

# 编辑距离问题

## 一. 问题分析:

对于给定的两个字符串,  $A[0:m-1]$  和  $B[0:n-1]$ , 定义  $P[i][j]$  表示字符串从  $A[0:i-1]$  转到  $B[0:j-1]$  的距离,  $p[0][0]=0$ , 故对于字符串的转换, 有如下几种情况。

- (1) 删除字符串  $A[i]$  需要 1 次操作
- (2) 插入字符串  $B[j]$  需要 1 次操作
- (3) 如果字符串  $A[i]=B[j]$ , 则不需要操作, 如果字符串  $A[i] \neq B[j]$  需要一次替换操作。

故其递归式子, 可以写成:

$$P[i][j] = \begin{cases} i; & \text{if } j = 0 \\ j; & \text{if } i = 0 \\ \min(p[i-1][j]+1, p[i][j-1]+1, p[i][j]), & \text{if } A[i] = B[j] \\ \min(p[i-1][j]+1, p[i][j-1]+1, p[i][j]+1), & \text{if } A[i] \neq B[j] \end{cases}$$

前两个是边界条件, 表示当一个为空的字符串时, 距离就是另外一个字符串的长度, 即通过增加或者删除得到。

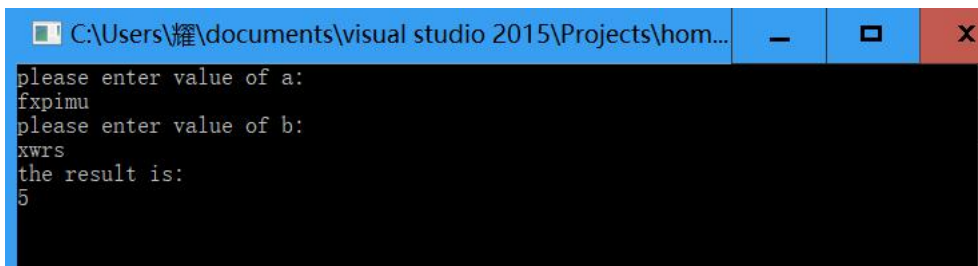
由反证法, 可以轻易证明, 其具有最优的子结构。

## 二. 算法分析:

由上面的递归子式子, 可以看出, 需要计算  $p$  的所有值,  $p$  的大小为  $mn$ , 故其算法的时间复杂度和空间复杂度均为:  $O(mn)$ 。

## 三. 程序设计:

通过控制台输入两个变量的字符串  $a, b$ ,  $d$  为存储的距离矩阵,  $m, n$  为  $a, b$  的长度。其运行的结果如图下所示:



以上为运行题目中测试样本的结果, 可以看出符合要求。

## 四. 代码:

```
// homework3.cpp : 定义控制台应用程序的入口点。
//

#include "stdafx.h"
using namespace std;
#include <iostream>
#include <vector>
#include <sstream>
int min(int a, int b, int c) //找到三个值中最小的一个
```

```

{
    int r=a;
    if (a > b)
    {
        r = b;
    }
    if (r > c) r = c;
    return r;
}

int main()
{
    //初始化各个变量
    string b = "";
    string a = "";
    int m = 0;
    int n =0;
    vector<vector <int>> d;//存储距离矩阵
    //输入:
    cout << "please enter value of a:" <<endl;
    getline(cin, a);
    cout << "please enter value of b:"<< endl;
    getline(cin, b);
    //初始化 d 的值, 以及边界条件
    m = a.length();
    n = b.length();
    for (int i = 0; i<=m; i++)
    {
        vector<int> temp;
        for (int j = 0; j <= n; j++)
        {
            temp.push_back(0);
        }
        d.push_back(temp);
    }
    for (int j = 0; j <= m; j++)
    {
        d[j][0] = j;
    }
    for (int i = 0; i <=n; i++)
    {
        d[0][i] = i;
    }
    //计算距离

```

```
for (int i = 1; i <= m; i++)
{
    for (int j = 1; j <= n; j++)
    {
        int temp = a[i - 1] == b[j - 1] ? 0 : 1;
        d[i][j] = min(d[i - 1][j] + 1, d[i][j - 1] + 1, temp + d[i-1][j-1]);
    }

}

//输出结果:
cout <<"the result is:"<<endl<< d[m][n];
int wait;
cin >> wait;
return 0;
}
```