

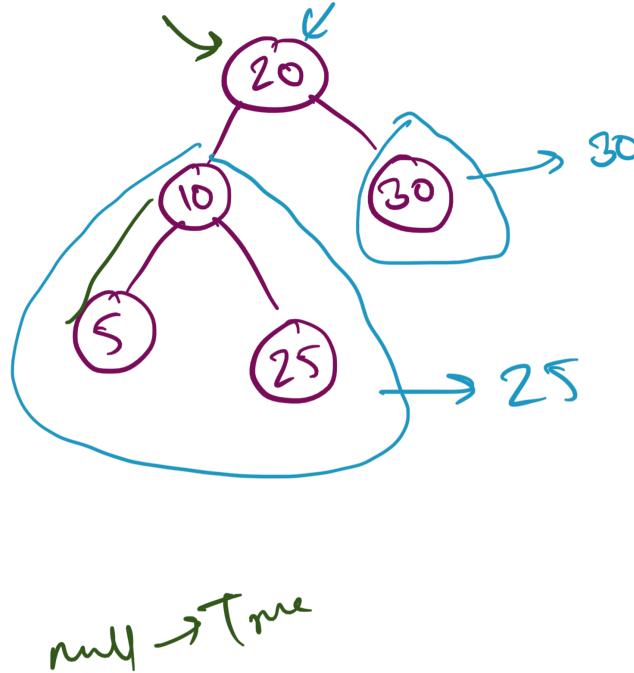
# Binary Search Tree - II



---

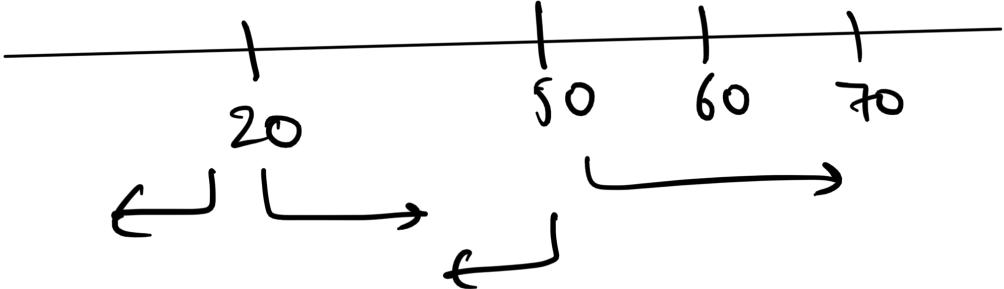
## Check if a Tree is BST or Not

10



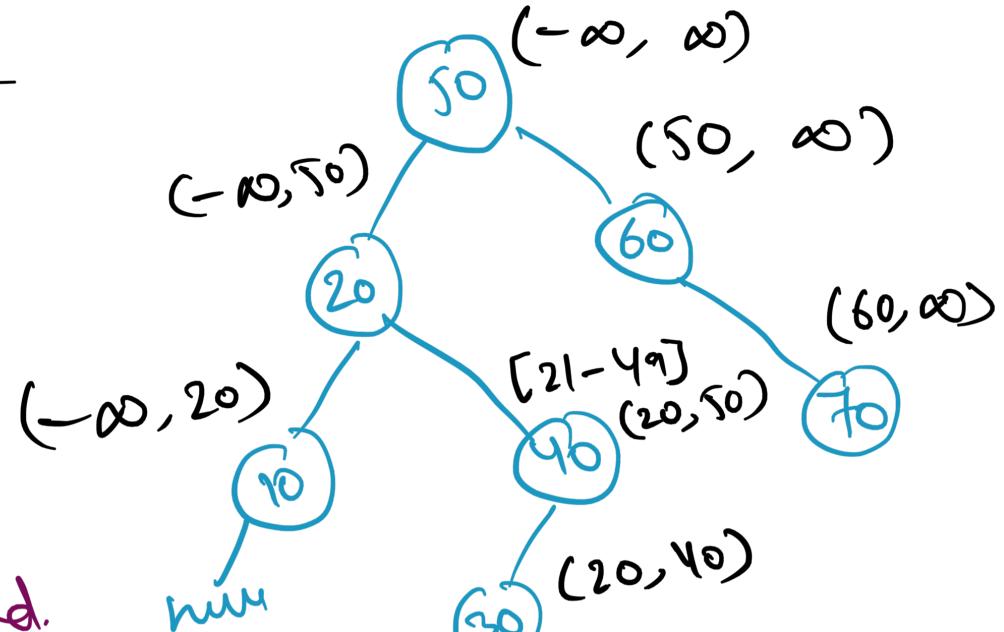
$\Rightarrow$  1. find the smallest element in right sub-tree.  
 $\Leftarrow$  2. find the largest element in left sub-tree  
node  $>$  ll and  
node  $<$  rr  
True for every node  $\hookrightarrow$  BST ✓

