Solution keys: HW12

Problem 1:

Consider $A_1A_2A_3$, where $A_1 : 2\times 1$, $A_2 : 1\times 3$, and $A_3 : 3\times 3$. The greedy algorithms (a) and (c) will compute in the order of $(A_1A_2)A_3$ with the cost to be 24, however the optimal order is $A_1(A_2A_3)$ with the minimum cost of 15.

Consider $A_1A_2A_3$, where $A_1 : 2\times 1$, $A_2 : 1\times 3$, and $A_3 : 3\times 1$. The greedy algorithm (b) will compute in the order of $(A_1A_2)A_3$ with the cost to be 12, however the optimal order is $A_1(A_2A_3)$ with the minimum cost of 5.

Problem 2: What is the optimal way to compute $A_1A_2A_3A_4A_5A_6$, where the dimensions of the matrices are: $A_1 : 10\times 20$, $A_2 : 20\times 1$, $A_3 : 1\times 40$, $A_4 : 40\times 5$, $A_5 : 5\times 30$, $A_6 : 30\times 15$? To answer the question, you must first use dynamic programming to build the 6×6 table.

Solution: We first recognize that $p_0 = 10$, $p_1 = 20$, $p_2 = 1$, $p_3 = 40$, $p_4 = 5$, $p_5 = 30$, and $p_6 = 15$. Then we use the formula

$$m[i,j] = \min_{i\le k\le j-1}\{m[i,k]+m[k+1,j]+p_{i-1}p_kp_j\}, i<j$$

to construct the table. Note the number inside the parenthesis is the k that gives the minimum for the corresponding $m[i,j]$

$i\backslash j$	1	2	3	4	5	6
1	0	200 (1)	600 (2)	450 (2)	850 (2)	1150 (2)
2		0	800 (2)	300 (2)	950 (2)	1100 (2)
3			0	200 (3)	350 (4)	800 (5)
4				0	6000 (4)	5250 (4)
5					0	2250 (5)
6						0

To construct the ordering, we first go to entry $m[1,6]$. Its k-value of 2 suggests the following partial ordering of

$$(A_1\times A_2)\times(A_3\times A_4\times A_5\times A_6).$$

Then we go to $m[3,6]$. Its k-value of 5 suggests the following partial ordering of

$$(A_1\times A_2)\times((A_3\times A_4\times A_5)\times A_6).$$

Finally we go to $m[3,5]$. Its k-value of 4 completes the optimal ordering of

$$(A_1\times A_2)\times(((A_3\times A_4)\times A_5)\times A_6).$$