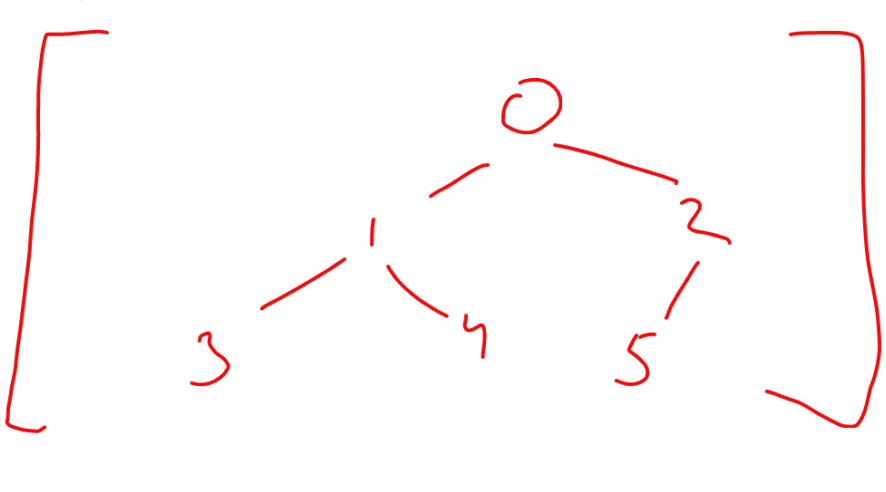
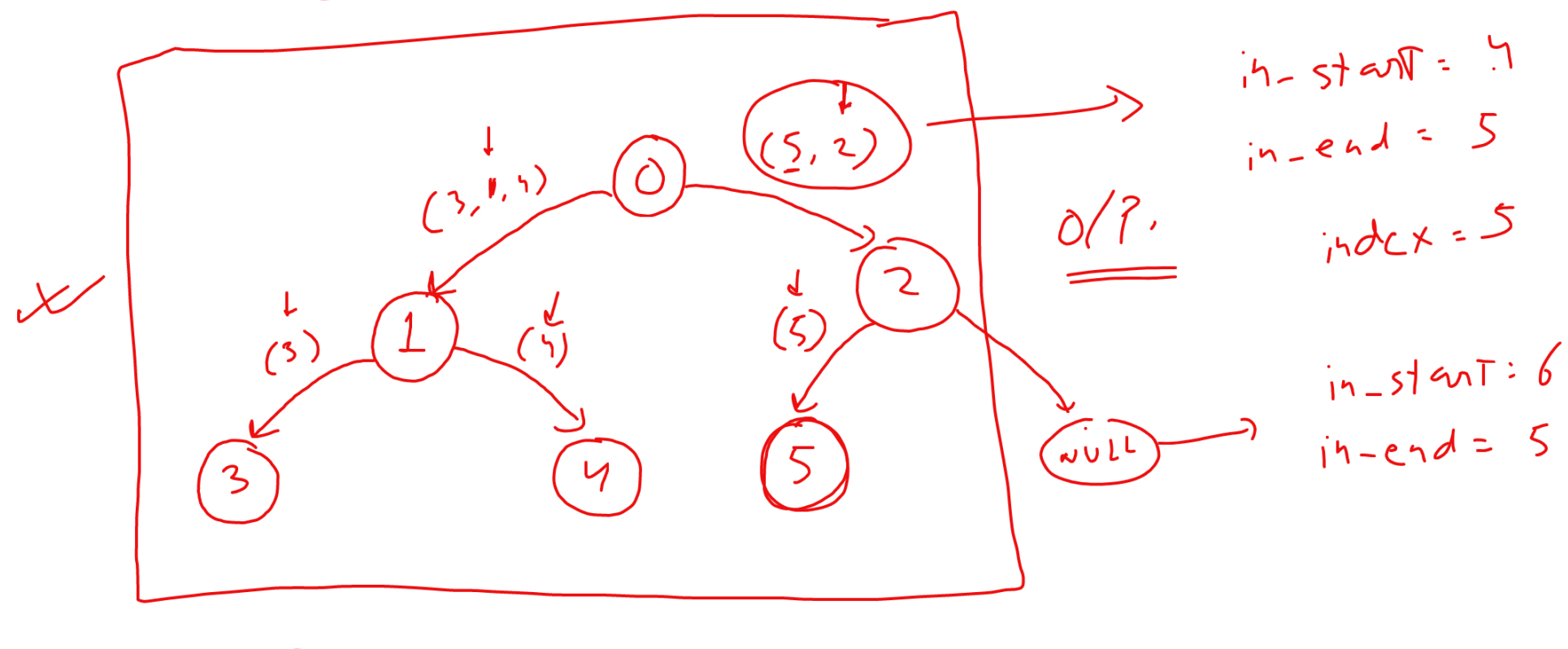
in[] = [3, 1, 4, 0, 5, 2]
pre[] = [0, 1, 3, 4, 2, 5]   I/P


