return;
        if(!mp[col]) mp[col] = [root.val];
        else  mp[col].push(root.val)
        min-col = min (min-col, col);
        max-col = max (max-col, col);
       f(root.left, col-1)

       f(root.right, col+1)

}
```

Top view



Bottom view

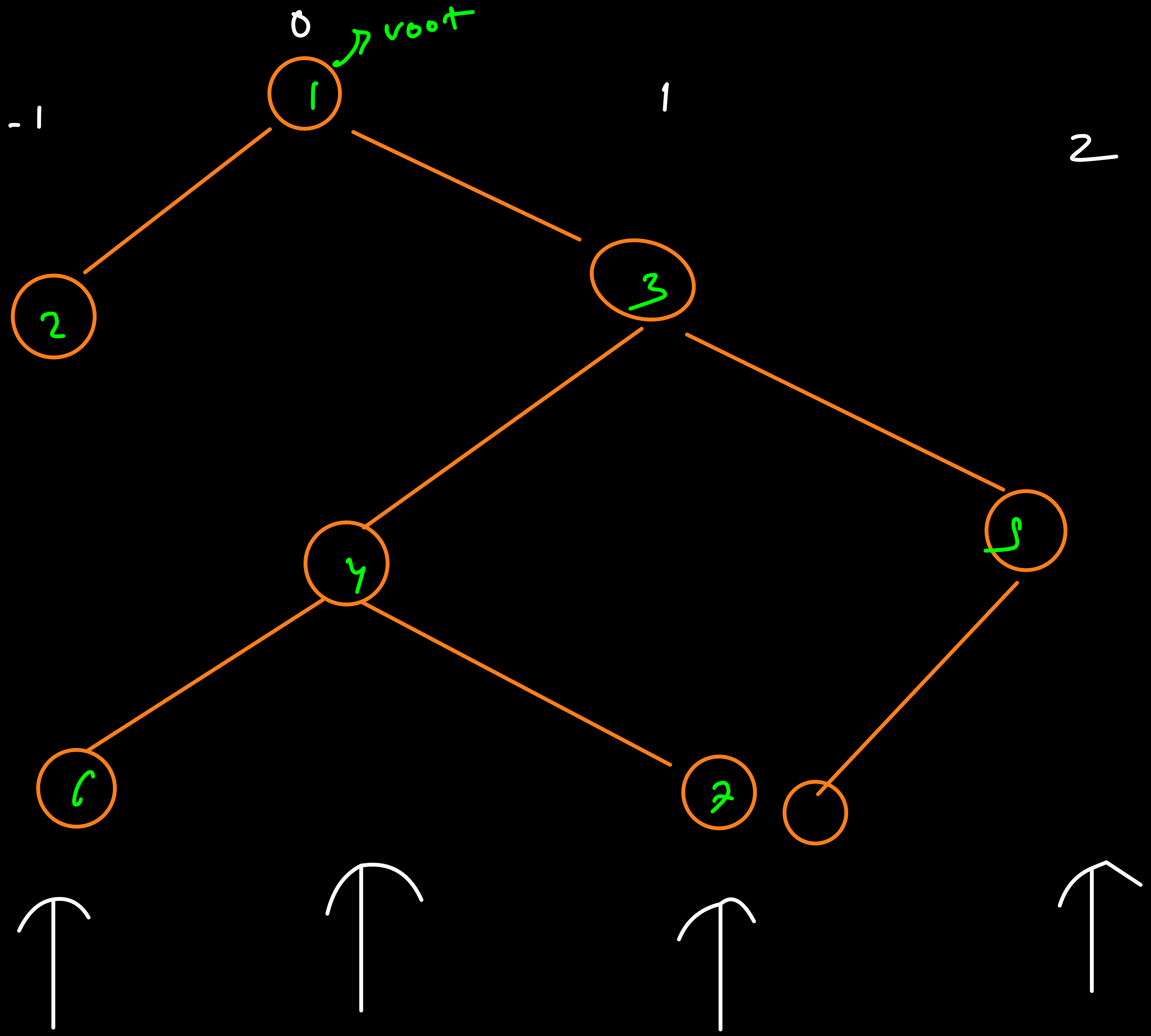
Bottom view

-1 [2, 6]

0 : [1, 4]

