# 题目

给定一个二叉搜索树的根节点root，返回树中任意两节点的差的最小值。

**示例：**

输入: root = [4,2,6,1,3,null,null]

输出: 1

解释:

注意，root是树节点对象(TreeNode object)，而不是数组。

给定的树 [4,2,6,1,3,null,null] 可表示为下图:

4

/ \

2 6

/ \

1 3

最小的差值是 1, 它是节点1和节点2的差值, 也是节点3和节点2的差值。

**注意：**

二叉树的大小范围在 2 到 100。

二叉树总是有效的，每个节点的值都是整数，且不重复。

本题与 530：

https://leetcode-cn.com/problems/minimum-absolute-difference-in-bst/ 相同

# 分析

## 方法一：递归法

class Solution {

public:

int min\_dif = 100;

int temp1 = -100;

int temp2 = -100;

int minDiffInBST(TreeNode\* root) {

return inorder(root);

}

int inorder(TreeNode\* r)

{

if(!r)

return min\_dif;

if(r -> left)

inorder(r -> left);

temp1 = temp2;

temp2 = r -> val;

min\_dif = min(min\_dif, temp2 - temp1);

if(r -> right)

inorder(r -> right);

return min\_dif;

}

};

## 方法二：深度优先遍历（推荐）

思路：

代码：

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 minDiffInBST(TreeNode\* root) {

int prev = -1;

int ret = INT\_MAX;

dfs(root, prev, ret);

return ret;

}

};