$\swarrow$  is Valid BST (root) &  
int ll = getMin(root.left);  $\rightarrow O(N)$   
int rr = getMax(root.right);  $\rightarrow O(N)$   
if (root.val  $\leq$  ll || root.val  $\geq$  rr) return false;  
return is Valid BST (root.left) && is Valid BST (root.right);  
 $\Downarrow$   $\rightarrow O(N^2)$



when we go

- ① left → update upper bound
- ② right → update lower bound.



$$50 \leq -\infty \rightarrow \text{false}$$

$$50 \geq \infty$$

10

Array  
(prev)

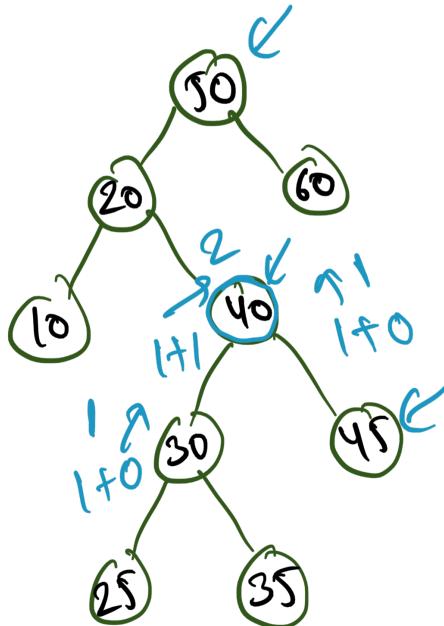
## 2<sup>nd</sup> Method

Maintain a prev value while doing the Inorder Traversal.

L N R

~~H. W.~~

## Shortest distance between two nodes in BST

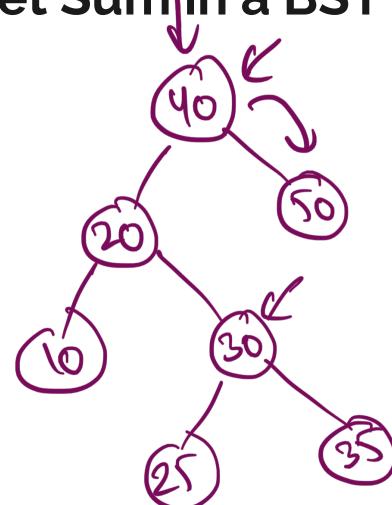
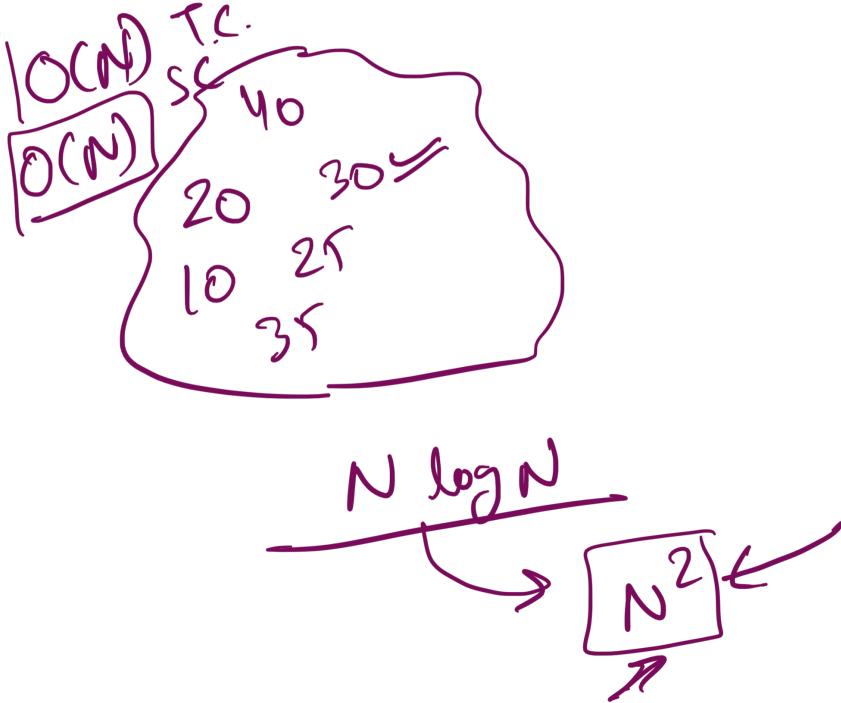


(25, 45)

50  
→ 20  
→ 40

$$2 + 1 = \underline{\underline{3}}$$

## Find a Pair with the Given Target Sum in a BST



$$k = 80$$

20

---

## Practice Problems

1. Find the Floor and Ceil Values in a BST
2. Construct BST from given preorder traversal
3. <https://www.interviewbit.com/courses/programming/tree-data-structure>