

给定一个二叉搜索树的根结点 `root`, 返回树中任意两节点的差的最小值。

示例:

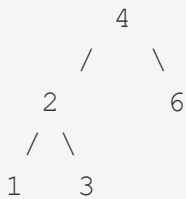
输入: `root = [4,2,6,1,3,null,null]`

输出: 1

解释:

注意, `root`是树结点对象(`TreeNode object`), 而不是数组。

给定的树 `[4,2,6,1,3,null,null]` 可表示为下图:



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:

1. 二叉树的大小范围在 2 到 100。
2. 二叉树总是有效的, 每个节点的值都是整数, 且不重复。

思路, 二叉搜索树如果用中序遍历, 那么其遍历的队列, 是递增的数组, 在动态遍历波数组, 即可得到两节点差最小的值。

```
/**
 * 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, vector<int> &nums) {
        if (root->left != NULL) {
            dfs(root->left, nums);
```

```

    }
    nums.push_back(root->val);
    if (root->right != NULL) {
        dfs(root->right, nums);
    }
}

int minDiffInBST(TreeNode* root) {
    vector<int> nums;
    dfs(root, nums);
    int min=nums[1]-nums[0];
    for (int i=2; i<nums.size(); i++) {
        if (nums[i]-nums[i-1]<min) {
            min=nums[i]-nums[i-1];
        }
    }
    return min;
}

};

```