O/P

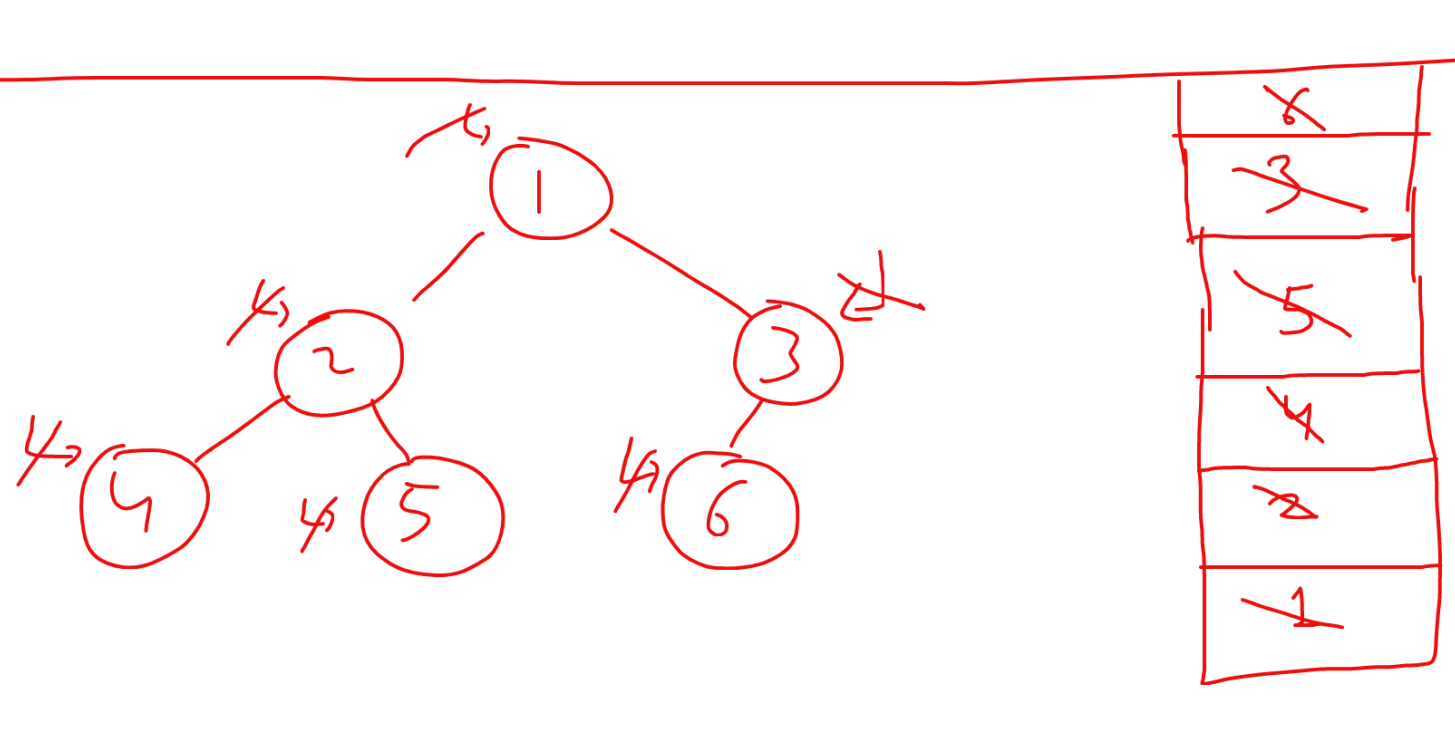
inorder: left subtree → root → right subtree
preorder: root → left subtree → right subtree

in[] = [3, 1, 4, 0, 5, 2]
pre[] = [0, 1, 3, 4, 2, 5]

