51nod 1006 最长公共子序列Lcs 难点：在于如何输出一个最优解

1、LCS 求最大子序列, 一个办法当然是穷举法了, 但是穷举法太耗时间了O(2^m\*2^n), 基本没用.

2、聪明的程序员想到了，一个用矩阵来查找的算法，就是把两个队列用整形矩阵表示， 相同的为1， 不同的为0， 然后求最大对角线，优化是优化了很多， 不过求最大对角线也不省心。

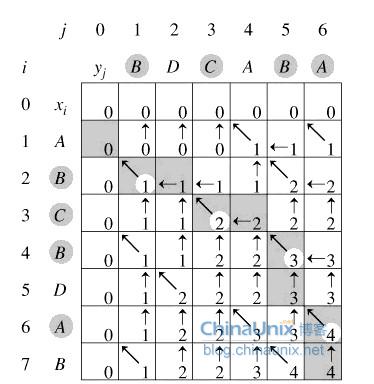
3、聪明的程序员再次优化了算法，就是相同的不是用1表示， 而是数字叠加，这样再找最大对角线的时候，就比较舒服了，时间复杂度也降到了 O（mn）+O(m+n) 如下图：

在画下图时总结成两条，这也是输出的算法，倒序枚举遵循以下规则：

1、字符相同，则指向左上，并加1

2、字符不同，则指向左边或者上边较大的那个

公共子串：B C B A 和 B D A B



#include<iostream>

#include<cstdio>

#include<cstring>

#include<algorithm>

using namespace std;

char a[1001];

char b[1001];

int f[1005][1005];

char s[2000];

int main(){

memset(f,0,sizeof(f));

scanf("%s%s",&a,&b);

int lena,lenb,i,j;

lena=strlen(a);

lenb=strlen(b);

/\*//下面这种写法会越界，问题出现在输出的时候用到i-1,j-1,当i,j=0时也满足，所以Runtime Error

for(i=0;i<lena;i++)

for(j=0;j<lenb;j++)

{

if(a[i]==b[j])

{

f[i][j]=f[i-1][j-1]+1;

}

else

{

f[i][j]=max(f[i-1][j],f[i][j-1]);

}

}

\*/

for(i=1;i<=lena;i++)

for(j=1;j<=lenb;j++)

{

if(a[i-1]==b[j-1])

{

f[i][j]=f[i-1][j-1]+1;

}

else

{

f[i][j]=max(f[i-1][j],f[i][j-1]);

}

}

int k=f[lena][lenb];

s[k--]='\0';

for(i=lena,j=lenb;i>0&&j>0;)//倒着查询

{

if(k<0)break;

if(a[i-1]==b[j-1])//相等

{

s[k--]=a[i-1];

i--;

j--;

}

else//不相等，f[i][j]=max(f[i-1][j],f[i][j-1]);则要么是比较后从上方复制来，要么是从左边复制过来，现在还原回去

{

if(f[i][j-1]>=f[i-1][j])

j--;

else

i--;

}

}

i=0;

while(s[i]!='\0')

{

cout<<s[i];

i++;

}

cout<<endl;

return 0;

}