ADw6&7
Name : **Peter Garamvoelgyi** Student ID: **2018280070**

CLRS 15.2-1. Find an optimal parenthesization of a matrix-chain product whose sequence of dimensions is $\langle 5, 10, 3, 12, 5, 50, 6 \rangle$. (pg. 378)

$$A_{5\times 10}A_{10\times 3}A_{3\times 12}A_{12\times 5}A_{5\times 50}A_{50\times 6}$$

$$m[i,j] = \begin{cases} 0 & i = j \\ \min\limits_{i\le k<j}\{m[i,k] + m[k+1,j] + p_{i-1}p_kp_j\} & i < j \end{cases}$$

i \ j	1	2	3	4	5	6
1	**0**					
2	-	**0**				
3	-	-	**0**			
4	-	-	-	**0**		
5	-	-	-	-	**0**	
6	-	-	-	-	-	**0**

$m[1,2] = m[1,1] + m[2,2] + p_0p_1p_2 = 0 + 0 + 5\cdot 10\cdot 3 = 150$
$m[2,3] = m[2,2] + m[3,3] + p_1p_2p_3 = 0 + 0 + 10\cdot 3\cdot 12 = 360$
$m[3,4] = m[3,3] + m[4,4] + p_2p_3p_4 = 0 + 0 + 3\cdot 12\cdot 5 = 180$
$m[4,5] = m[4,4] + m[5,5] + p_3p_4p_5 = 0 + 0 + 12\cdot 5\cdot 50 = 3000$
$m[5,6] = m[5,5] + m[6,6] + p_4p_5p_6 = 0 + 0 + 5\cdot 50\cdot 6 = 1500$

i \ j	1	2	3	4	5	6
1	0	**150** **k=1**				
2	-	0	**360** **k=2**			
3	-	-	0	**180** **k=3**		
4	-	-	-	0	**3000** **k=4**	
5	-	-	-	-	0	**1500** **k=5**
6	-	-	-	-	-	0

$$m[1,3] = \min\begin{Bmatrix} m[1,1] + m[2,3] + p_0p_1p_3 \\ m[1,2] + m[3,3] + p_0p_2p_3 \end{Bmatrix} = \min\begin{Bmatrix} 0 + 360 + 5 \cdot 10 \cdot 12 \\ 150 + 0 + 5 \cdot 3 \cdot 12 \end{Bmatrix}$$
$$= \min\begin{Bmatrix} 960 \\ 330 \end{Bmatrix} = 330$$
$$m[2,4] = \min\begin{Bmatrix} m[2,2] + m[3,4] + p_1p_2p_4 \\ m[2,3] + m[4,4] + p_1p_3p_4 \end{Bmatrix} = \min\begin{Bmatrix} 0 + 180 + 10 \cdot 3 \cdot 5 \\ 360 + 0 + 10 \cdot 12 \cdot 5 \end{Bmatrix}$$
$$= \min\begin{Bmatrix} 330 \\ 960 \end{Bmatrix} = 330$$
$$m[3,5] = \min\begin{Bmatrix} m[3,3] + m[4,5] + p_2p_3p_5 \\ m[3,4] + m[5,5] + p_2p_4p_5 \end{Bmatrix} = \min\begin{Bmatrix} 0 + 3000 + 3 \cdot 12 \cdot 50 \\ 180 + 0 + 3 \cdot 5 \cdot 50 \end{Bmatrix}$$
$$= \min\begin{Bmatrix} 4800 \\ 930 \end{Bmatrix} = 930$$
$$m[4,6] = \min\begin{Bmatrix} m[4,4] + m[5,6] + p_3p_4p_6 \\ m[4,5] + m[6,6] + p_3p_5p_6 \end{Bmatrix} = \min\begin{Bmatrix} 0 + 1500 + 12 \cdot 5 \cdot 6 \\ 3000 + 0 + 12 \cdot 50 \cdot 6 \end{Bmatrix}$$
$$= \min\begin{Bmatrix} 1860 \\ 6600 \end{Bmatrix} = 1860$$

j

i

	1	2	3	4	5	6
1	0	150 k=1	**330** **k=2**			
2	-	0	360 k=2	**330** **k=2**		
3	-	-	0	180 k=3	**930** **k=4**	
4	-	-	-	0	3000 k=4	**1860** **k=4**
5	-	-	-	-	0	1500 k=5
6	-	-	-	-	-	0

