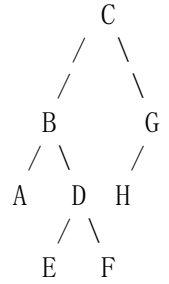
P1827 [USACO3.4] 美国血统 American Heritage

题目描述

农夫约翰非常认真地对待他的奶牛们的血统。然而他不是一个真正优秀的记帐员。他把他的奶牛 们的家谱作成二叉树，并且把二叉树以更线性的“树的中序遍历”和“树的前序遍历”的符号加以记录而 不是用图形的方法。

你的任务是在被给予奶牛家谱的“树中序遍历”和“树前序遍历”的符号后，创建奶牛家谱的“树的 后序遍历”的符号。每一头奶牛的姓名被译为一个唯一的字母。（你可能已经知道你可以在知道树的两 种遍历以后可以经常地重建这棵树。）显然，这里的树不会有多于 26 个的顶点。这是在样例输入和 样例输出中的树的图形表达方式：



树的中序遍历是按照左子树，根，右子树的顺序访问节点。

树的前序遍历是按照根，左子树，右子树的顺序访问节点。

树的后序遍历是按照左子树，右子树，根的顺序访问节点。

输入格式

第一行： 树的中序遍历

第二行： 同样的树的前序遍历

输出格式

单独的一行表示该树的后序遍历。

输入样例

```
ABEDFCHG
CBADEFGH
```

输出样例

```
AEFDBHGC
```

解析

模板题，根据前序遍历和中序遍历，生成后序遍历。

编码

```
#include <bits/stdc++.h>
using namespace std;
string pre_str; //前序
string in_str;  //中序
/**
 * 功能：对二叉树进行后序遍历.根据四个参数，
 * (1)确定子树的根，
 * (2)确定左右子树的大小范围，
 * (3)按后序进行输出递归
 * @param l1 前序遍历的起点
 * @param r1 前序遍历的终点
 * @param l2 中序遍历的起点
 * @param r2 中序遍历的终点
 */
void dfs(int l1, int r1, int l2, int r2) {
    if (l1 > r1 || l2 > r2) {
        return;//规定边界条件
    }
    //利用根左右的特性来在中序队列中查找根的位置
    int i = in_str.find(pre_str[l1]);
    int cnt = i - l2;//左子树的节点个数
    //前序：l1是根，左子树是从l1+1开始，到l1+cnt结束
    //中序：l2开始，到i-1结束
    dfs(l1 + 1, l1 + cnt, l2, i - 1);        //递归左子树
    dfs(l1 + cnt + 1, r1, i + 1, r2);           //递归右子树
    cout << pre_str[l1];    //输出根结点
}
```

```
int main() {
    //输入中序遍历，前序遍历字符串
    cin >> in_str >> pre_str;
    //right索引：因为是0开始，所以要-1
    int right = in_str.size() - 1;
    //递归构建还原二叉树
    //每个字符串有两个指针指向头和尾
    dfs(0, right, 0, right);
    return 0;
}
```