# 题目

给定一个二叉树，判断其是否是一个有效的二叉搜索树。

假设一个二叉搜索树具有如下特征：

节点的左子树只包含小于当前节点的数。

节点的右子树只包含大于当前节点的数。

所有左子树和右子树自身必须也是二叉搜索树。

**示例 1:**

输入:

2

/ \

1 3

输出: true

**示例 2:**

输入:

5

/ \

1 4

  / \

  3 6

输出: false

解释: 输入为: [5,1,4,null,null,3,6]。

  根节点的值为 5 ，但是其右子节点值为 4 。

# 分析

/\*\*

\* 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:

bool helper(TreeNode\* root, long long lower, long long upper) {

if (root == nullptr) return true;

if (root -> val <= lower || root -> val >= upper) return false;

return helper(root -> left, lower, root -> val) && helper(root -> right, root -> val, upper);

}

bool isValidBST(TreeNode\* root) {

return helper(root, LONG\_MIN, LONG\_MAX);

}

};