# 题目

给你一棵所有节点为非负值的二叉搜索树，请你计算树中任意两节点的差的绝对值的最小值。

**示例：**

输入：

1

\

3

/

2

输出：

1

解释：

最小绝对差为 1，其中 2 和 1 的差的绝对值为 1（或者 2 和 3）。

提示：

树中至少有 2 个节点。

本题与 783

https://leetcode-cn.com/problems/minimum-distance-between-bst-nodes/ 相同

# 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:

void dfs(TreeNode\* root, int& prev, int& ret) {

if (root == NULL) return;

dfs(root->left, prev, ret);

if (prev >= 0) ret = min(ret, root->val - prev);

prev = root->val;

dfs(root->right, prev, ret);

}

int getMinimumDifference(TreeNode\* root) {

int prev = -1;

int ret = INT\_MAX;

dfs(root, prev, ret);

return ret;

}

};