Dynamic Programming

基础动态规划

第2课 子序列、子串、矩阵

By helang

模型1：公共子串

最长公共子串nkoj 1052

两个序列的最长公共子串，这个子串要求在序列中是连续的。例如："bab"和"caba"(可以看出来最长公共子串是"ba"或"ab")

再比如下列X和Y两个数字序列的最长公共子串是5 3 2

X 1,5,3,2,3

Y 2,3,5,3,2,5,3

X 1,5,3,2,3

Y 2,3,5,3,2,5,3

每条左上往右下的斜线
就代表了一种配对方式

XY	2	3	5	3	2	5	3
1	0	0	0	0	0	0	0
5	0	0	1	0	0	1	0
3	0	1	0	1	0	0	1
2	1	0	0	0	1	0	0
3	0	1	0	1	0	0	1

X 1,5,3,2,3

Y 2,3,5,3,2,5,1

填1时让它等于其**左上角元素**加1。

XY	2	3	5	3	2	5	3
1	0	0	0	0	0	0	0
5	0	0	1	0	0	1	0
3	0	1	0	2	0	0	2
2	1	0	0	0	3	0	0
3	0	2	0	1	0	0	1

矩阵中的最大元素就是
最长公共子串的长度。

$$m[\,i\,][\,j\,]=\begin{cases} m[\,i-1\,][\,j-1\,]+1 \\ (\,x[\,i\,]==y[\,j\,]\,) \\ 0 \\ (\,x[\,i\,]!=y[\,j\,]\,) \end{cases}$$

最长公共子串，参考代码：

```
cin>>lenX>>lenY;
for(i=1;i<=lenX;i++)cin>>x[i];
for(j=1;j<=lenY;j++)cin>>y[j];

for(i=1;i<=lenX;i++)
    for(j=1;j<=lenY;j++)
        if(x[i]==y[j])f[i][j]=f[i-1][j-1]+1;
        else f[i][j]=0;
```

最后，输出f[][]数组中最大的一个元素即可。

模型2：公共子序列

最长公共子序列nkoj 1051

问题描述

一个给定序列的子序列是在该序列中删去若干元素后得到的序列。例如，序列Z=<B,C,D,B>是序列X=<A,B,C,B,D,A,B>的子序列。

给定两个序列X和Y，当另一**序列Z既是X的子序列又是Y的子序列**时，**称Z是序列X和Y的公共子序列**。例如，若X=<A，B，C，B，D，A，B>和Y=<B，D，C，A，B，A>，则序列<B，C，A>是X和Y的一个公共子序列，序列<B，C，B，A>也是X和Y的一个公共子序列。而且，后者是X和Y的一个最长公共子序列。

最长公共子序列(LCS)问题：给定两个序列X=<x_1，x_2，…，x_m>和Y=<y_1，y_2，…，y_n>，要求找出X和Y的一个最长公共子序列。

输入

第一行，两个不超过500的整数x,y，表示两个序列的长度。
接下来有两行，第一行有空格间隔的x个整数，表示序列x
第二行有空格间隔的y个整数，表示序列y （x,y中的整数大小不超过200）

输出

输出文件第一行为一个非负整数，表示所求得的最长公共子序列长度

样例输入：

7 6
15 10 7 10 3 15 10
10 3 7 15 12 10

样例输出：

4

最长公共子序列的性质

设序列$X=\langle x_1, x_2, \cdots, x_m\rangle$和$Y=\langle y_1, y_2, \cdots, y_n\rangle$的最长公共子序列为$A=\langle a_1, a_2, \cdots, a_k\rangle$，则有如下性质：

(1)如果$x_m = y_n$，且$a_k = x_m = y_n$，则

$A'=\langle a_1, a_2, \cdots, a_{k-1}\rangle$是

$X'=\langle x_1, x_2, \cdots, x_{m-1}\rangle$和$Y'=\langle y_1, y_2, \cdots, y_{n-1}\rangle$最长公共子序列。

