# 题目

给定一个二叉搜索树，编写一个函数 kthSmallest 来查找其中第 k 个最小的元素。

**说明：**

你可以假设 k 总是有效的，1 ≤ k ≤ 二叉搜索树元素个数。

**示例 1:**

输入: root = [3,1,4,null,2], k = 1

3

/ \

1 4

\

  2

输出: 1

**示例 2:**

输入: root = [5,3,6,2,4,null,null,1], k = 3

5

/ \

3 6

/ \

2 4

/

1

输出: 3

进阶：

如果二叉搜索树经常被修改（插入/删除操作）并且你需要频繁地查找第 k 小的值，你将如何优化 kthSmallest 函数？

# 分析

/\*\*

\* 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 n=0;

int res;

int kthSmallest(TreeNode\* root, int k) {

//递归

dfs(root,k);

return res;

}

void dfs(TreeNode\* root,int k){

if(!root)return ;

dfs(root->left,k);

n++;

if(n==k) res=root->val;

dfs(root->right,k);

}

};