1. Kth Smallest Element in a BST

Given a binary search tree, write a function kthSmallest to find the **k**th smallest element in it.

**Example 1:**

Input: root = [3,1,4,null,2], k = 1  
 3  
 / \  
 1 4  
 \  
 2  
Output: 1

**Example 2:**

Input: root = [5,3,6,2,4,null,null,1], k = 3  
 5  
 / \  
 3 6  
 / \  
 2 4  
 /  
 1  
Output: 3

**Follow up:** What if the BST is modified (insert/delete operations) often and you need to find the kth smallest frequently? How would you optimize the kthSmallest routine?

**Constraints:**

* The number of elements of the BST is between 1 to 10^4.
* You may assume k is always valid, 1 ≤ k ≤ BST's total elements.

**解法1** 利用BST中序序列有序的特点找第k次访问的节点

class Solution {  
public:  
 int kthSmallest(TreeNode\* root, int k) {  
 int order = 0, ans = INT\_MAX;  
 bool flag = false;  
 in\_order(root, order, k, flag, ans);  
 return ans;  
 }  
 void in\_order(TreeNode\* root, int &order, int k, bool &flag, int &ans){  
 if(flag)return;  
 if(!root)return;  
 in\_order(root->left, order, k, flag, ans);  
 order++;  
 if(order == k){  
 flag = true;  
 ans = root->val;  
 }  
 in\_order(root->right, order, k, flag, ans);  
 }  
};

**解法2** 将BST修改成为中序的线索二叉树