(2)如果$x_m \neq y_n$，且$a_k \neq x_m$ 则

$A=\langle a_1, a_2, \cdots, a_k\rangle$是

$X'=\langle x_1, x_2, \cdots, x_{m-1}\rangle$和$Y'=\langle y_1, y_2, \cdots, y_n\rangle$最长公共子序列。

(3)如果$x_m \neq y_n$，且$a_k \neq y_n$ 则

$A=\langle a_1, a_2, \cdots, a_k\rangle$是

$X'=\langle x_1, x_2, \cdots, x_m\rangle$和$Y'=\langle y_1, y_2, \cdots, y_{n-1}\rangle$最长公共子序列。

性质1

- 性质1表明，如果两个字符串最后一位字符相等，则它们的最长公共子串的最后一位就是这个字符，并且公共子串的第k-1个前缀是两个字符串前缀的最长公共子串。

- 证明：用反证法。若$a_k!=x_m$,则因为$x_m=y_n$，把x_m添加在A的后面形成$\langle a_1,a_2,\cdots,a_k,x_m\rangle$，是X和Y的长度为K+1的最长公共子序列。这与A是X和Y的最长公共子序列矛盾。因此，必有$a_k=x_m=y_n$。由此可知，A[1...k-1]是X[1...m-1]和Y[1...n-1]的一个长度为k-1的公共子序列。

性质2

- 性质2表明，如果两个字符串最后一位不等，且它们的最长公共子串A的最后一位不等于X的最后一位，则A是X的前缀与Y的最长公共子串。换句话说，Xm不可能是公共子序列的最后一位。

- 证明：用反证法。由于a_k!=x_m，A是x_{m-1}和Y的一个公共子序列。若x_{m-1}和y有一个长度大于K的公共子序列W，则W也必然是X和Y的一个长度大于K的公共子序列，这与A是X和Y的一个最长公共子序列矛盾。

性质3同理可证！

问题分析：

要找出$X=\langle x_1, x_2, \cdots, x_m\rangle$和$Y=\langle y_1, y_2, \cdots, y_n\rangle$的最长公共子序列，可按以下方式进行讨论：

阶段: 以每个元素作为阶段，从前往后依次讨论序列中的每一个元素。

决策: 1. 当$x_m==y_n$时，只需找出$\langle x_1,\ldots,x_{m-1}\rangle$和$\langle y_1,\ldots,y_{n-1}\rangle$的最长公共子序列，然后在其尾部加上$x_m(==y_n)$这一项，即可得X和Y的一个最长公共子序列；