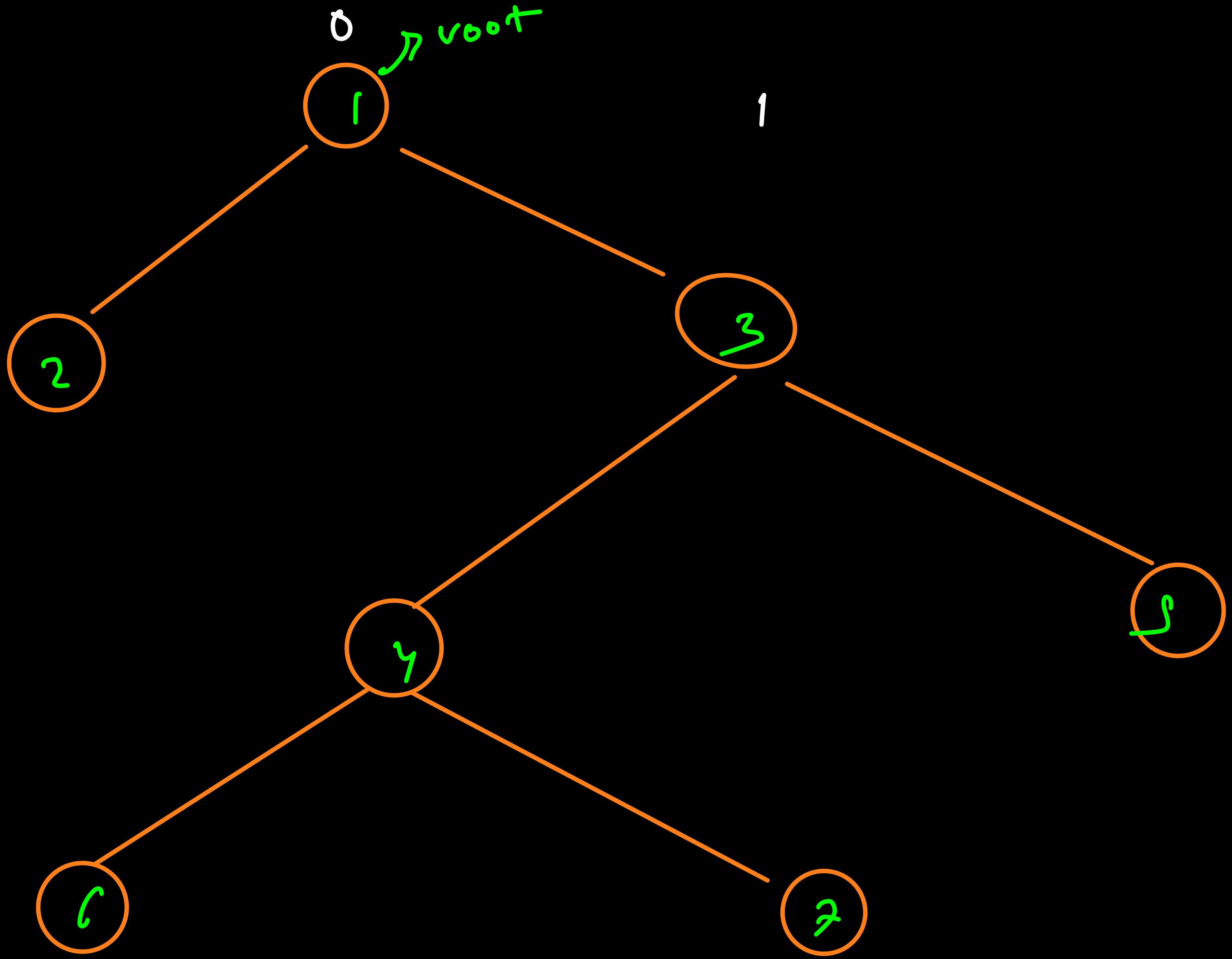
1 : [3, 7]

2 : [5]



Boundary Traversal

Leftview
+
Bottom view
+
Right view



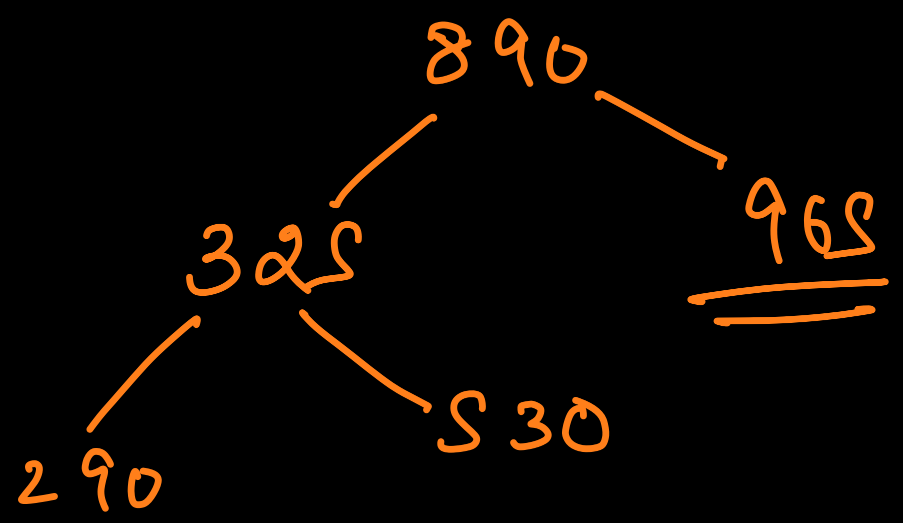
Q→ Given Preorder traversal of a BST, print the leaf nodes of BST without creating the BST.

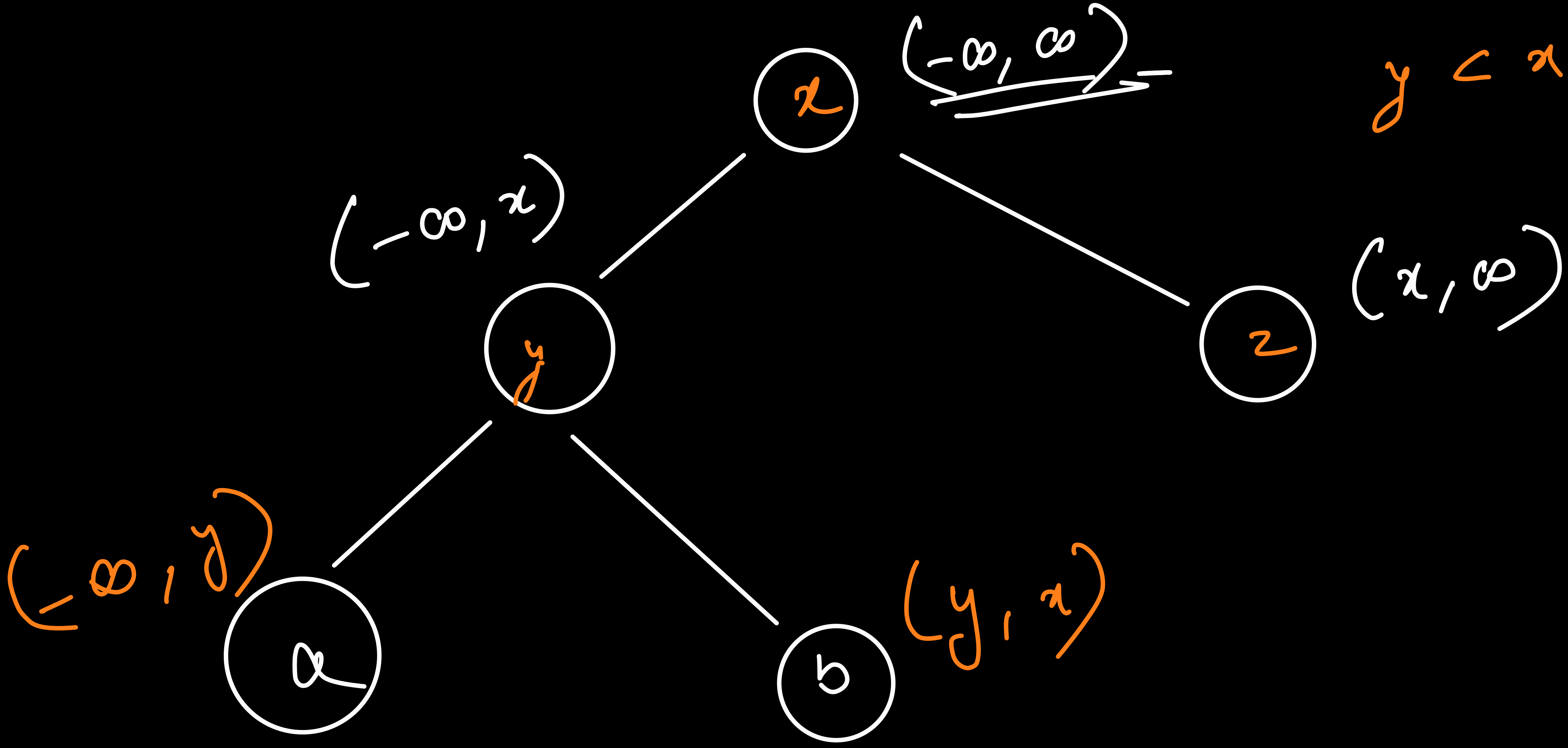
