```cpp
/**
 * 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:
    bool isValidBST(TreeNode* root) {

        vector<int> ordered_tree = bianliBST(root);
        int i;
        for (i = 1; i < ordered_tree.size(); i++) {
            if (ordered_tree[i] <= ordered_tree[i - 1]) {
                break;
            }
        }
        if (i < ordered_tree.size()) {
            return false;
        } else {
            return true;
        }
    }
    // 递归遍历二叉搜索树
    vector<int> bianliBST(TreeNode* root) {
        vector<int> result;
        if (root != nullptr) {
            // 先遍历左子树
            vector<int> temp = bianliBST(root->left);
            // 合并左子树结果
            result.insert(result.end(), temp.begin(), temp.end());
            // 访问根节点
            result.push_back(root->val);
            // 然后遍历右子树
            vector<int> temp1 = bianliBST(root->right);
            // 合并右子树结果
            result.insert(result.end(), temp1.begin(), temp1.end());
        }
        return result;
    }
};```

```