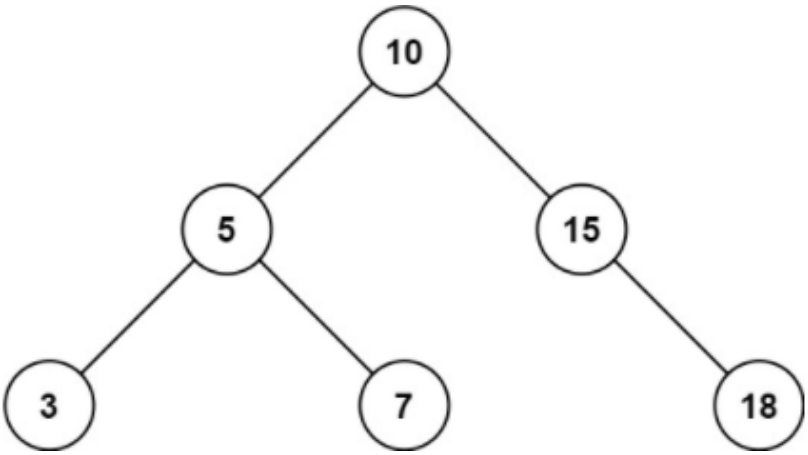
# 二叉搜索树的范围和

给定二叉搜索树的根结点 root，返回值位于范围 *[low, high]* 之间的所有结点的值的和。

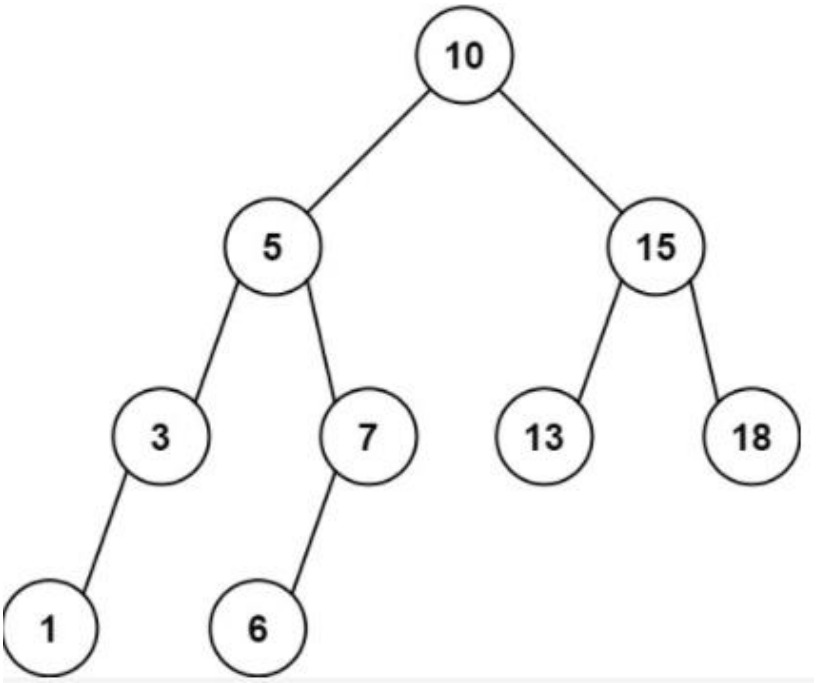
**示例 1：**




**输入：**root = [10,5,15,3,7,null,18], low = 7, high = 15

**输出：**32

**示例 2：**




**输入：**root = [10,5,15,3,7,13,18,1,null,6], low = 6, high = 10

**输出：**23

```
/**
```

```
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left
), right(right) {}
 * };
 */
class Solution {
public:
    void inordertravel(TreeNode* root,vector<int>& res)
    {
        if(root==nullptr)
        {
            return;
        }
        inordertravel(root->left,res);
        res.push_back(root->val);
        inordertravel(root->right,res);
    }
    int rangeSumBST(TreeNode* root, int low, int high) {
        if(root==nullptr)
            return 0;
        vector<int> res;
        inordertravel(root,res);
        int sum=0;
        for(int i=0;i<res.size();i++)
        {
            if(res[i]>=low&&res[i]<=high)
            {
                sum+=res[i];
            }
        }
        return sum;
    }
};
```