2. 当$x_m \neq y_n$时，必须解两个子问题：

a.找出$\langle x_1,\ldots,x_{m-1}\rangle$和$\langle y_1,\ldots,y_n\rangle$的最长公共子序列；

b.找出$\langle x_1,\ldots,x_m\rangle$和$\langle y_1,\ldots,y_{n-1}\rangle$的最长公共子序列；

这两个公共子序列中较长者即为X和Y的最长公共子序列；

状态: f[i][j]表示X序列的前i项与Y序列的前j项能构成的最长公共子序列的长度

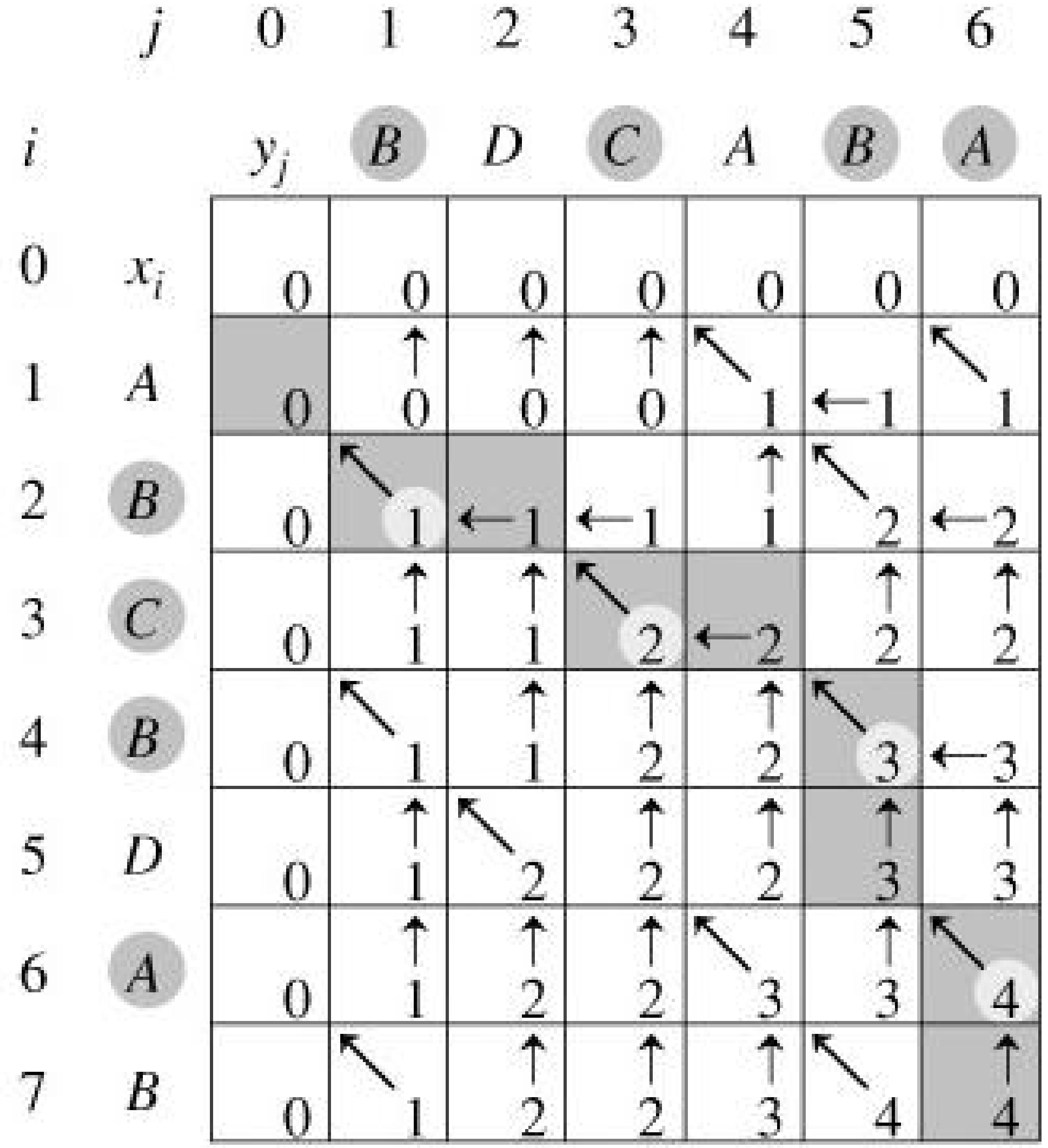
也就是$\langle x_1,x_2,\ldots,x_i\rangle$和$\langle y_1,y_2,\ldots,y_j\rangle$的最长公共子序列的长度值

方程:

$$f[i][j]=\begin{cases} f[i-1][j-1]+1 & (x[i]==y[j]) \\ \max\{ f[i-1][j] , f[i][j-1] \} & (x[i]!=y[j]) \\ 0 & (i==0 \text{ 或 } j==0) \end{cases}$$

边界条件： 1<=i<=n, 1<=j<=m

方程: $$f[i][j]=\begin{cases} f[i-1][j-1]+1 & (x[i]==y[j]) \\ \max\{\, f[i-1][j]\ ,\ f[i][j-1]\,\} & (x[i]!=y[j]) \\ 0 & (i==0 \text{ 或 } j==0) \end{cases}$$



