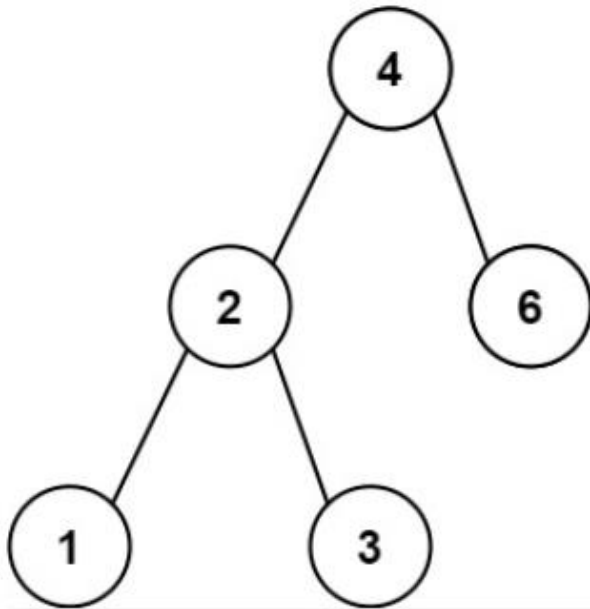


二叉搜索树节点最小距离

给你一个二叉搜索树的根节点 `root`，返回 树中任意两不同节点值之间的最小差值。

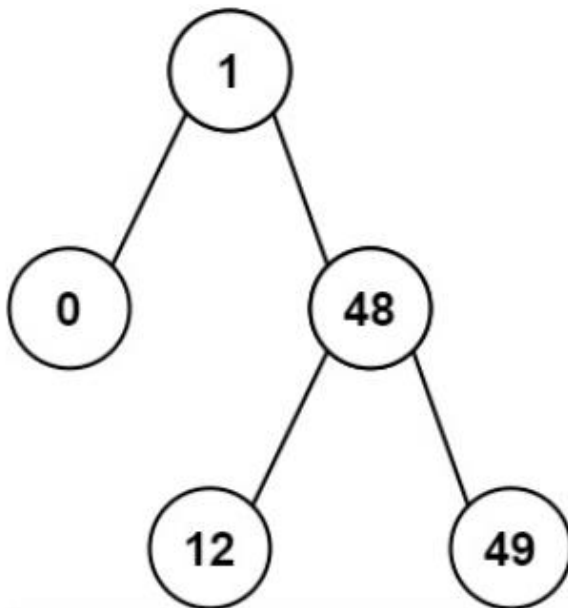
示例 1:



输入: `root = [4,2,6,1,3]`

输出: 1

示例 2:



输入: `root = [1,0,48,null,null,12,49]`

输出：1

提示：

- 树中节点数目在范围 $[2, 100]$ 内
- $0 \leq \text{Node.val} \leq 10^5$
- 差值是一个正数，其数值等于两值之差的绝对值

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left), right(right) {}
 * };
 */
class Solution {
public:
    void travel(TreeNode* root, vector<int>& res)
    {
        if(root==nullptr)
            return;
        res.push_back(root->val); // 前序遍历 根左右
        travel(root->left, res);
        travel(root->right, res);
    }
    int minDiffInBST(TreeNode* root) {
        vector<int> res;
        travel(root, res);
        sort(res.begin(), res.end());
        int min_num = res[res.size()-1];
        for(int i=0; i<res.size()-1; i++)
        {
            min_num = min(min_num, res[i+1]-res[i]);
        }
        return min_num;
    }
};
```