```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:
    int kthSmallest(TreeNode* root, int k) {
        // 深度有效搜索遍历二叉搜索树
        vector<int> order_tree = bianliBST(root);
        int result = 0;

        // 返回第 k 小的元素，注意 k 是从 1 开始的
        if (k > 0 && k ≤ order_tree.size()) {
            result = order_tree[k - 1];
        }
        return result;
    }

    // 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;
    }
};```
```