最长公共子序列，参考代码：

```
cin>>lenX>>lenY;                    //输出X、Y序列的长度
for(i=1;i<=lenX;i++)cin>>x[i];
for(j=1;j<=lenY;j++)cin>>y[j];
for(i=0;i……;i++){ f[i][0]=0; f[0][i]=0; } //初始化DP数组

for(i=1;i<=lenX;i++)
    for(j=1;j<=lenY;j++)
        if(x[i]==y[j])f[i][j]=f[i-1][j-1]+1;
        else if(f[i-1][j]>f[i][j-1])f[i][j]=f[i-1][j];
             else f[i][j]=f[i][j-1];

cout<<f[lenX][lenY];
```

延伸思考1：三个序列的最长公共子序列 nkoj 3636

给定三个长度不超过200的整数序列X,Y,Z，求他们的最长公共子序列。

例如：下列三个序列的最长公共子序列长度为4，如红色部分所示。

X=<1,3,5,7,9,11,13,15>

Y=<0,1,3,6,9,12,15,18>

Z=<1,2,3,4,9,10,15,20>

三个序列的最长公共子序列，问题分析：

状态: f[i][j][k]表示X序列的前i项、Y序列的前j项与Z序列的前k项能构成的最长公共子序列的长度

也就是 $<x_1,x_2,\ldots,x_i>$ 和 $<y_1,y_2,\ldots,y_j>$ 和 $<z_1,z_2,\ldots,z_k>$ 的最长公共子序列的长度值

方程:

f[i][j][k]=

f[i-1][j-1][k-1]+1　　若X[i]==Y[j]==Z[k]

max{ f[i][j][k-1] , f[i-1][j-1][k] }　　若X[i]==Y[j]!=Z[k]

max{ f[i-1][j][k] , f[i][j-1][k-1] }　　若X[i]!=Y[j]==Z[k]

max{ f[i][j-1][k] , f[i-1][j][k-1] }　　若X[i]==Z[k]!=Y[j]

max{ f[i-1][j][k] , f[i][j-1][k], f[i][j][k-1], f[i-1][j-1][k], f[i-1][j][k-1], f[i][j-1][k-1] }　　若X[i]!=Z[k]!=Y[j]

延伸思考2：求最长**上升公共**子序列

怎样求两个序列的最长上升公共子序列，NKOJ1035

算法$O(N^3)$和$O(N^2)$

课后习题：

1051 最长公共子序列

1052 最长公共子串

1036 回文词

模型3：矩阵

典型例题：建别墅 NKOJ1182

何老板买了一块面积为n*m的土地，他想在这块土地上建造一座别墅。按照中国传统四平八稳的思想，他希望这个别墅是**正方形**的。

但是，这块土地并非十全十美，上面有很多地方是不平坦的，以至于根本不能在上面盖一砖一瓦。

他希望找到一块最大的平坦的正方形土地来盖别墅。应该选哪一块土地呢？现在，你来告诉他吧。

Input

第一行为两个整数n,m（1<=n,m<=1000），接下来n行，每行m个数字，用空格隔开。0表示该块土地不平坦，1表示该块土地平坦。

Output

一个整数，最大正方形的边长。

Sample Input

```
4 4
0 1 1 1
1 1 1 0
0 1 1 0
1 1 0 1
```

Sample Output

```
2
```

动规分析：

阶段： 从上往下，从左往右依次讨论矩阵中每个数字。

状态： f[i][j]表示把点(i,j)当做一个正方形对角线(左上往右下)的右下角端点，能够得到的最大正方形的边长。

决策： 见右侧动画。

方程：

$$f[i][j]=\begin{cases} 0 & a[i][j]==0 \\ \min\{ f[i-1][j-1],f[i][j-1],f[i-1][j] \}+1 & a[i][j]==1 \end{cases}$$

边界条件： 1<=i<=n , 1<=j<=m

4 4

0	1	1	1
1	1	1	0
1	1	1	0
1	1	1	1

0	1	1	1
1	1	2	0
1	2	2	0
1	2	3	1

思考：建别墅 2.0

何老板买了一块面积为n*m的土地，他想在这块土地上建造一座别墅。按照中国传统四平八稳的思想，他希望这个别墅是**矩形**的。
但是，这块土地并非十全十美，上面有很多地方是不平坦的，以至于根本不能在上面盖一砖一瓦。