$(-\infty, \infty)$ $(-\infty, 890)$ $(-\infty, 325)$

Ex → [890, 325, 290, 530, 965]

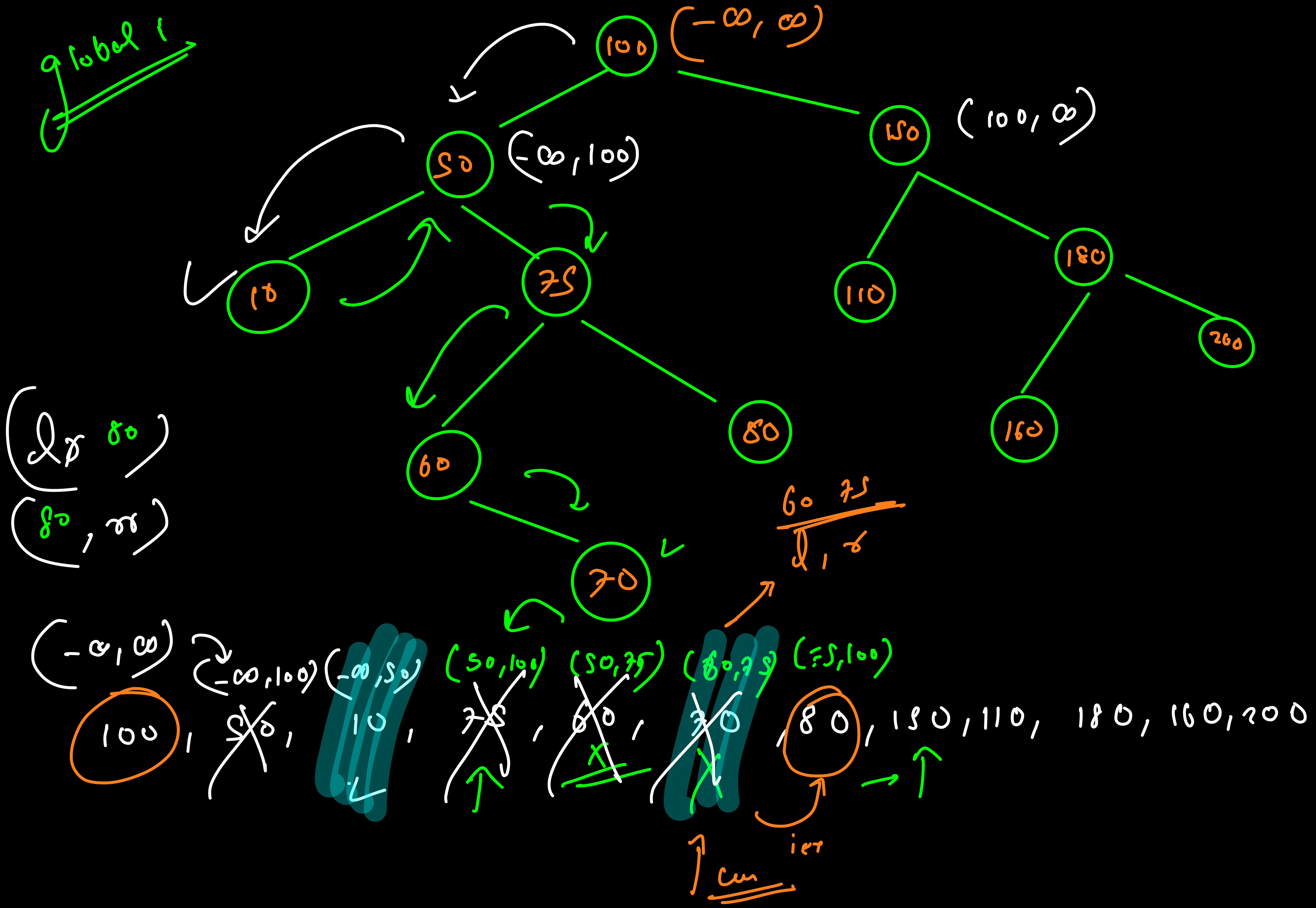
ans → [290, 530, 965]

$(-\infty, r)$ (r, ∞)





$y < x < z$


global i
100
(-∞, ∞)
50
(-∞, 100)
150
(100, ∞)
10
75
110
180
200
60
80
160
70
(l, r 80)
(80, r)
60 75
l, r
(-∞, ∞)
(-∞, 100)
(-∞, 50)
(50, 100)
(50, 75)
(60, 75)
