

剑指54 二叉搜索树的第k大节点

二叉树的先右后左的中序遍历。

递归+剪枝

```
1 我的答案，乱七八糟：
2 比较别扭，必须假设val都是自然数。
3
4 class Solution {
5 public:
6     int DFS(TreeNode* node, int k, int &index) {
7         if (!node) return -1;
8         int value = DFS(node->right, k, index);
9         if (value != -1) return value;
10        if (index == k) return node->val;
11        index++;
12        value = DFS(node->left, k, index);
13        if (value != -1) return value;
14        return -1; // 用-1表示未找到第k大的。
15    }
16
17    int kthLargest(TreeNode* root, int k) {
18        int index = 1;
19        return DFS(root, k, index);
20    }
21 };
```

优化：

第k大的，即只需要用k作为index，不断减减，遍历到0说明找到了。

DFS函数无需返回值，用成员变量存储最终结果。

```
1 // 优化后：
2 class Solution {
3 public:
4     int m_res, m_k;
5     void DFS(TreeNode* node) {
6         if (!node) return;
7         DFS(node->right);
8         // 当m_k == 0时，说明右子树中已经找到了，无需进行后面操作。
9         if (m_k == 0) return;
10        if (--m_k == 0) {
11            m_res = node->val;
12            return;
13        }
14        DFS(node->left);
15    }
16
17    int kthLargest(TreeNode* root, int k) {
18        m_k = k;
```

```
19     DFS(root);
20     return m_res;
21 }
22 };
```