**《算法分析与设计》第3次作业**

姓名：你的名字 学号：你的学号 成绩：

算法分析题

题目1：假设矩阵*A*、*B*、*C*、*D*、*E*的维数序列为<5, 10, 3, 12, 5, 50>，用动态规划算法找出其最佳连乘顺序。为方便起见，将矩阵连乘积A*i*A*i*+1…A*j*简记为A[*i*:*j*]。

(a) 若将计算A[*i*:*j*]所需的最少乘法次数记为*m*[*i*,*j*]，将矩阵A*i*的维数记为 ,请给出*m*[*i*,*j*]的递推方程（即递归定义）；

(b) 若将对应于*m*[*i*,*j*]的最佳断开位置记为s[*i,j*]。根据上述实例，写出*m*[*i*,*j*]表和s[*i,j*]表的填充过程，并结合这两张表给出该实例的最佳连乘顺序。

答：

1. *m*(*i*, *j*) = 如果 *j* > *i* + 1

*m*(*i*, *j*) = 0 如果 *j* = *i* +1

(b)

****

****

最佳顺序为 (((AB)(CD))E)。

题目2：假设我们要将一根大型长钢管锯成若干段。我们在要锯的地方打上标记，一共是*n*个标记。这些标记距离钢管左端的距离，从左到右，分别是*a*1，*a*2， …，*an* 厘米。这个钢管的总长是*l* 厘米，*l* > *an* (参见下图)。当我们将钢管锯为两截时，需要的代价与当时被锯钢管的长度成正比，比例是每一厘米*p*元。



(a) 请用动态规划的方法设计一个算法来找出最优的顺序来完成这*n*处的切割使总的代价最小。分析你的算法的复杂度。 (提示：用 *a*0 = 0 和*an*+1 = *l* 表示钢管左端和右端位置。用[*ai*, *aj*]表示从标记 *ai* 到 标记*aj* 这段钢管。用*C*(*i*, *j*) 表示完成对[*ai*, *aj*]这段钢管的切割任务所需的最小代价，也就是完成所有*ak* (*i* < *k* < *j*) 处的切割所需代价。 找出 *C*(*i*, *j*) 的归纳公式。)

(b) 以下面数据为例，用你在(a)中的算法找出最优的切割顺序和总代价。请显示你的计算过程。

*a*1 = 2, *a*2 = 5, *a*3 = 9, *a*4= 14, *l* = 15, *p* = 1.

答：

**(a)** 定义 *C*(*i*, *j*) = 切割钢管[*ai*, *aj*]所需最小代价。我们有以下归纳公式：

*C*(*i*, *j*) = { *C*(*i*, *k*) + *C*(*k*, *j*) + *p*(*aj* – *ai*) } 如果 *j* > *i* + 1

*C*(*i*, *j*) = 0 如果 *j* = *i* +1

在下面的伪码中，我们用表*S*[*i*, *j*] 记录切割钢管[*ai*, *aj*]的位置。

**Optimal-Cutting** (*A*, *n*)

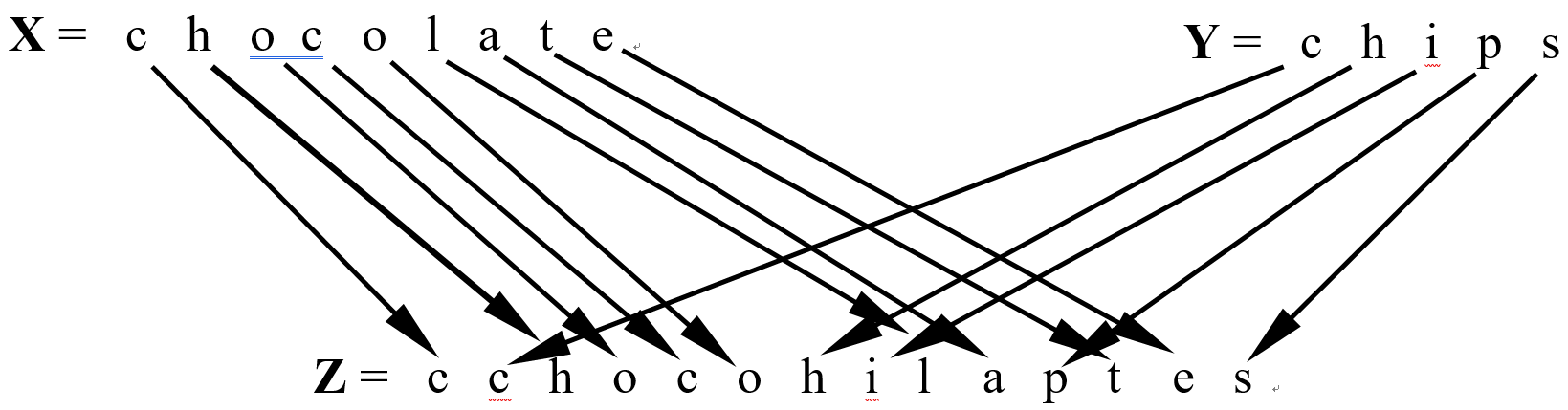
1. **for** *i* ← 0 **to** *n*
2. *C*[*i*, *i*+1] ← 0 //初始解
3. **endfor**
4. **for** *l* ← 1 **to** *n*
5. **for** *i* ← 0 **to** *n* – *l*
6. *j* ← *i* + 1 + *l*
7. *C*[*i*, *j*] ← ∞
8. **for** *k* ← *i* +1 **to** *j* - 1
9. *q* ← *C*[*i*, *k*] + *C*[*k*, *j*] + *p*(*aj* – *ai*)
10. **if** *q* < *C*[*i*, *j*]
11. **then** *C*[*i*, *j*] ← *q*
12. *S*[*i*, *j*] ← *k*
13. **endif**
14. **endfor**
15. **endfor**
16. **endfor**
17. **return** *C* and *S*
18. **End**