$$m[1,4] = \min\begin{Bmatrix} m[1,1] + m[2,4] + p_0p_1p_4 \\ m[1,2] + m[3,4] + p_0p_2p_4 \\ m[1,3] + m[4,4] + p_0p_3p_4 \end{Bmatrix} = \min\begin{Bmatrix} 0 + 330 + 5 \cdot 10 \cdot 5 \\ 150 + 180 + 5 \cdot 3 \cdot 5 \\ 330 + 0 + 5 \cdot 12 \cdot 5 \end{Bmatrix}$$
$$= \min\begin{Bmatrix} 580 \\ 405 \\ 630 \end{Bmatrix} = 405$$

$$m[2,5] = \min\begin{Bmatrix} m[2,2] + m[3,5] + p_1p_2p_5 \\ m[2,3] + m[4,5] + p_1p_3p_5 \\ m[2,4] + m[5,5] + p_1p_4p_5 \end{Bmatrix} = \min\begin{Bmatrix} 0 + 930 + 10 \cdot 3 \cdot 50 \\ 360 + 3000 + 10 \cdot 12 \cdot 50 \\ 330 + 0 + 10 \cdot 5 \cdot 50 \end{Bmatrix}$$
$$= \min\begin{Bmatrix} 2430 \\ 9360 \\ 2830 \end{Bmatrix} = 2430$$

$$m[3,6] = \min\begin{Bmatrix} m[3,3] + m[4,6] + p_2p_3p_6 \\ m[3,4] + m[5,6] + p_2p_4p_6 \\ m[3,5] + m[6,6] + p_2p_5p_6 \end{Bmatrix} = \min\begin{Bmatrix} 0 + 1860 + 3\cdot 12\cdot 6 \\ 180 + 1500 + 3\cdot 5\cdot 6 \\ 930 + 0 + 3\cdot 50\cdot 6 \end{Bmatrix}$$

$$= \min\begin{Bmatrix} 2076 \\ 1770 \\ 1830 \end{Bmatrix} = 1770$$

j

i

	1	2	3	4	5	6
1	0	150 k=1	330 k=2	**405** **k=2**		
2	-	0	360 k=2	330 k=2	**2430** **k=2**	
3	-	-	0	180 k=3	930 k=4	**1770** **k=4**
4	-	-	-	0	3000 k=4	1860 k=4
5	-	-	-	-	0	1500 k=5
6	-	-	-	-	-	0

$$m[1,5] = \min\begin{Bmatrix} m[1,1] + m[2,5] + p_0p_1p_5 \\ m[1,2] + m[3,5] + p_0p_2p_5 \\ m[1,3] + m[4,5] + p_0p_3p_5 \\ m[1,4] + m[5,5] + p_0p_4p_5 \end{Bmatrix} = \min\begin{Bmatrix} 0 + 2430 + 5\cdot 10\cdot 50 \\ 150 + 930 + 5\cdot 3\cdot 50 \\ 330 + 3000 + 5\cdot 12\cdot 50 \\ 405 + 0 + 5\cdot 5\cdot 50 \end{Bmatrix}$$

$$= \min\begin{Bmatrix} 4930 \\ 1830 \\ 6330 \\ 1655 \end{Bmatrix} = 1655$$

$$m[2,6] = \min\begin{Bmatrix} m[2,2] + m[3,6] + p_1p_2p_6 \\ m[2,3] + m[4,6] + p_1p_3p_6 \\ m[2,4] + m[5,6] + p_1p_4p_6 \\ m[2,5] + m[6,6] + p_1p_5p_6 \end{Bmatrix} = \min\begin{Bmatrix} 0 + 1770 + 10\cdot 3\cdot 6 \\ 360 + 1860 + 10\cdot 12\cdot 6 \\ 330 + 1500 + 10\cdot 5\cdot 6 \\ 2430 + 0 + 10\cdot 50\cdot 6 \end{Bmatrix}$$

$$= \min\begin{Bmatrix} 1950 \\ 2940 \\ 2130 \\ 5430 \end{Bmatrix} = 1950$$

i \ j	1	2	3	4	5	6
1	0	150 k=1	330 k=2	405 k=2	**1655 k=4**	
2	-	0	360 k=2	330 k=2	2430 k=2	**1950 k=2**
3	-	-	0	180 k=3	930 k=4	1770 k=4
4	-	-	-	0	3000 k=4	1860 k=4
5	-	-	-	-	0	1500 k=5
6	-	-	-	-	-	0

