```
/*
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？
来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/kth-smallest-element-in-a-bst
著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
*/
```

分析:
二叉搜索树的中序遍历的结果就是升序的,我们只需要对原二叉搜索树进行遍历(同时计数),只要计数到第k个返回即可.

方法一:C++_中序遍历计数

```
/**
 * 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
{
    private:
        int i       = 0;
        int kk      = 0;
        int ret_val = 0;
        void __midOrderDfs(TreeNode* node)
        {
```

```
            if(node==NULL)
            {
                return ;
            }
            __midOrderDfs(node->left);

            if(++i == kk)
            {
                ret_val = node->val;
                return;
            }

            __midOrderDfs(node->right);
        }
    public:
        int kthSmallest(TreeNode* root, int k)
        {
            ret_val = 0;
            i       = 0;
            kk      = k;
            __midOrderDfs(root);
            return ret_val;
        }
};
/*
执行结果:
通过
显示详情
执行用时 :16 ms, 在所有 C++ 提交中击败了99.56% 的用户
内存消耗 :21.2 MB, 在所有 C++ 提交中击败了98.83%的用户
*/
```

AlimyBreak
2019.10.08