上面算法的复杂度显然是O(*n*3)。

**(b)** *a*1 = 2, *a*2 = 5, *a*3 = 9, *a*4= 14, *l* = 15, *p* = 1。表*C*、表*S*和表示顺序的二叉树构造如下，总的代价是35元。



题目3：假设我们有三个字母或数字的序列，*X*[1..*m*] = *x*1*x*2 ... *xm*， *Y*[1..*n*] = *y*1*y*2...*yn*，和*Z*[1..*m*+*n*] = *z*1*z*2...*zm*+*n*。如果序列*Z* 是由*X*和*Y*中的元素按顺序交汇而成，那么*Z* 被称为*X*和*Y*的一个洗牌。例如，下面图中的序列*Z* = *cchocohilaptes* 是 *X* = *chocolate* 和*Y* = *chips* 的一个洗牌。



(a) 用动态规划的方法设计一个算法来确定序列*Z*是否是*X*和*Y*的一个洗牌。(提示：用*M*[*i*, *j*] = 1 表示*Z*[1..*i*+*j*] 是 *X*[1..*i*] 和 *Y*[1..*j*]的一个洗牌。然后找出归纳公式。)

(b) 用你的算法确定以下三序列中，*Z*是否是*X*和*Y*的一个洗牌。

*X* = FEAST, *Y* = LOVE, *Z* = FLOEVASET。

答：

**解一：**

有个简单的方法是先求*X*和*Z*的最长公共子序列*W*。如果*W* ≠ *X*，则*Z*不可能是*X*和*Y*的一个洗牌。如果*W* = *X*，检查把*X*从*Z*中删去后的序列是否等于*Y*。如果是，则*Z*是*X*和*Y*的一个洗牌，否则不是。为了练习起见，建议学生不借用已有算法而用动态规划直接解(参考解二)。

**解二：**

**(a)** 我们将建一个表*M*，其中*M*[*i*, *j*] = 1 表示*Z*[1..*i*+*j*] 是 *X*[1..*i*] 和 *Y*[1..*j*]的一个洗牌。我们有以下归纳公式。

初始解是

*M*[0, 0] = 1

当*i* >0 或*j* > 0时公式为

如果(*j* > 0, *M*[*i*, *j*-1] = 1 和 *Z*[*i*+*j*] = *Y*[*j*] ) 或者 (*i* > 0, *M*[*i*-1, *j*] = 1 和 *Z*[*i*+*j*] = *X*[*i*])

*M*[*i*, *j*] = 1

否则

*M*[*i*, *j*] = 0

如果 *M*[*m*, *n*] = 1，那么*Z*是*X*和*Y*的一个洗牌。伪码如下。

**Shuffle** (*X*[1..*m*], *Y*[1..*n*], *Z*[1..*m*+*n*]，*M*，*D*)

1. *M*[0, 0] ← 1
2. **for** *i* ← 0 **to** *m*
3. **for** *j* ← 0 **to** *n*
4. **if** (*j* > 0 **and** *M*[*i*, *j*-1] = 1 **and** *Z*[*i*+*j*] = *Y*[*j*] )
5. **then** *M*[*i*, *j*] ← 1
6. *D*[*i*, *j*] ← “ ← ”
7. **else if** (*i* > 0 **and** *M*[*i*-1, *j*] = 1 **and** *Z*[*i*+*j*] = *X*[*i*])
8. **then** *M*[*i*, *j*] ← 1
9. *D*[*i*, *j*] ← “ ↑ ”
10. **else** *M*[*i*, *j*] ← 0
11. *D*[*i*, *j*] ← *nil*
12. **endif**
13. **endif**
14. **endfor**
15. **endfor**
16. **if** *M*[*m*, *n*] = 1
17. **then return** *yes* **and** *D*
18. **else return** *no*
19. **endif**
20. **End**

算法显然有复杂度O(*mn*)。算法中表*D*可省略，但如果要知道*Z*是如何洗牌得到的，那么可从*D*得到。

**(b)** *X* = FEAST, *Y* = LOVE, *Z* = FLOEVASET。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表M | *Y*[*j*] |  | L | O | V | E |
| *X*[*i*] |  | 0 | 1 | 2 | 3 | 4 |
|  | 0 | 1 | 0 | 0 | 0 | 0 |
| F | 1 | 1  **↑** | 1  **←** | 1  **←** | 0 | 0 |
| E | 2 | 0 | 0 | 1  **↑** | 1  **←** | 0 |
| A | 3 | 0 | 0 | 0 | 1  **↑** | 0 |
| S | 4 | 0 | 0 | 0 | 1  **↑** | 1  **←** |
| T | 5 | 0 | 0 | 0 | 0 | 1  **↑** |

由上表知*Z*是*X*和*Y*的一个洗牌。