938. Range Sum of BST

Tuesday, December 14, 2021 8:25 PM

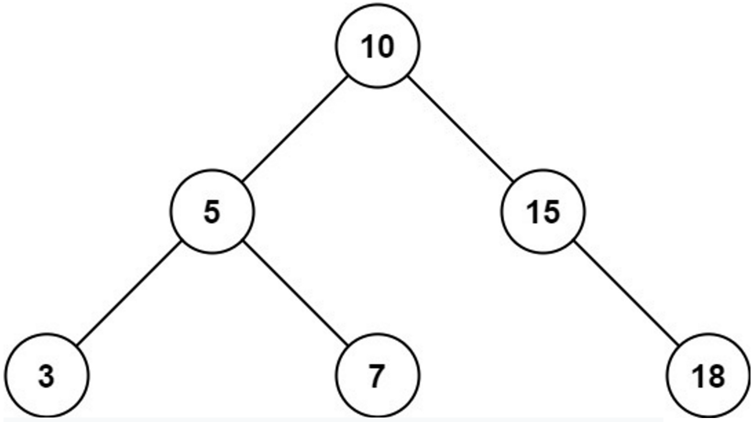
938. Range Sum of BST

Easy

3448318Add to ListShare

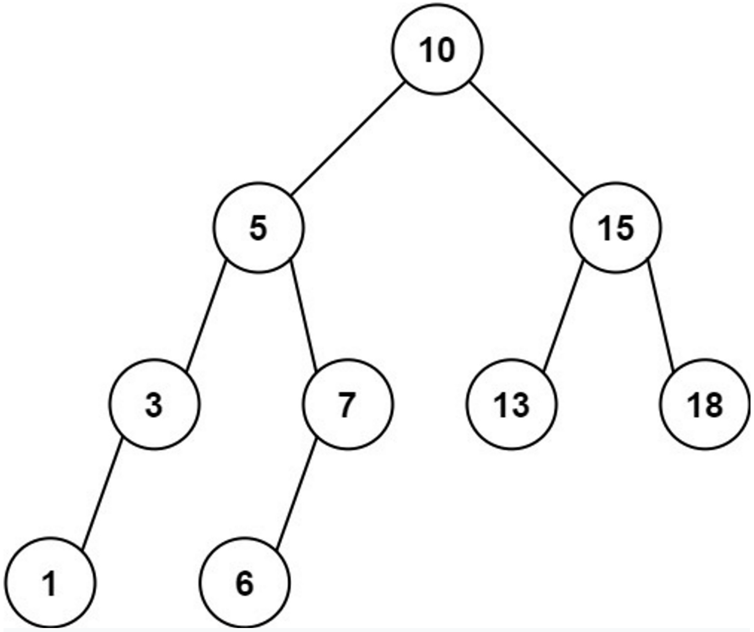
Given the root node of a binary search tree and two integers low and high, return *the sum of values of all nodes with a value in the* ***inclusive*** *range* [low, high].

Example 1:



Input: root = [10,5,15,3,7,null,18], low = 7, high = 15
Output: 32
Explanation: Nodes 7, 10, and 15 are in the range [7, 15]. 7 + 10 + 15 = 32.

Example 2:



Input: root = [10,5,15,3,7,13,18,1,null,6], low = 6, high = 10
Output: 23
Explanation: Nodes 6, 7, and 10 are in the range [6, 10]. 6 + 7 + 10 = 23.

Constraints:

- The number of nodes in the tree is in the range [1, 2 * 104].
- 1 <= Node.val <= 105
- 1 <= low <= high <= 105
- All Node.val are **unique**.

Accepted

509,947

Submissions

603,751

From <https://leetcode.com/problems/range-sum-of-bst/>

BASIC SOLUTION

```
/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode() {}
 *     TreeNode(int val) { this.val = val; }
 *     TreeNode(int val, TreeNode left, TreeNode right) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }
 */
```

```
class Solution {
    int sum =0;
    public int rangeSumBST(TreeNode root, int low, int high) {

 getSum(root,low,high);



        return  sum;

    }

     public  void getSum(TreeNode root, int low, int high) {
              if (root==null)
            return;
        getSum(root.left,low,high);
```

```
         if (root.val>=low && root.val<=high) {

            sum=sum+root.val;
            System.out.println(root.val+" Sum =: "+sum);


        }
        getSum(root.right,low,high);

        return;

    }

}
```

BEST SOLUTION

```
/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode() {}
 *     TreeNode(int val) { this.val = val; }
 *     TreeNode(int val, TreeNode left, TreeNode right) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }
 */
class Solution {

    public int rangeSumBST(TreeNode root, int l, int h) {
         if(root==null)
             return 0;

           if (root.val<l )
               return rangeSumBST(root.right,l,h);

               if( root.val>h) {
                return rangeSumBST(root.left,l,h);

           }
```

```
        return  root.val+ rangeSumBST(root.left,l,h)+ rangeSumBST(root.right,l,h);

    }


}
```