

98. Validate Binary Search Tree

Medium

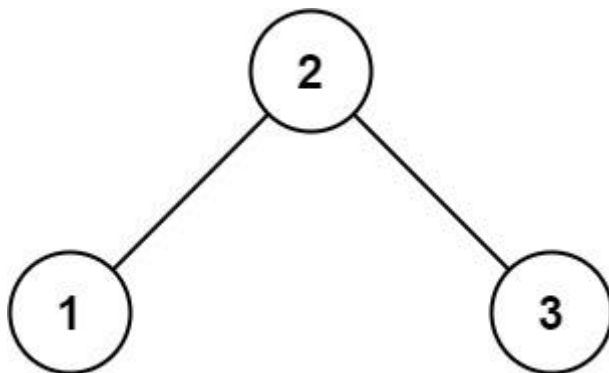
6523719Add to ListShare

Given the `root` of a binary tree, *determine if it is a valid binary search tree (BST)*.

A **valid BST** is defined as follows:

- The left subtree of a node contains only nodes with keys **less than** the node's key.
- The right subtree of a node contains only nodes with keys **greater than** the node's key.
- Both the left and right subtrees must also be binary search trees.

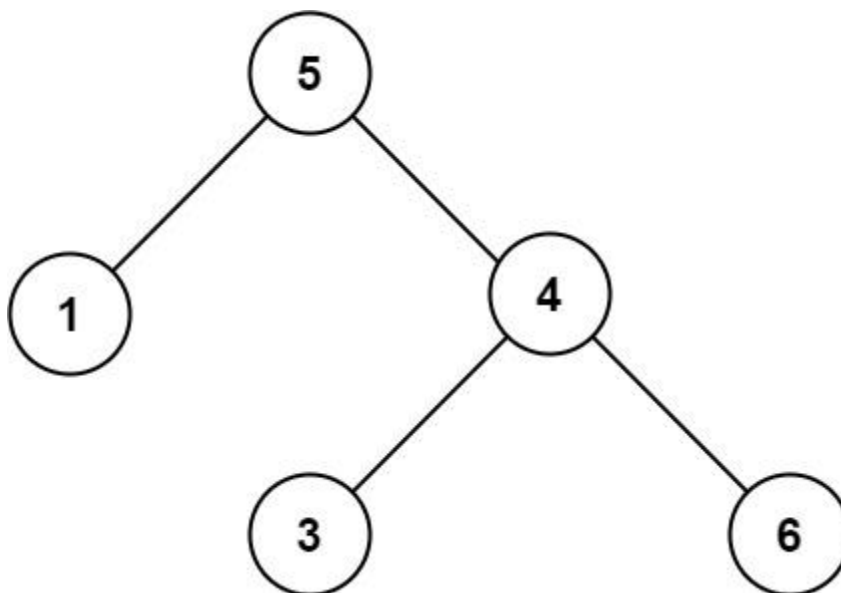
Example 1:



Input: root = [2,1,3]

Output: true

Example 2:



Input: root = [5,1,4,null,null,3,6]

Output: false

Explanation: The root node's value is 5 but its right child's value is 4.

Constraints:

- The number of nodes in the tree is in the range $[1, 10^4]$.
- $-2^{31} \leq \text{Node.val} \leq 2^{31} - 1$

SOLUTION

```
class Solution {  
  
    public boolean isValidBST(TreeNode root) {  
  
        return isValidBST(root, Long.MIN_VALUE, Long.MAX_VALUE);  
  
    }  
  
    public boolean isValidBST(TreeNode root, long min, long max){  
  
        if(root == null) return true;  
  
        if(root.val >= max || root.val <= min) return false;  
  
        return isValidBST(root.left, min, root.val) && isValidBST(root.right, root.val, max);  
  
    }  
}
```

OUTPUT

Testcase

Run Code Result

Debugger

Accepted

Runtime: 0 ms

Your input

[2,1,3]

Output

true

☐ Diff

Expected

true

< Prev

2/225

Next >

Console

[Use Example Testcases](#)

Run Code

Submit

Show all

22:39
02-07-2021