$$m[1,6] = \min\begin{Bmatrix} m[1,1] + m[2,6] + p_0p_1p_6 \\ m[1,2] + m[3,6] + p_0p_2p_6 \\ m[1,3] + m[4,6] + p_0p_3p_6 \\ m[1,4] + m[5,6] + p_0p_4p_6 \\ m[1,5] + m[6,6] + p_0p_5p_6 \end{Bmatrix} = \min\begin{Bmatrix} 0 + 1950 + 5 \cdot 10 \cdot 6 \\ 150 + 1770 + 5 \cdot 3 \cdot 6 \\ 330 + 1860 + 5 \cdot 12 \cdot 6 \\ 405 + 1500 + 5 \cdot 5 \cdot 6 \\ 1655 + 0 + 5 \cdot 50 \cdot 6 \end{Bmatrix}$$

$$= \min\begin{Bmatrix} 2250 \\ 2010 \\ 2550 \\ 2055 \\ 3155 \end{Bmatrix} = 2010$$

i \ j	1	2	3	4	5	6
1	0	150 k=1	330 k=2	405 k=2	1655 k=4	**2010 k=2**
2	-	0	360 k=2	330 k=2	2430 k=2	1950 k=2
3	-	-	0	180 k=3	930 k=4	1770 k=4
4	-	-	-	0	3000 k=4	1860 k=4
5	-	-	-	-	0	1500 k=5
6	-	-	-	-	-	0

The best parenthesization is

$$(A_{5\times 10}A_{10\times 3})\big((A_{3\times 12}A_{12\times 5})(A_{5\times 50}A_{50\times 6})\big)$$

… with cost 2010.

CLRS 15.3-3. Consider a variant of the matrix-chain multiplication problem in which the goal is to parenthesize the sequence of matrices so as to maximize, rather than minimize, the number of scalar multiplications. Does this problem exhibit optimal substructure? (pg. 389)

> "*A problem exhibits optimal substructure if an optimal solution to the problem contains within it optimal solutions to subproblems.*" (CLRS pg. 379)

Let us prove that this problem has the optimal substructure property by using the "cut-and-paste" technique.

Let us suppose that, for a given input, the algorithm gives us the optimal solution

$$S \sim \{m[1,k] + m[k+1,n] + p_0 p_k p_n\}$$

… consisting of the two sub-solutions

$$S_1 \sim m[1,k]$$
$$S_2 \sim m[k+1,n]$$

The cost (that is to be maximized in this case) is given by the formula

$$\text{cost}(S) = \text{cost}(S_1) + \text{cost}(S_2) + p_0 p_k p_n$$

Let us assume the sub-solutions are not optimal, i.e.

$$\forall\, S_1' : \ \text{cost}(S_1') > \text{cost}(S_1)$$

By replacing S_1 with S_1', the cost of the final solution S'' becomes

$$\text{cost}(S'') = \text{cost}(S) - \text{cost}(S_1) + \text{cost}(S_1')$$

As $\text{cost}(S_1') - \text{cost}(S_1) > 0$, we can see that

$$\text{cost}(S'') > \text{cost}(S)$$

… which is a contradiction as S was said to be an optimal solution. Thus, the subproblems must be optimal.

CLRS 25.2-4. As it appears above, the Floyd-Warshall algorithm requires $\Theta(n^3)$ space, since we compute $\mathrm{d}_{ij}^{(k)}$ for $i, j, k = 1, 2, \ldots, n$. Show that the following procedure, which simply drops all the superscripts, is correct, and thus only $\Theta(n^2)$ space is required. (pg. 699).

The $d_{ij}^{(k)}$ values depend on

$$d_{ij}^{(k-1)} \quad d_{ik}^{(k-1)} \quad d_{kj}^{(k-1)}$$

… i.e. only values from the previous $(k-1)$ matrix. Thus, it is trivial that keeping only two matrices (last and current) at any time is sufficient. The space requirement is $\Theta(n^2)$.

The fact that having one matrix is sufficient is less trivial, as the values

$$d_{ik} \quad d_{kj}$$

… might change before our update of d_{ij}, making it incorrect.

However, let us notice that values in row and column k in fact do not change:

$$\begin{array}{rlll} \text{row k:} & i = k & \rightarrow & d_{kj} = \min\left(d_{kj}, d_{kk} + d_{kj}\right) = \min\left(d_{kj}, 0 + d_{kj}\right) = d_{kj} \\ \text{column k:} & j = k & \rightarrow & d_{ik} = \min(d_{ik}, d_{ik} + d_{kk}) = \min(d_{ik}, d_{ik} + 0) = d_{ik} \end{array}$$

This means that in every iteration of k, recent updates of D will not affect subsequent updates (d_{ij}).

Thus, keeping only a single matrix D, the algorithm remains correct. This results in a non-asymptotic reduction in space requirements. The asymptotic bound remains $\Theta(n^2)$.