in_start = 0
in_end = 5
index = 3

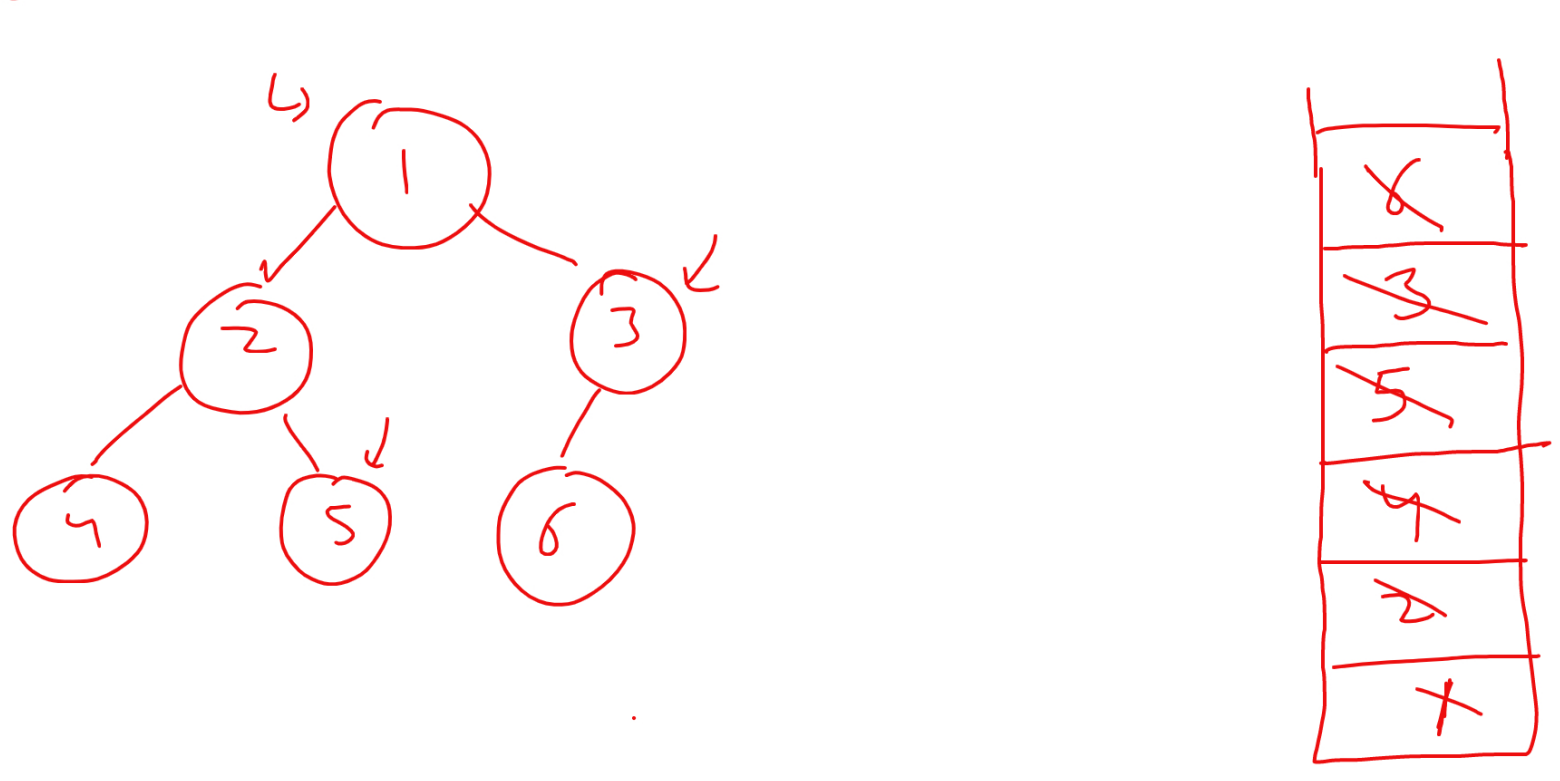

(3, 1, 4)
0
(5, 2)
1
2
(3)
(4)
(5)
3
4
5
NULL
O/P.
in_start = 4
in_end = 5
index = 5
in_start : 6
in_end = 5

0

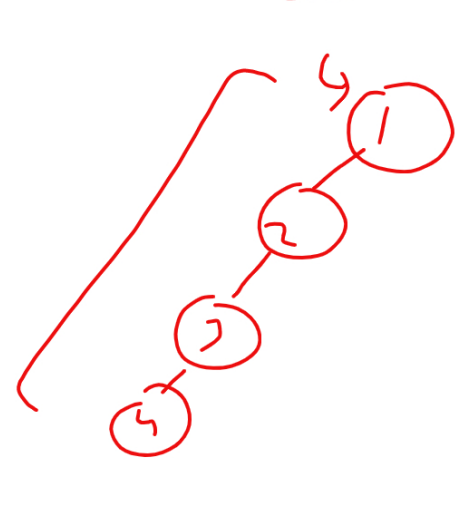

1
2
3
4
5
6
1
2
4
5
3
X

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

push_left() → It will keep pushing the subsequent left nodes till it encounters NULL


1
2
3
4
5
6
1
2
4
5
3
X

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

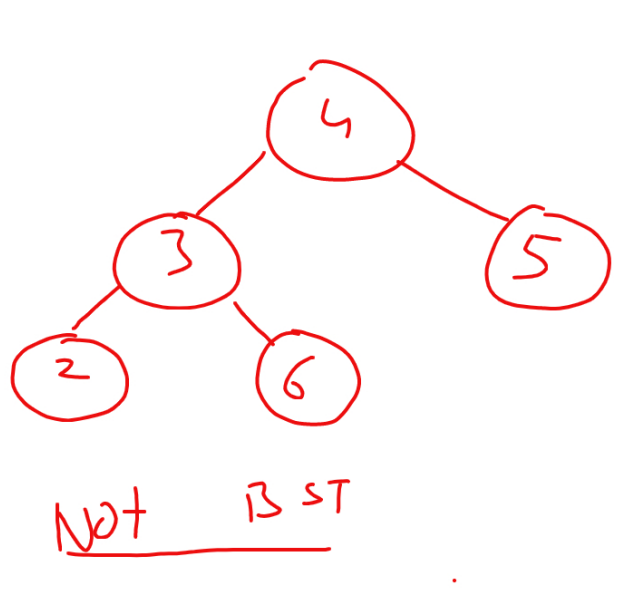


HW:
1) Construct BT from postorder & inorder
2) Iterative preorder traversal
3) Iterative postorder traversal

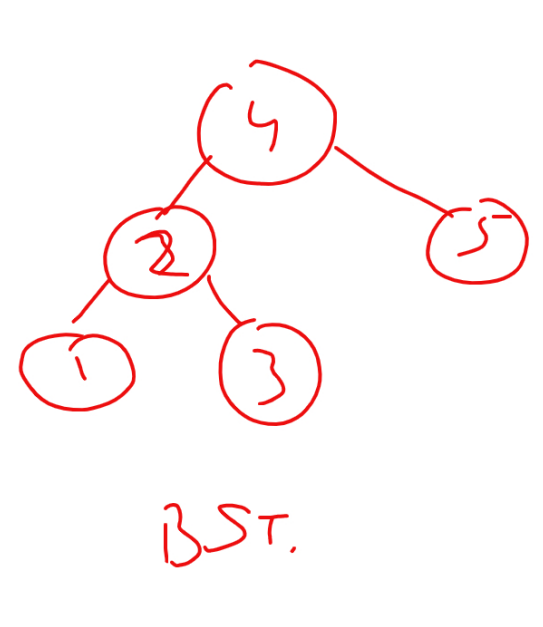
BST

1) The left subtree of a node contains values only less than the node.
2) The right subtree of a node contains values only greater than the node.
3) The left & the right subtrees should also be BSTs.

NOTE: BSTs should not have duplicate values.



