具体程序实现时，我们以字符串来表示所有可能的情况，例如样例所对应的字符串应为图4.31所示。

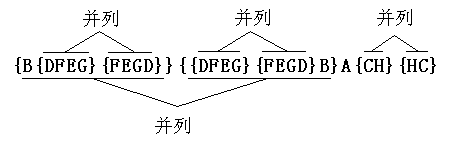


图4.31

并列的大括号表示可能的两种情况，每次碰到大括号时任选一个，最后就得到一个可能的中序遍历。那么如何任选或者说跳转呢？我们可以采用数组的方式来记录该字符串需要跳转的位置，数组cho[]各元素初始值为-1，对于任意并列的两对大括号，处理规则为：第一个大括号处记录第三个大括号的位置，第二、四个大括号处记录第四个大括号所在位置后的第一个位置，第三个大括号处值不变，表示不应跳转。标记好的数组见图4.32。

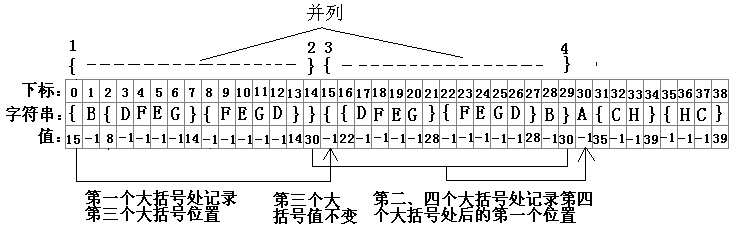


图4.32

处理完毕后，就可以用深度优先搜索算法（DFS）来输出所有结果了，操作规则如下：

1. 遇到字母就记录下来。

（２）遇到左大括号（且该位置数组元素值非-1），就分开两条路。

①直接向前走一步。

②跳到该位置数组元素所记录的位置处。

（３）遇到右括号就移动到该位置数组元素所记录的位置。

完整代码如下所示。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92 | //已知前序中序求后序，此程序仅供学有余力的读者选学  #include <bits/stdc++.h>  using namespace std;  #define ANSN 10000  string rx78; //用来记录上例中的那个包含括号和字母的字符串  int final,cho[ANSN+1]; //用来跳转的数组  void hwg(string a,string b)  {  int i,mark,ll,lr,len=a.length();  if(len>1) //如果不止有一个结点  {  if(a[1]==b[len-2]) //如果符合ABXXX，XXXBA的形式，就要分类处理  {  string p1(a,1,len-1),p2(b,0,len-1);//取出子树  //可能性1，该树是右子树，先访问根结点  rx78+='{';  mark=rx78.length()-1; //记录下左括号的位置  rx78+=a[0];  hwg(p1,p2); //打下标记后处理子树  rx78+='}';  cho[mark]=rx78.length(); //给左括号赋下跳转的位置  mark=rx78.length()-1; //记录下右括号的位置  //可能性2，该树是右子树，后访问根结点  rx78+='{';  hwg(p1,p2); //同上  rx78+=a[0];  rx78+='}';  cho[rx78.length()-1]=rx78.length();  cho[mark]=rx78.length(); //给两个右括号赋上跳转的位置  }  else //如果可以确定左右子树  {  for(i=0; i<len-2; i++)  if(a[1]==b[i])  break; //先找到左子树的根结点  ll=i+1;  lr=len-1-ll; //计算左右子树的长度  if(ll)  {  string part1(a,1,ll),part2(b,0,ll); //取出左子树  hwg(part1,part2); //递归求左子树  }  rx78+=a[0];//中序遍历，根结点在访问左子树后，访问右子树前访问  if(lr)  {  string part3(a,1+ll,lr),part4(b,ll,lr);//取出右子树  hwg(part3,part4); //递归求右子树  }  }  }  else //如果只有一个结点  rx78+=a[0];  }  void dfs(string have,int pos)  {  char ty=rx78[pos];  if(pos>=final) //如果深搜到尽头  cout<<have<<endl;  else if(ty=='{') //当碰到左括号时  {  if(cho[pos]==-1) //如果这个左括号不能跳转，就只能向前走一格  dfs(have,pos+1);  else //如果可以跳转  {  dfs(have,pos+1); //要么直接向前走一格  dfs(have,cho[pos]); //要么跳转  }  }  else if(ty=='}')  dfs(have,cho[pos]); //碰到右括号都跳转  else  {  have+=rx78[pos];  dfs(have,pos+1); //如果碰到字符就存入，并继续深搜  }  }  int main()  {  string front,back;  cin>>front>>back;  for(int i=0; i<=ANSN; i++)  cho[i]=-1;  hwg(front,back);  final=rx78.length();  dfs("",0);  return 0;  } |