



链滴

[每日 LeetCode] 938. Range Sum of BST

作者: [Hanseltu](#)

原文链接: <https://ld246.com/article/1557671381141>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

原文链接 [\[每日LeetCode\] 938. Range Sum of BST](#)

Description:

Given the **root** node of a binary search tree, return the sum of values of all nodes with value between **L** and **R** (inclusive).

The binary search tree is guaranteed to have unique values.

Example 1:

Input: root = [10,5,15,3,7,null,18], L = 7, R = 15

Output: 32

Example 2:

Input: root = [10,5,15,3,7,13,18,1,null,6], L = 6, R = 10

Output: 23

思路：本题题为：找出一个BST中，计算在[L,R]双闭区间内的所有节点的值的和。依旧使用递归思想，递归基本条件是root为空，返回sum。另外分三种情况考虑。

- 如果root节点在[L,R]内，那么把结果加上root的值，然后再分别加上左右子树的值。
 - 如果root的值比L还小，说明左子树一定不会满足[L,R]区间，那么直接向右边找就行。
 - 如果root的值比R还大，说明右子树一定不会满足[L,R]区间，那么直接向左边找就行。
-

C++代码

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    int rangeSumBST(TreeNode* root, int L, int R) {
        return helper(root,L,R,0);
    }
    int helper(TreeNode* node,int L,int R,int sum){
        if(node==NULL)
            return sum;
        int curr = node->val;
        if(curr<L)
            return helper(node->right,L,R,sum);
        else if(curr>R)
```

```
        return helper(node->left,L,R,sum);
    else
        return helper(node->left,L,R,sum)+helper(node->right,L,R,sum)+curr;
    }
};
```

运行时间: 140ms

运行内存: 41.2M