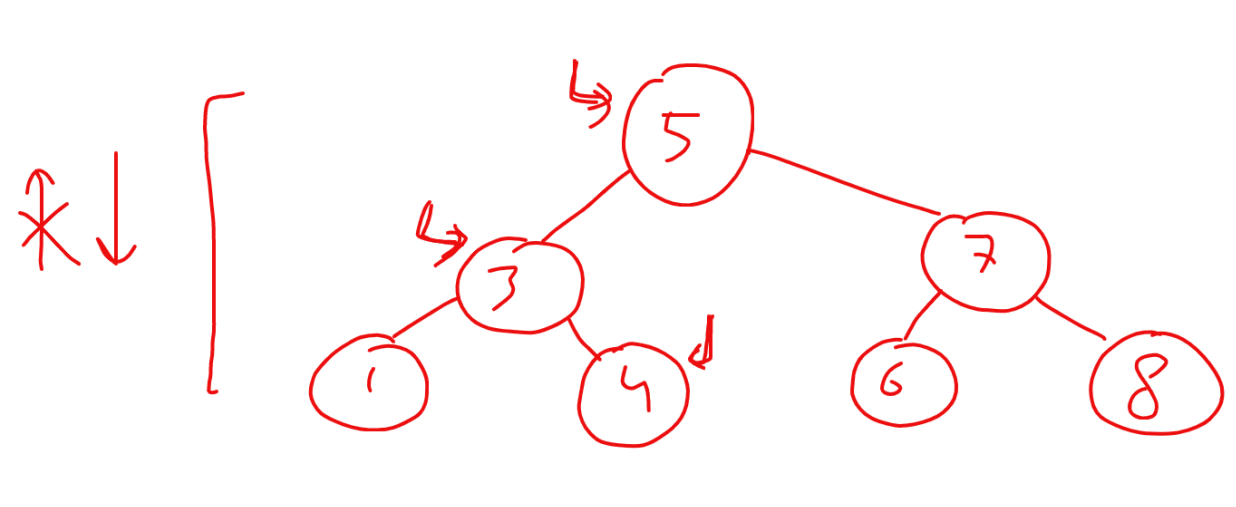
Not BST



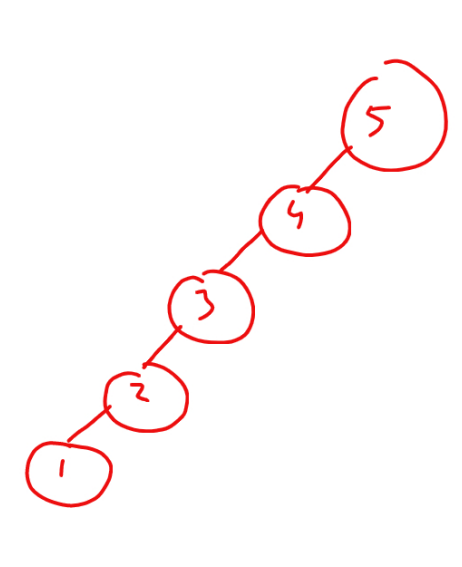
BST.

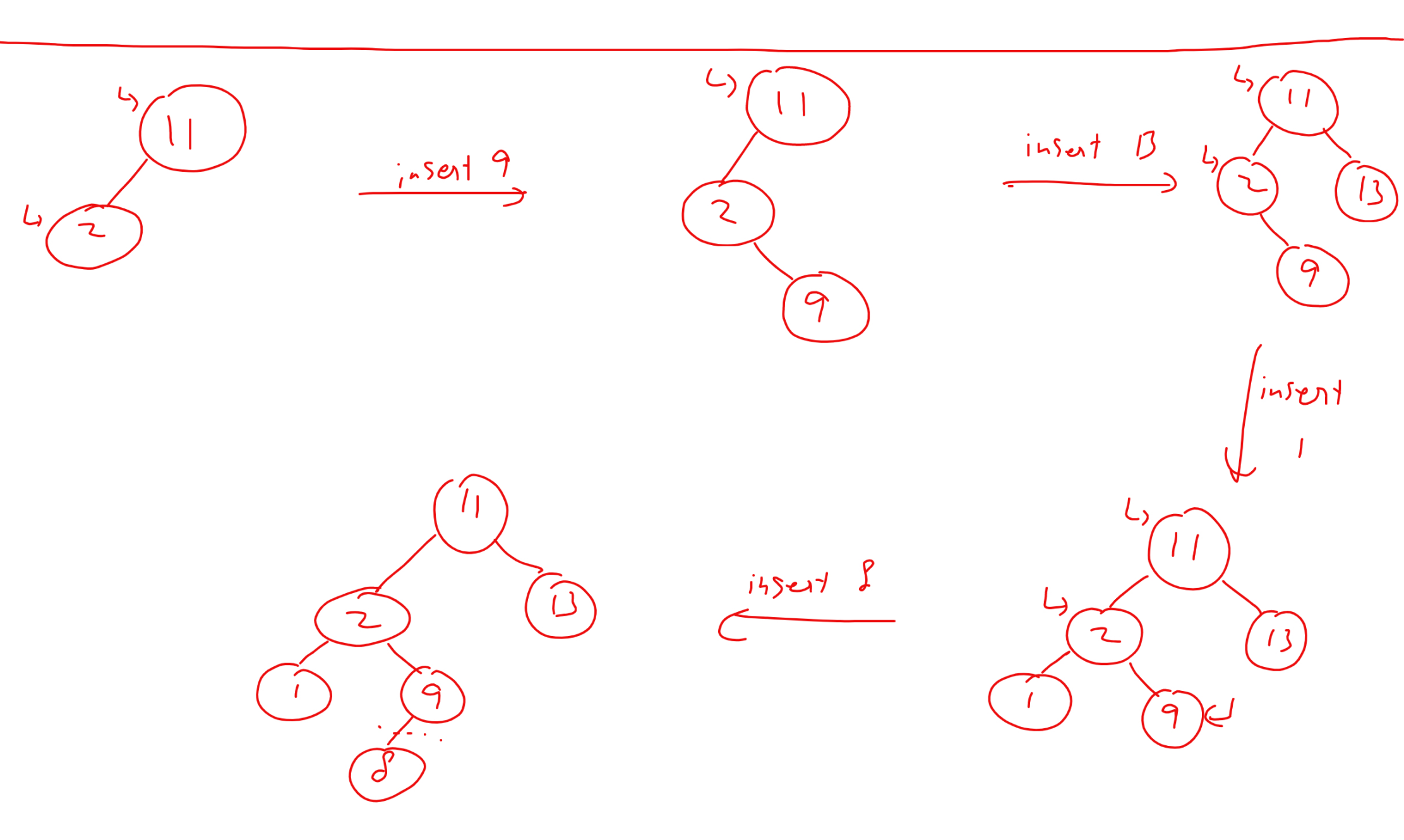
inorder: [1, 2, 3, 4, 5]

