**计算机科学与技术卓越班 杨小川 20174179**

**（1）15.4-1 求{1，0，0，1，0，1，0，1}和{0，1，0，1，1，0，1，1，0}的一个LCS。**

LCS = {1，0 ，0，1，1，0}

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Yj | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | Xi | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 4 | 1 | 0 | 1 | 2 | 2 | 3 | 3 | 3 | 4 | 4 | 4 |
| 5 | 0 | 0 | 1 | 2 | 3 | 3 | 3 | 4 | 4 | 4 | 5 |
| 6 | 1 | 0 | 1 | 2 | 3 | 4 | 4 | 4 | 5 | 5 | 5 |
| 7 | 0 | 0 | 1 | 2 | 3 | 4 | 4 | 5 | 5 | 5 | 6 |
| 8 | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 5 | 6 | 6 | 6 |

**（2）15.4-3 设计LCS-LENGTH的带备忘的版本，运行时间为O（mn）。**

伪代码：

LCS-LENGTH(X, Y, b, c,i,j)

if c[i, j] != -1

return c[i, j]

If x[i] == y[j]

c[i, j] = LCS-LENGTH(X,Y,b,c,i-1,j-1)+1

b[i, j] = “\'

else

up = LCS-LENGTH(X,Y,b,c,i-1,j)

left = LCS-LENGTH(X,Y,b,c,i,j-1)

if up >= left

c[i, j] = up

b[i, j] = “ |

else

c[i, j] = left

b[i, j] = “<-”

return c[i, j]

C++描述：

**#define** P -999

**for**(**int** i=0;i<x.len+1;i++)

**for**(**int** j=0;j<y.len+1;j++)

c[i][j]=P;

**for**(**int** i=0;i<x.len+1;i++)

c[i][0]=0;

**for**(**int** j=0;j<y.len+1;j++)

c[0][j]=0;

**int** LCS-LENGTH(**char** x[], **char** y[], **int** b[][N], **int** c[][N], **int** i,**int** j)

{

**if** (c[i][j] != P)

**return** c[i][j];

**if** (x[i] == y[j]) {

c[i][j] = LCS-LENGTH(x,y,b,c,i-1,j-1) + 1;

b[i][j] = ‘\’;

}

**else** {

**int** up=LCS-LENGTH(x,y,b,c,i-1,j);

**int** left=LCS-LENGTH(x,y,b,c,i,j-1);

**if** ( up >= left ) {

c[i][j] = up;

b[i][j] = ‘|’;

}

**else** {

c[i][j] = left;

b[i][j] = ‘<-’;

}

}

**return** c[i][j];

}

**（3）M排N列的木桩，从第一排开始跳到最后一排。每次跳到下一排的同一列、前一列或后一列木桩。**

**设R[i,j]为跳到i排j列木桩的奖金。求从第一排跳到最后一排获得的奖金总量的最大值，给出递推方程。**

设最大值为SUM(i, j)，R[i, j]为该位置上的金额，则：



则MAX{ SUM(m, 0..n) }就为最大值

1. **上述问题，如果允许在同一排向左或向右跳，且横跳总次数不超过H，给出求奖金总量的最大值的递推方程**

构造三维的立方 m\*n\*h，假设起点为最底层,即h=0 i=1，则对任意一点(m, n, h)，能从五个点跳达，分别为(i, j-1, k-1),(i, j+1, k-1),(i-1, j, k),(i-1, j-1, k),(i-1, j+1, k)

设SUM(i, j, k)为当前位置[i, j]的最大值



则MAX{ SUM(m, 0..n, h) }就为最大值

（5）如果每个木桩只有第一次跳上去的时候有奖金，如何求最大值

由于每个木桩上的奖励只能获得一次，故对于每一行若横跳，则单方向跳才能使收益最大。

故使每次横跳时跳k格，并加上跳过的k格中所有的数之和