他希望找到一块最大的平坦的矩形土地来盖别墅。应该选哪一块土地呢？现在，你来告诉他吧。

Input

第一行为两个整数n,m（1<=n,m<=1000），接下来n行，每行m个数字，用空格隔开。0表示该块土地不平坦，1表示该块土地平坦。

Output

一个整数，最大矩形的面积。

Sample Input

```
4 4
0 1 1 1
1 1 1 0
0 1 1 0
1 1 0 1
```

单调队列的"滑动窗口"模型

Sample Output

```
6
```

模型4：最大子矩阵

例题：开垦农田 NKO2553

何老板买了一块土地，他想在这块土地上开垦出一块矩形的农田。这块土地可看做由n*n个小方块土地构成，何老板告诉你每小块土地的肥沃值，他希望开垦出一片肥沃值总和最大的**矩形**农田。

问：这个最大肥沃值总和是多少？

Input

第一行为整数n（1<=n<=100）

接下来是一个由整数构成的n*n的矩阵，数字间以空格间隔，每个数字的范围在-10000到10000之间。

Output

一个整数，表示所求的结果。

Sample Input

4

0	-2	-7	0
9	2	-6	2
-4	1	-4	1
-1	8	0	-2

Sample Output

15

复习：最大连续子序列(子串)

我们可以把问题抽象为下列数学模型：

给定K个整数的序列{ A_1, A_2, ..., A_n }，求其中一段任意长度的连续序列，要求该序列的数字之和最大。

方法二，动态规划：

事先把这n个整数依次存到a数组中

按**阶段**讨论问题：从右到左依次讨论每个数字

确定要求解的**状态**：f[i]表示以第i个数字作为开头的连续序列的最大和

做出最佳**决策**：　如果 f[i+1]<=0，那么f[i]=a[i];
　　　　　　　　　如果 f[i+1]>0，　那么f[i]=f[i+1]+a[i];

最后得出**方程**：
$$f[i]=\begin{cases} f[i+1]+a[i]; & (\ f[i+1]>0\ ) \\ a[i]; & (\ f[i+1]<=0\ ) \end{cases}$$

边界条件：1<=i<=n

最大子矩形

状态：f[i][j][k]表示恰好以第i行到第j行构成的，且以第k列作为结束的最大子矩形

原题即转化为：枚举i,j,k找出最大的f[i][j][k]。

把第i行到第j行每一列的数相加，看作是一个数，则求f[i][j][k] 即变成了求最大子串。

把每个蓝色方框框选区域的数字相加看做一个数字。那么我们就得到了k个数字。

求f[i][j][k]就变成了在这k个数字中求以第k个数最为结尾的最大子串。

	-50	2	-3	4	1	-2	3
	-9	7	2	-4	6	-3	5
i	-8	6	9	-2	-1	-10	7
	7	-5	3	8	7	2	-3
	-2	-4	7	6	5	1	2
j	-5	0	2	3	-4	5	-6
	5	-19	-8	4	-3	-9	-1

Sum[3][6][k]	-8	-3	19	15	7	-2	0
f[3][6][k]	-8	-3	19	34	41	39	39

最大子矩形 参考代码：

```
for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
    {
        scanf("%d",&t);
        Sum[i][j]=t+Sum[i-1][j];
    }//前缀和Sum[i][j]表示第1行第j列到第i行第j列数字之和。
```

时间复杂度O(n^3)

```
for(i=1;i<=n;i++)
    for(j=i;j<=n;j++)
        for(k=1;k<=n;k++)
        {
            t=Sum[j][k]-Sum[i-1][k];//将第i到第j行的第k列求和，转换为一个数字
            if(Dp[i][j][k-1]<=0)Dp[i][j][k]=t; //讨论最大连续和
            else Dp[i][j][k]=t+Dp[i][j][k-1];
            if(Dp[i][j][k]>Max)Max=Dp[i][j][k]; //更新答案
        }
```

本课动规模型：

1.最长公共子序列
2.最长公共子串
3.矩阵
4.最大子矩阵

课后习题：1051,1052,1182,2553
1006，1835