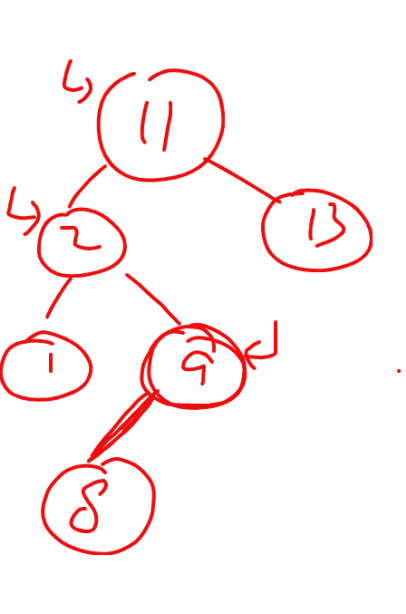
NOTE: Inorder traversal of a BST is always sorted. #



(search for 4)



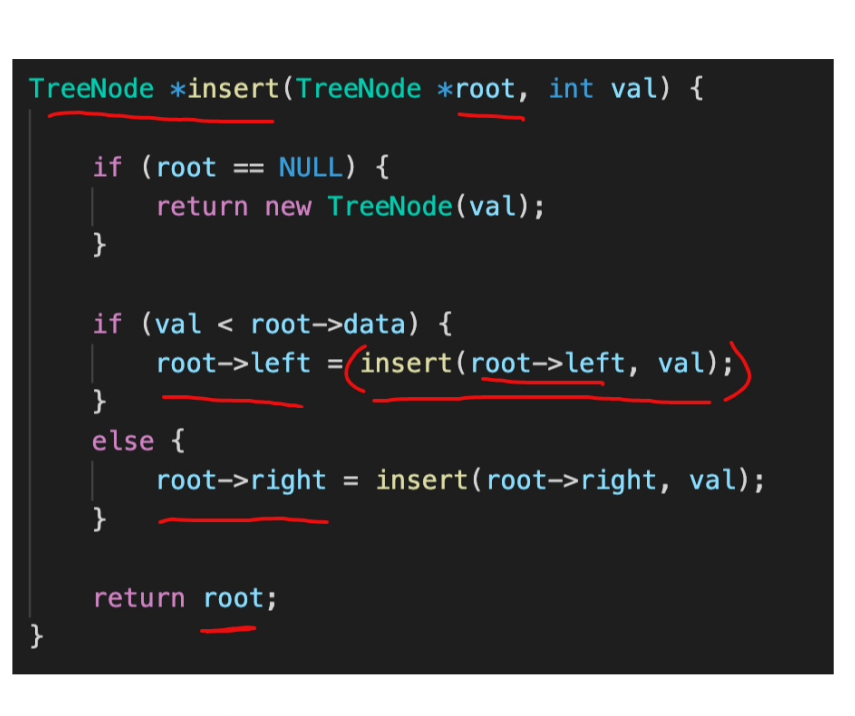

11
2
insert 9
11
2
9
insert 13
11
2
13
9
insert 1
11
2
13
1
9
insert 8
11
2
13
1
9
8



