

## 230. 二叉搜索树中第K小的元素

题目：230. Kth Smallest Element in a BST

语言：python3

英文版链接：<https://leetcode.com/problems/kth-smallest-element-in-a-bst/description/>

中文版链接：<https://leetcode-cn.com/problems/kth-smallest-element-in-a-bst/description/>

### 题目分析

我们知道二分查找数(BST)的性质——任何一个节点的值均大于左子树的任意节点值，而小于右子树的任一节点值。那么这样就可以知道最小值的一个节点在树的最左端，最大值的一个节点在树的最右端。树从小到大顺序刚好满足树的中序遍历。因而，我们可以用中序遍历来处理。

由于 k 是个基本类型的数，我们知道它与应用类型不同，本轮递归的 k 值改变不会引起下一轮的改变，那么我们的处理办法可以增加一个全局变量、增加一个引用变量或增加一个方法形参，用这样的参数记录目前遍历到的是第几小的数。

### 答案

```
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, x):
#         self.val = x
#         self.left = None
#         self.right = None

class Solution:
    def kthSmallest(self, root: TreeNode, k: int) -> int:
        self.cnt, self.val = 0, 0
        self.inOrder(root, k)
        return self.val

    def inOrder(self, root: TreeNode, k: int):
        if not root:
            return
        self.inOrder(root.left, k)
        self.cnt += 1
        if self.cnt == k:
            self.val = root.val
            return
        self.inOrder(root.right, k)
```