

## 二叉搜索树的最小绝对差

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）。

```
/**
 * 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:
    //定义全局
    vector<int> res; //存储遍历到所有节点值
    void travel(TreeNode* root)
    {
        if(root==nullptr)
            return;
```

```
        travel(root->left);
        res.push_back(root->val);
        travel(root->right);
    }
    int getMinimumDifference(TreeNode* root) {
        travel(root);
        int minans=INT_MAX;
        for(int i=0;i<res.size()-1;i++)
        {
            minans=min(minans,res[i+1]-res[i]);
        }
        return minans;
    }
};
```