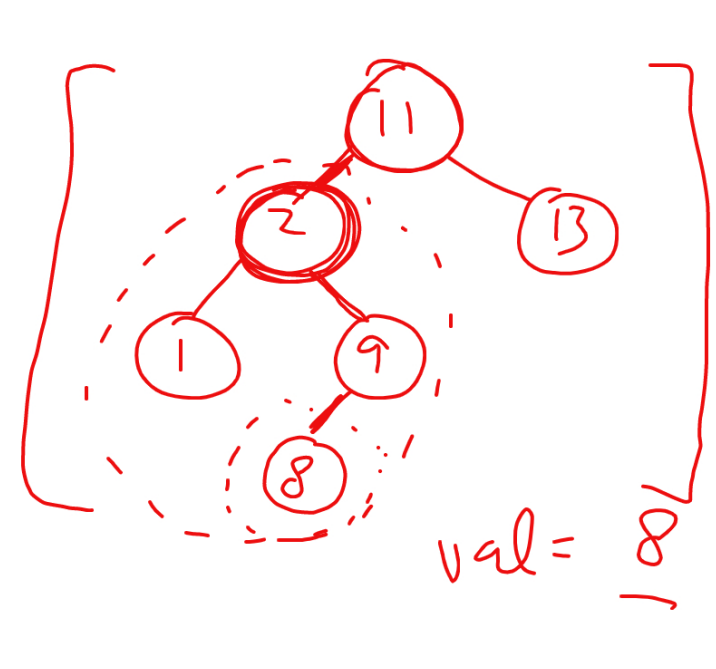
val = 8

8

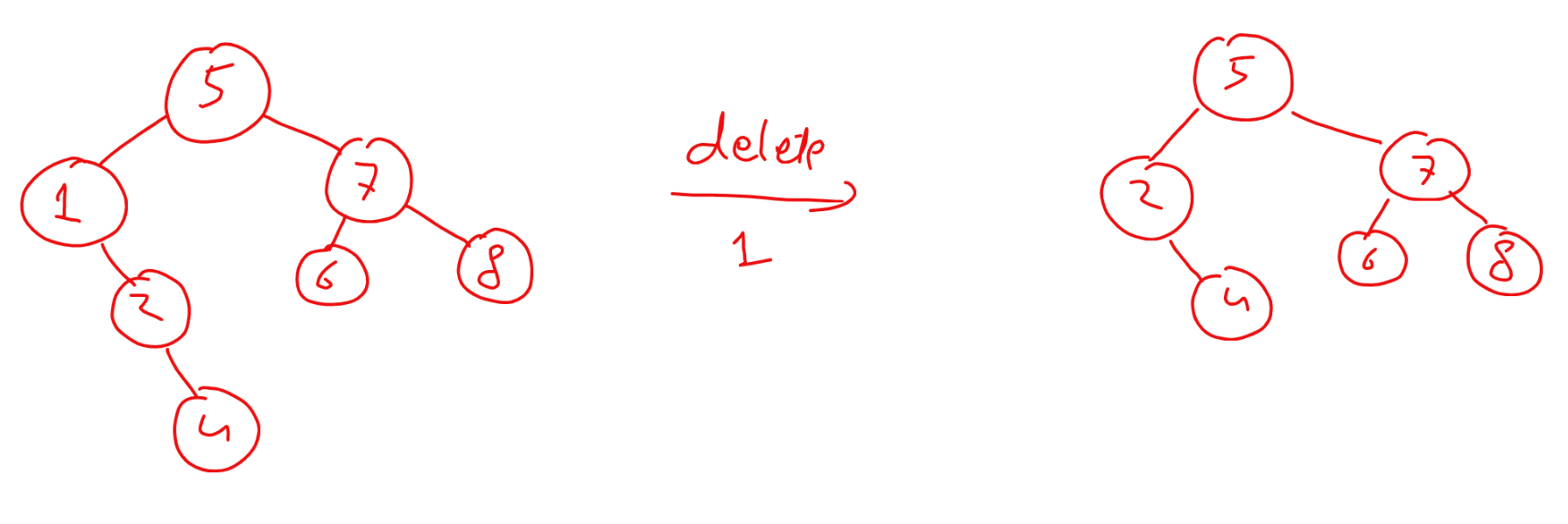
[9→left = 8]



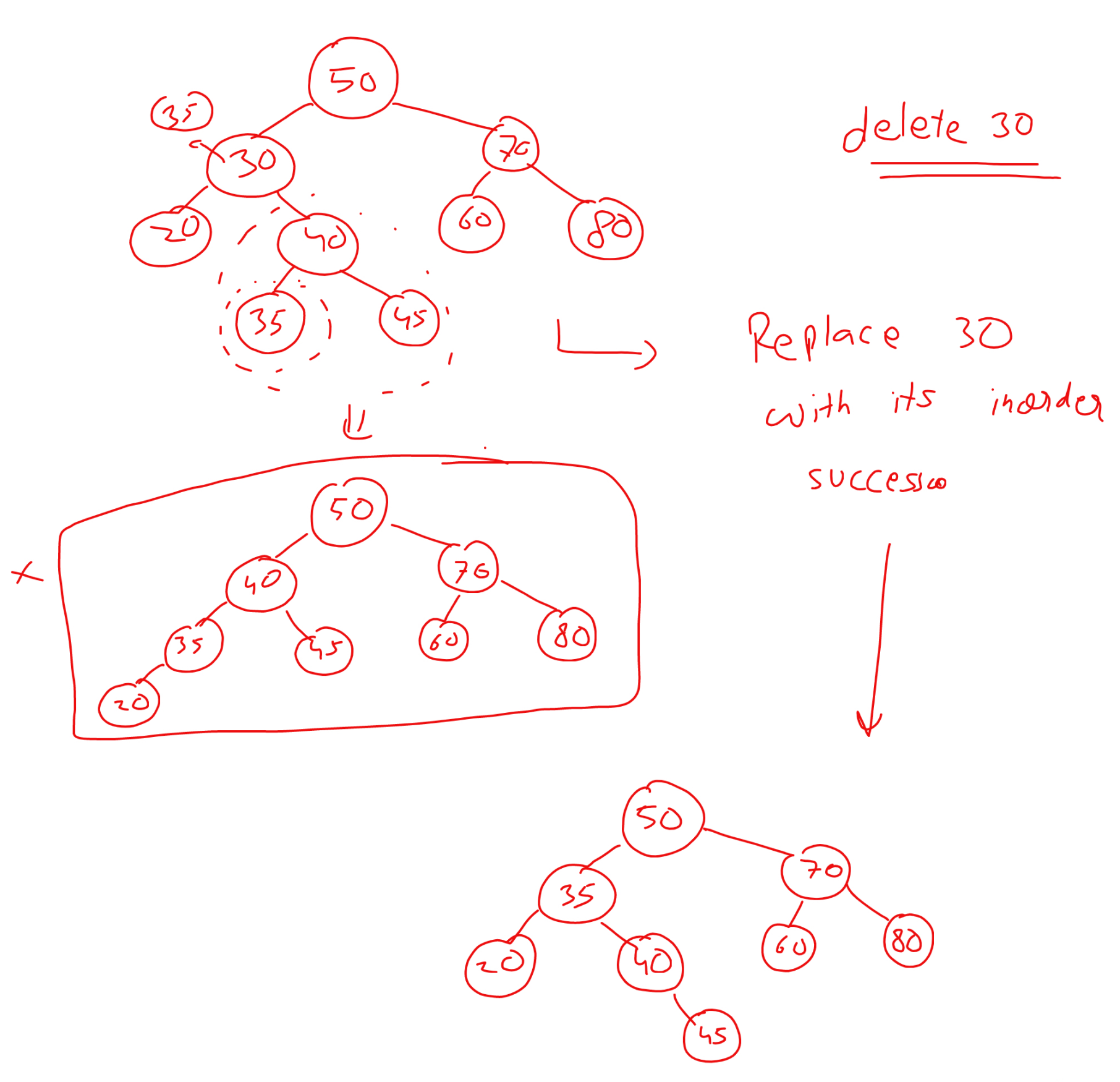
```
TreeNode *insert(TreeNode *root, int val) {
    if (root == NULL) {
        return new TreeNode(val);
    }

    if (val < root->data) {
        root->left = insert(root->left, val);
    }
    else {
        root->right = insert(root->right, val);
    }

    return root;
}
```