(75, 100)
100, 50, 10, 75, 60, 70, 80, 150, 110, 180, 160, 200
curr
i

$f(\text{root}, lr, rr)$

$f(\text{root}, -\infty, \infty)$

pre → [ ] → global i

global

i → curr node

pre → [ ]

i → [ ]

pre[i]

50,100 50,75 60,75 75,100)

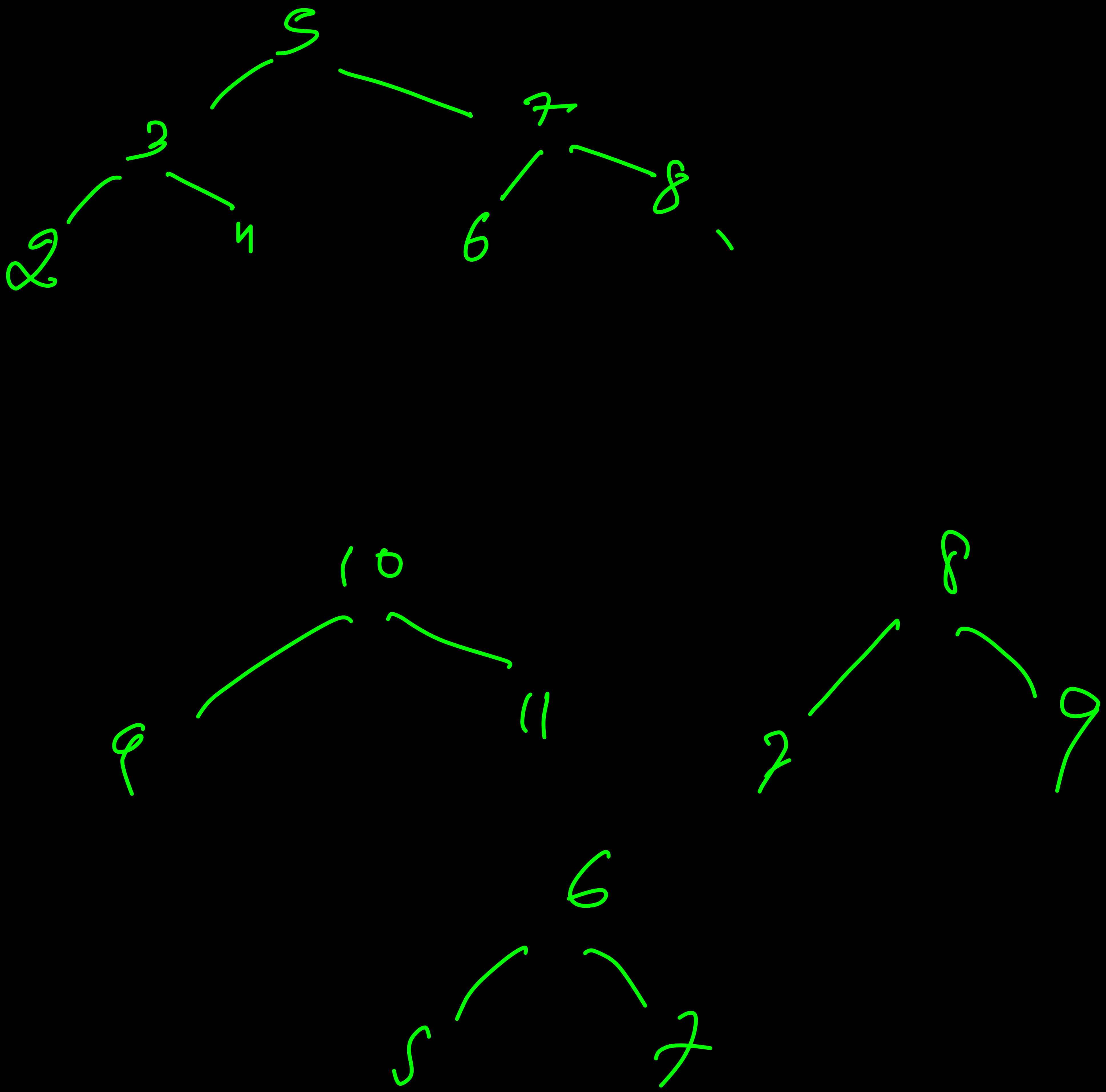
75, 60, 70, 80

f(75, 50, 100)

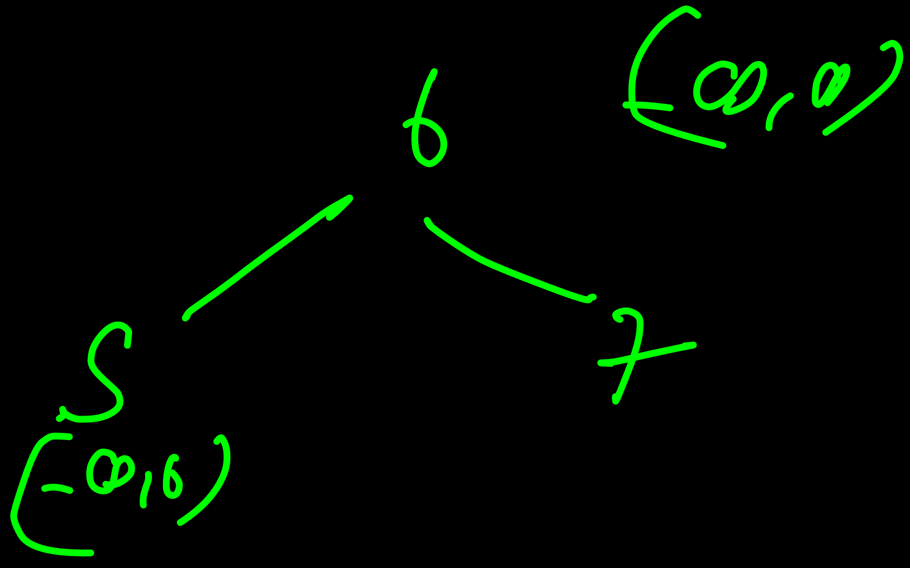
