Analysis of Algorithms: Solutions 9

```
    X                   X
    X   X       X       X
X   X   X   X   X       X       X
X   X   X   X   X       X   X   X
X   X   X   X   X       X   X   X
-----------------------------------
2   3   4   5   6   7   8   9   10
```

Problem 1
Suppose that you need to compute a product of five matrices, and their dimensions are $p[0..5] = \langle 3, 8, 5, 2, 20, 4 \rangle$. Give the optimal parenthesization of this product and the corresponding value of $m[0, 5]$.

5	312				
4	248	304			
3	128	400	200		
2	120	80	200	160	
1	0	0	0	0	0
	1	2	3	4	5

The optimal parenthesization is $(A_1 \cdot (A_2 \cdot A_3)) \cdot (A_4 \cdot A_5)$, and $m[0, 5]$ is 312.

Problem 2
Determine a longest common subsequence of $\langle a, b, a, b, a, a \rangle$ and $\langle b, a, a, b, b, a, a, b \rangle$. Using Figure 16.3 in the book as a model, draw the table constructed by LCS-Length.

8	b	0	1	2	3	4	4	5
7	a	0	1	2	3	3	4	5
6	a	0	1	2	3	3	4	4
5	b	0	1	2	2	3	3	3
4	b	0	1	2	2	3	3	3
3	a	0	1	1	2	2	3	3
2	a	0	1	1	2	2	2	2
1	b	0	0	1	1	1	1	1
0		0	0	0	0	0	0	0
i			a	b	a	b	a	a
	j	0	1	2	3	4	5	6

The longest common subsequence is $\langle b, a, b, a, a \rangle$.

Problem 3

Give pseudocode for a modified version of LCS-LENGTH that uses only $O(n)$ memory.

LINEAR-LCS(X, Y, m, n)
for $j \leftarrow 0$ **to** n
    do $c[j] \leftarrow 0$
for $i \leftarrow 1$ **to** m
    do $old \leftarrow 0$
        for $j \leftarrow 1$ **to** n
            do $temp \leftarrow c[j]$
                if $X[i] = Y[j]$
                    then $c[j] \leftarrow old + 1$
                    else $c[j] \leftarrow \max(temp, c[j-1])$
                $old \leftarrow temp$
return $c[n]$

Problem 4

Consider two recurrences:

$$A_0 = 0$$
$$A_n = A_{n-1} + n \quad (\text{where } n \geq 1)$$

$$B_0 = 0$$
$$B_n = B_{n-1} + A_n \quad (\text{where } n \geq 1)$$

Give a constant-time algorithm FIND-B(n) for computing B_n.

We can prove the following expressions by induction:

$$A_n = \frac{n \cdot (n+1)}{2}$$
$$B_n = \frac{n \cdot (n+1) \cdot (n+2)}{6}$$

The constant-time algorithm uses the resulting expression for computing B_n:

FIND-B(n)
return $\frac{n(n+1)(n+2)}{6}$