11
2
13
1
9
8
val = 8

11
root = 11
11→left ↓ 2
root = 2
2→right ↓ 9
root = 9
9→left ↓ 8
root = NULL


5
1
7
2
6
8
4
delete 1
5
2
7
4
6
8

CASE-1: No child node: Snap-off the branch
CASE-2: One child node: Replace the node with its non-NULL child
CASE-3: Two child nodes:

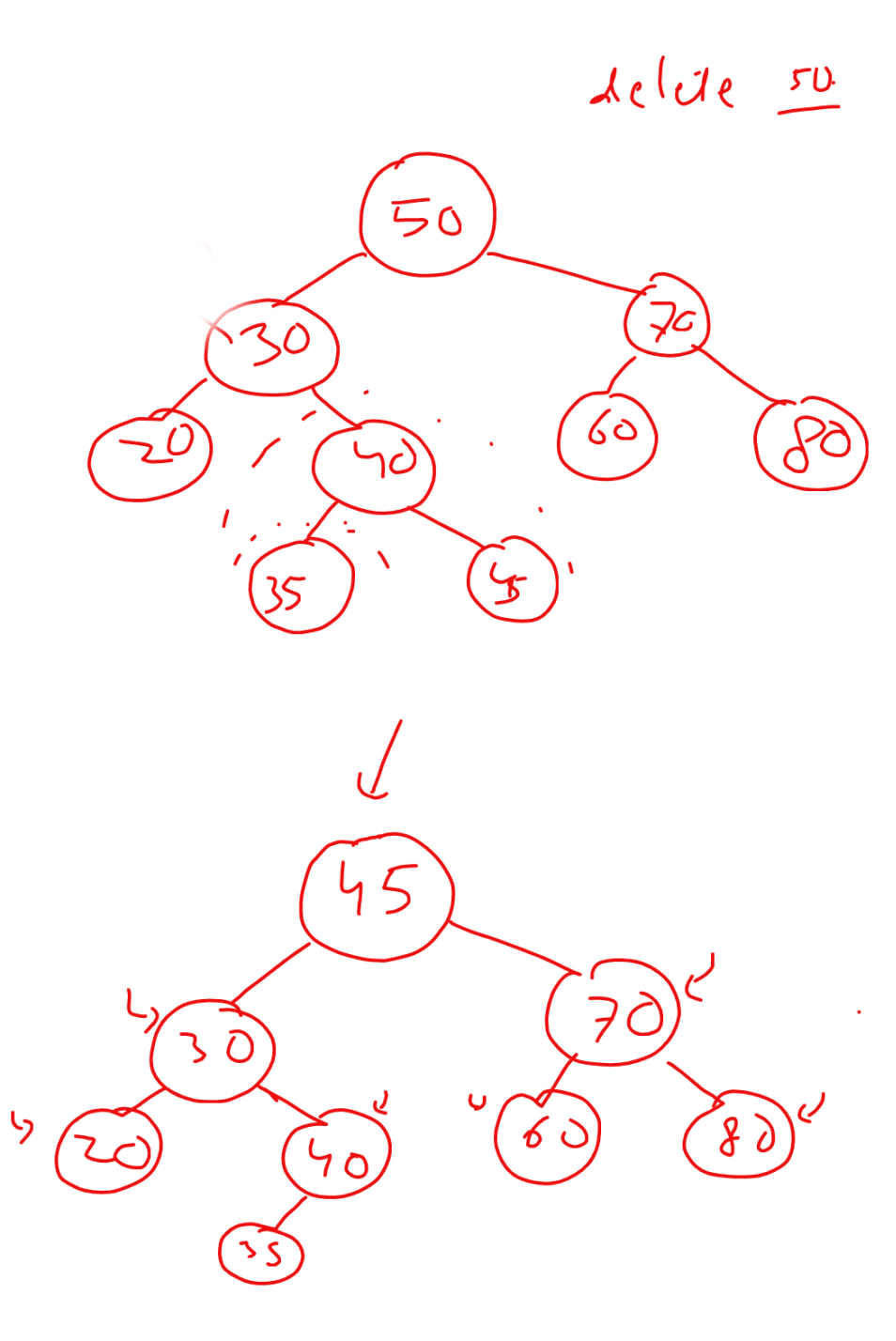

50
30
70
35
20
40
60
80
35
45
50
40
70
35
45
60
80
20
50
35
70
20
40
60
80
45

delete 30

Replace 30 with its inorder successor

→ Search for the inorder successor & replace the node's value with the value of inorder successor.
→ Delete the inorder successor.

