

剑指Offer（六十二）：二叉搜索树的第k个结点

🕒 2018年1月31日 09:55:02 📄 8 🌡 3,644 °C 🛠 编辑



一、前言

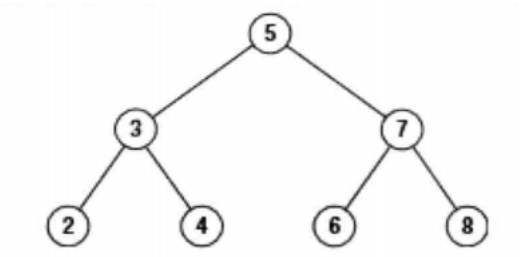
本系列文章为《剑指Offer》刷题笔记。

刷题平台：[牛客网](#)

书籍下载：[共享资源](#)

二、题目

给定一颗二叉搜索树，请找出其中的第k大的结点。例如，在下图中，按结点数值大小顺序第三个结点的值为4。



这棵树是二叉搜索树，首先想到的是二叉搜索树的一个特点：左子结点的值 < 根结点的值 < 右子结点的值。

1、思路

如上图所示，如果使用终须遍历，则得到的序列式为{2,3,4,5,6,7,8}。因此，只需要用中序遍历一棵二叉搜索树，就很容易找出它的第k大结点。

2、代码

C++:

C+

```
1  /*
2  struct TreeNode {
3      int val;
4      struct TreeNode *left;
5      struct TreeNode *right;
6      TreeNode(int x) :
7          val(x), left(NULL), right(NULL) {
8      }
9  };
10 */
11 class Solution {
12 public:
13     TreeNode* KthNode(TreeNode* pRoot, int k)
14     {
15         if(pRoot == NULL || k == 0){
16             return NULL;
17         }
18         return KthNodeCore(pRoot, k);
19     }
20 private:
21     TreeNode* KthNodeCore(TreeNode* pRoot, int &k){
22         TreeNode* target = NULL;
23         // 先遍历左结点
24         if(pRoot->left != NULL){
25             target = KthNodeCore(pRoot->left, k);
```

```
26     }
27     // 如果没有找到target, 则继续减小k, 如果k等于1, 说明到了第k大的数
28     if(target == NULL){
29         if(k == 1){
30             target = pRoot;
31         }
32         k--;
33     }
34     // 如果没有找到target, 继续找右结点
35     if(pRoot->right != NULL && target == NULL){
36         target = KthNodeCore(pRoot->right, k);
37     }
38     return target;
39 }
40 };
```



微信公众号

分享技术, 乐享生活: 微信公众号搜索

「JackCui-AI」关注一个在互联网摸爬滚
打的潜行者。

猛兽总是独行, 牛羊才成群结队。--- 鲁迅