

hihoCoder 笔记

AbnerZheng

2016/08/25

摘要

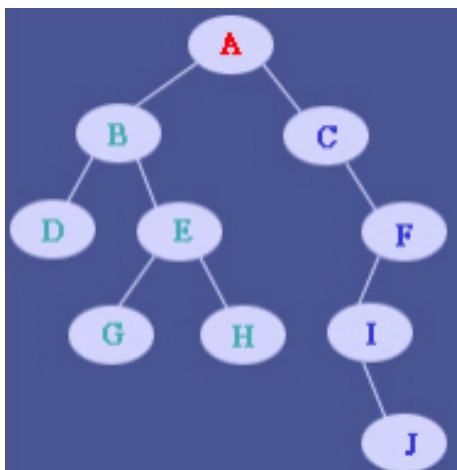
知道 hihoCoder 这个网站，大概是 15 年九月份校招的时候，当时 Line 和微软就是采用这个 OA 系统，体验很一般，当时并没有太多的印象。后面发现他每周都有一个 hiho 一下，已经做了一百多期了，就稍稍体验了把，感觉非常不错。前面的几十周，有教学提示，题目安排统一而有序，感觉非常适合我，然后就想着每天中午都刷刷题，做做笔记，希望自己的算法能力能有较大的提升。

1 第十周-后序遍历

1.1 题解

这题是已知前序和中序遍历，求后序遍历。在二叉树还原中，通过两个遍历得到另一个遍历，有一个结论是中序遍历必须存在。

对于一个二叉树，如下：



其前序遍历为: ABDEGHCFIG 其后序遍历为: DBGEHACIJF 由前中后序遍历的定义, 有

$$pre(T) = T + pre(T.leftChildren) + pre(T.rightChildren) \quad (1)$$

$$in(T) = in(T.leftChildren) + T + in(T.rightChildren) \quad (2)$$

$$post(T) = post(T.leftChildren) + post(T.rightChildren) + T \quad (3)$$

现在已知前序和中序遍历, 现求解

$$post(pre(T), in(T)) = post(pre(T.leftChildren), in(T.leftChildren)) + post(pre(T.rightChildren), in(T.rightChildren)) + T$$

所以关键的地方在于找到如何划分前序和中序遍历。因为子树大小相同, 那么对于子树的前中后序的长度是一致的, 根据这一条件, 可以很方面地得到这些划分。代码如下:

```

1 void post(string pre, string in){
2     int pos = 0;
3     int len_pre = pre.length();
4     int len_in = in.length();
5     if(len_in == 0){
6         return;
7     }
8
9     string::iterator iter;
10    for(iter = in.begin(); iter!=in.end(); ++iter){
11        if(*iter == pre[0]){
12            break;
13        } else{
14            ++pos;
15        }
16    }
17
18    /**
19     in(0,(pos-1)) -> in.leftChild -> len(in.leftChild) = pos
20     in(pos) -> T
21     in(pos+1, len(in)-1) -> in.rightChild -> len(in.
        rightChild) = len(in) - (pos+1)

```

```
22     */
23
24     int len_left = pos;
25     int len_right = len_pre - pos - 1;
26
27     post(pre.substr(1, len_left), in.substr(0, len_left));
28     post(pre.substr(pos+1, len_right), in.substr(pos+1, len_right
29         ));
29     cout<< pre[0];
30 }
```