delete 50

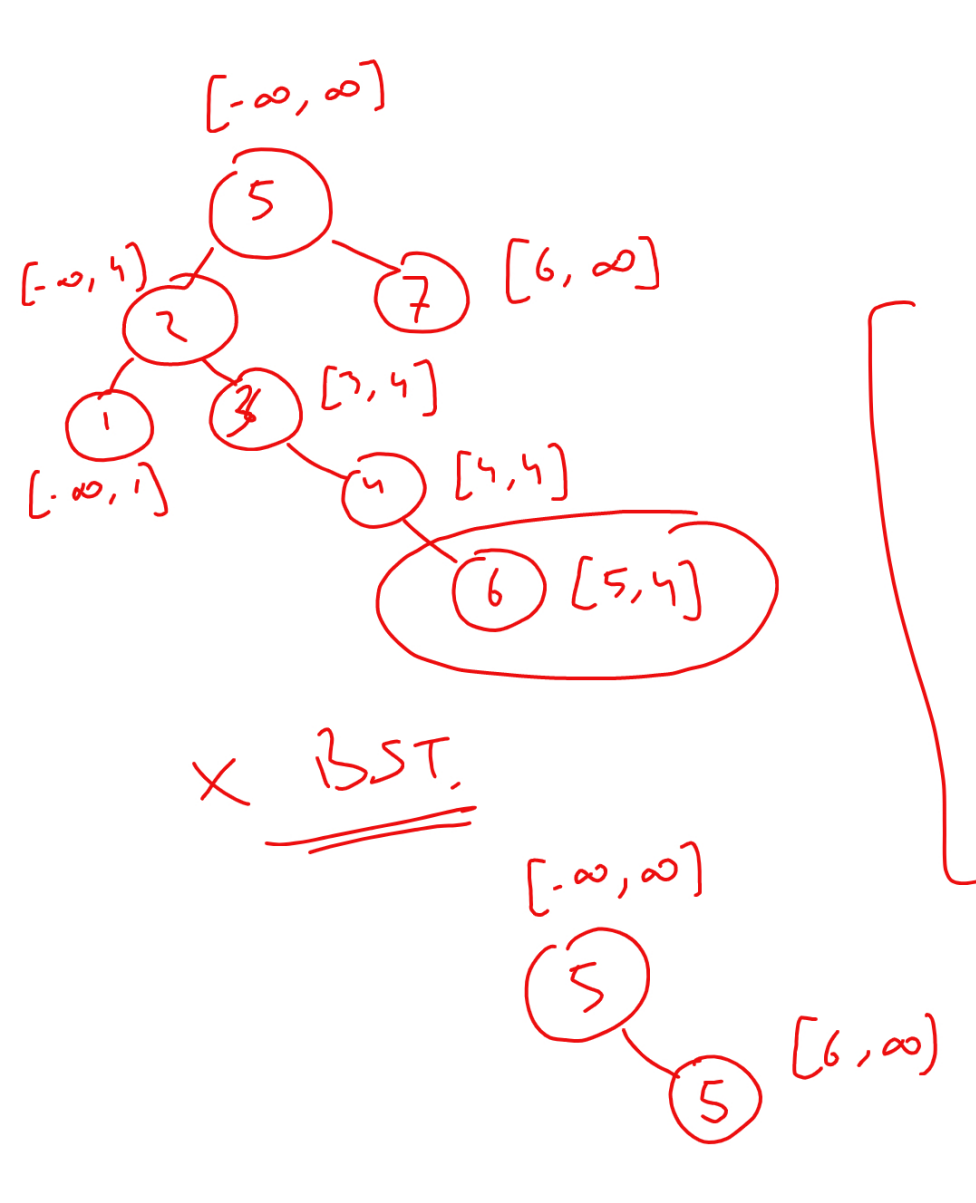

50
30
70
20
40
60
80
35
45
45
30
70
20
40
60
80
35

We can also replace the node with its inorder predecessor.

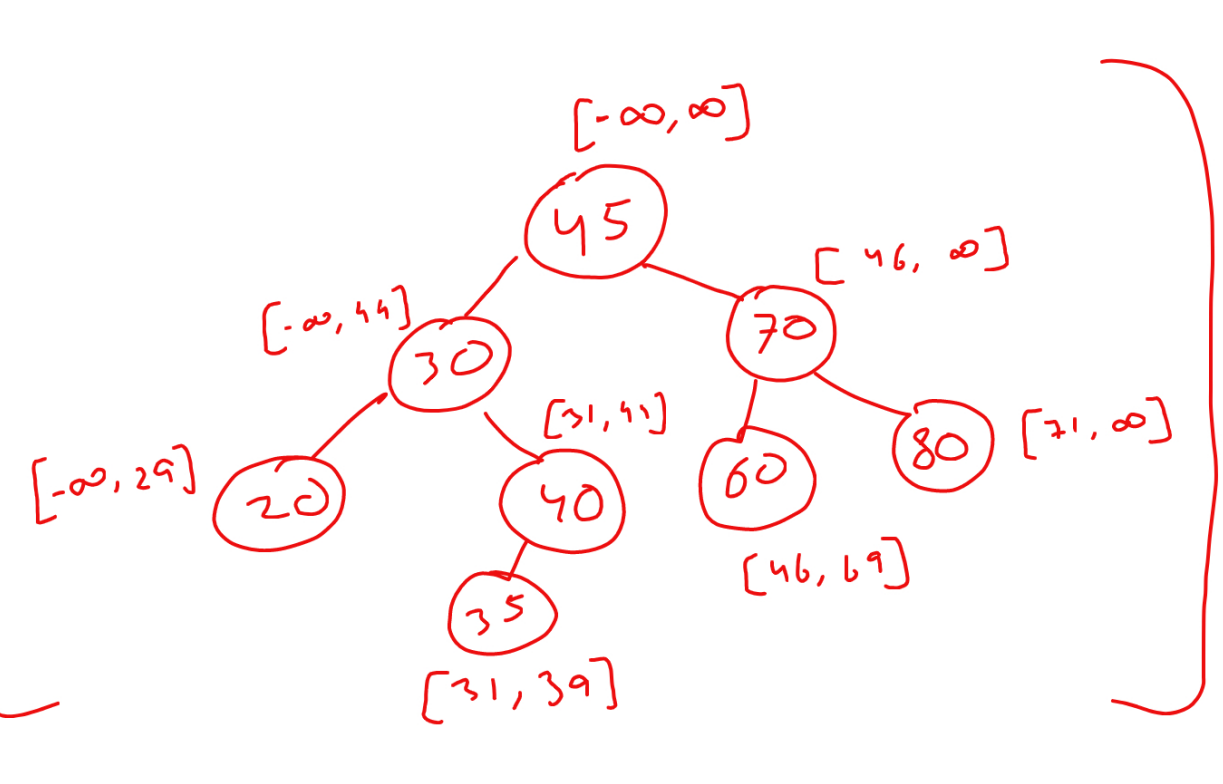
[20, 30, 40, 35]

O(n)

[TC: O(n²)]


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

× BST.


[-∞, ∞]
45
[-∞, 44]
30
[46, ∞]
70
[-∞, 29]
20
[31, 44]
40
35
[31, 39]
60
[46, 69]
80
[71, ∞]

✓ BST.