# 题目

实现一个函数，检查一棵二叉树是否为二叉搜索树。

**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

## 方法一：递归法/深度优先遍历

/\*\*

\* 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 {

private:

bool flag;

long val;

void dfs(TreeNode\* root){

if(root==nullptr||!flag){

return;

}

dfs(root->left);

if(root->val>val){

val=root->val;

}else{

flag=false;

return;

}

dfs(root->right);

}

public:

bool isValidBST(TreeNode\* root) {

flag=true;

val=LONG\_MIN;

dfs(root);

return flag;

}

};