

剑指Offer（五十七）：二叉树的下一个结点

© 2018年1月25日 10:32:57 发表评论 4,556 °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](#)

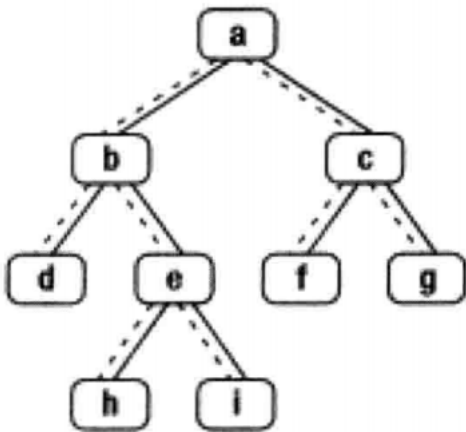
书籍下载：[共享资源](#)

二、题目

给定一个二叉树和其中的一个结点，请找出中序遍历顺序的下一个结点并且返回。注意，树中的结点不仅包含左右子结点，同时包含指向父结点的指针。

1、思路

这道题思路捋清楚，还是很简单的。



我们以上图为例进行讲解，上图二叉树的中序遍历是d,b,h,e,i,a,f,c,g。我们以这棵树为例来分析如何找出二叉树的下一个结点。

如果一个结点有右子树，那么它的下一个结点就是它的右子树的最左子结点。也就是说从右子结点出发一直沿着指向左子树结点的指针，我们就能找到它的下一个结点。例如，图中结点b的下一个结点是h，结点a的下一个结点是f。

接着我们分析一下结点没有右子树的情形。如果结点是它父结点的左子结点，那么它的下一个结点就是它的父结点。例如，途中结点d的下一个结点是b，f的下一个结点是c。

如果一个结点既没有右子树，并且它还是父结点的右子结点，这种情形就比较复杂。我们可以沿着指向父结点的指针一直向上遍历，直到找到一个不是父结点的左子结点的结点。如果这样的结点存在，那么这个结点的父结点就是我们要找的下一个结点。例如，为了找到结点g的下一个结点，我们沿着指向结点的指针向上遍历，先到达结点c。由于结点c是父结点a的右结点，我们继续向上遍历到达结点a。由于结点a是树的根结点。它没有父结点。因此结点g没有下一个结点。

2、代码

C++:

```
1  /*
2  struct TreeLinkNode {
3      int val;
4      struct TreeLinkNode *left;
5      struct TreeLinkNode *right;
6      struct TreeLinkNode *next;
7      TreeLinkNode(int x) :val(x), left(NULL), right(NULL), next(NULL) {
8
9      }
10 };
11 */
12 class Solution {
13 public:
14     TreeLinkNode* GetNext(TreeLinkNode* pNode)
15     {
16         if(pNode == NULL){
17             return NULL;
18         }
19         TreeLinkNode* pNext = NULL;
20         // 当前结点有右子树，那么它的下一个结点就是它的右子树中最左子结点
21         if(pNode->right != NULL){
22             TreeLinkNode* pRight = pNode->right;
23             while(pRight->left != NULL){
24                 pRight = pRight->left;
25             }
26             pNext = pRight;
27         }
28         // 当前结点无右子树，则需要找到一个它是它父结点的左子树结点的结点
29         else if(pNode->next != NULL){
30             // 当前结点
31             TreeLinkNode* pCur = pNode;
32             // 父节点
33             TreeLinkNode* pPar = pNode->next;
34             while(pPar != NULL && pCur == pPar->right){
35                 pCur = pPar;
36                 pPar = pCur->next;
37             }
38             pNext = pPar;
39         }
40         return pNext;
41     }
42 };
```



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。

面具戴太久，就会长到脸上，再想揭下来，除非伤筋动骨扒皮。--- 鲁迅