↓

f(80, 75, 100)

```
f((curr) l, r)
    if (i >= pre.length) return;

    i++;

    if (pre[i] > l and pre[i] < r) {
            if(pre[i] > l and pre[i] < curr) {
                f(pre[i], l, curr);
            else f(pre[i], curr, r);
    }
    else {
        res.push(curr);
    }

    if( pre[i] > l and pre[i] < curr) {
            f(pre[i], l, curr)
    if (pre[i] > curr and pre[i] < r) {
        f(pre[i], curr, r)
    }
```



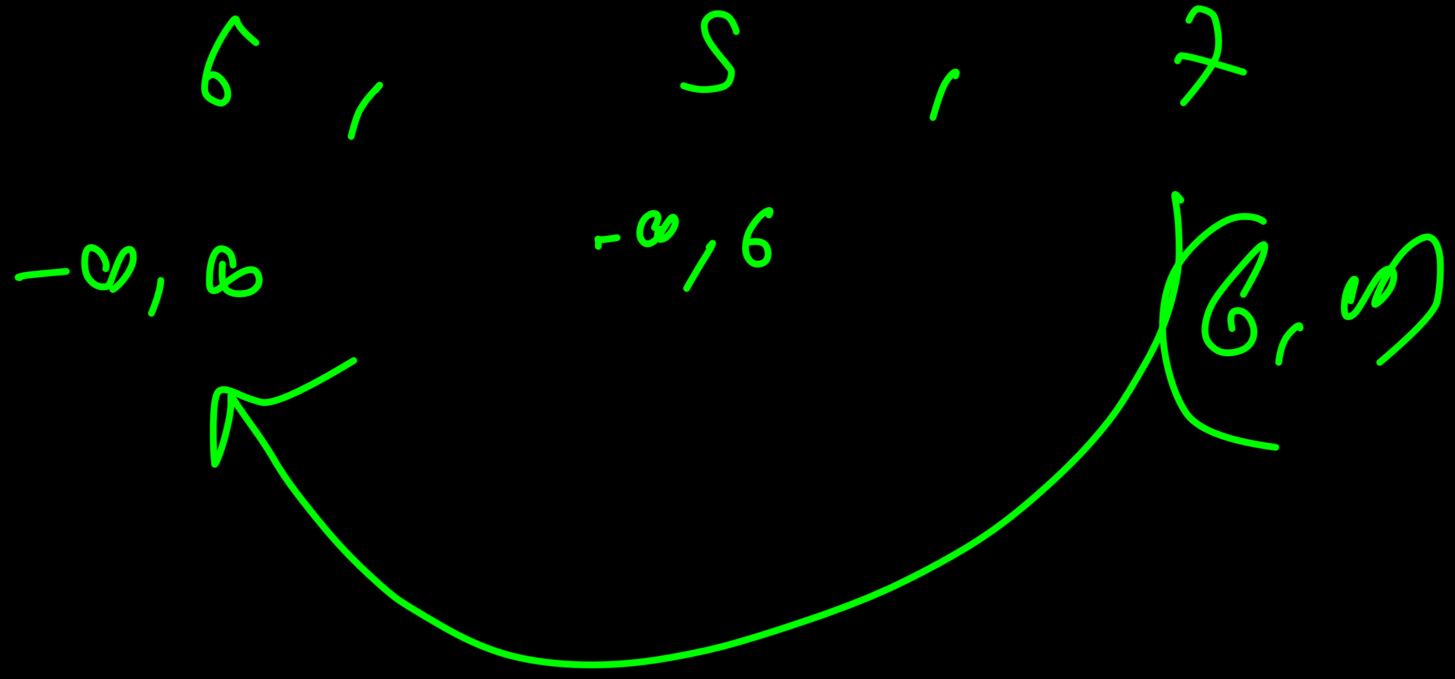
5
3
7
2
4
6
8
10
9
11
8
7
9
6
5
7



6, 5, 7

[5,]

Li°



```
f(curr, l, r) {
    if (i >= pre.length) retun
    i++;

    if (pre[i] > l  and  pre[i] < r) {
        if (pre[i] > l and  pre[i] < curr) // left
            f(pre[i], l, curr);
        else
            f(pre[i], curr, r);
    }
    else {
        res.push(curr);
    }
    if (pre[i] > l  and  pre[i] < curr)
            f(pre[i], l, curr);
    if (pre[i] > curr  and  pre[i] < r)
            f(pre[i], curr, r);
}
```