$m[i,j] = m[i,k] + m[k+1,j] + p_{i-1} \cdot p_k \cdot p_j$

p = first dimension of each matrix

$\overset{i=1}{A_1}$	A_2	A_3	A_4	A_5	$\overset{j=6}{A_6}$
5×10	10×3	3×12	12×5	5×50	50×6

p_0 5×3 p_1 5×12 p_2 5×5 p_3 $5 \cdot 50$ p_4 $5 \cdot 6$ p_5 p_6

$5 \cdot 10 \cdot 3 + (15 \cdot 12) + (5 \cdot 5 \cdot 12) + (5 \cdot 5 \cdot 50) + (5 \cdot 50 \cdot 6) =$

$m[1,6] = m[\overset{0}{1,1}] + m[\overset{1950}{2,6}] + \overset{p_0}{5} \cdot \overset{p_1}{10} \cdot 6 = 2250$

$= m[\overset{150}{1,2}] + m[\overset{1770}{3,6}] + 5 \cdot 3 \cdot 6 = 2010$

$= m[\overset{330}{1,3}] + m[\overset{1800}{4,6}] + 5 \cdot 12 \cdot 6 = 2550$

$= m[\overset{480}{1,4}] + m[\overset{1500}{5,6}] + 5 \cdot 5 \cdot 6 = 2130$

$= m[\overset{1750}{1,5}] + m[\overset{0}{6,6}] + 5 \cdot 50 \cdot 6 = 3250$

$\to m[2,6] = m[2,2] + m[\overset{1770}{3,6}] + 10 \cdot 3 \cdot 6 = 1950$

$\to m[3,6] = m[3,3] + m[\overset{1860}{4,6}] + 3 \cdot 12 \cdot 6 = 2076$

$m[4,6] = m[4,4] + m[\overset{1500}{5,6}] + 12 \cdot 5 \cdot 6 = 1860$

$= m[\overset{3000}{4,5}] + m[6,6] + 12 \cdot 50 \cdot 6 = 6600$

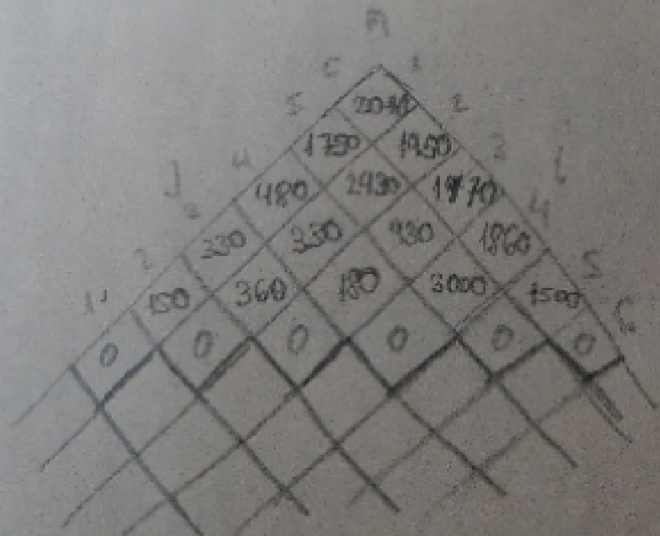
$= m[\overset{180}{3,4}] + m[\overset{1500}{5,6}] + 3 \cdot 5 \cdot 6 = 1770$

$= m[3,5] + m[6,6] + 3 \cdot 50 \cdot 6 = 1830$

$= m[\overset{360}{2,3}] + m[\overset{1860}{4,6}] + 10 \cdot 12 \cdot 6 = 2940$

$= m[\overset{330}{2,4}] + m[\overset{1500}{5,6}] + 10 \cdot 5 \cdot 6 = 2130$

$= m[\overset{2430}{2,5}] + m[\overset{0}{6,6}] + 10 \cdot 50 \cdot 6 = 5430$



Final Solution:

$(A_1 A_2)((A_3 A_4)(A_5 A_6))$

$(A_1 \cdot A_2) \cdot [(A_3 \cdot A_4) \cdot (A_5 \cdot A_6)]$

$m[1,2] = m[\overset{0}{1,1}] + m[\overset{0}{2,2}] + \overset{p_0}{5} \cdot \overset{p_1}{10} \cdot \overset{p_2}{3} = 150$

$m[1,3] = m[\overset{0}{1,1}] + m[\overset{360}{2,3}] + 5 \cdot 10 \cdot \overset{p_3}{12} = 960$

$\hookrightarrow m[2,3] = m[2,2] + m[3,3] + 10 \cdot 3 \cdot 12 = 360$

$\hookrightarrow \quad = m[\overset{150}{1,2}] + m[3,3] + 5 \cdot 3 \cdot 12 = 380$

$m[1,4] = m[1,1] + m[\overset{330}{2,4}] + \overset{p_0}{5} \cdot \overset{p_1}{10} \cdot \overset{p_4}{5} = 580$

$\hookrightarrow m[2,4] = m[2,2] + m[\overset{180}{3,4}] + 10 \cdot 3 \cdot 5 = 330$

$= m[2,3] + m[4,4] + \overset{p_1}{10} \cdot \overset{p_3}{12} \cdot 5 = 2160$

$= m[\overset{150}{1,2}] + m[\overset{180}{3,4}] + 10 \cdot 3 \cdot 5 = 480$

$= m[\overset{960}{1,3}] + m[4,4] + 3 \cdot 12 \cdot 5 = 1500$

$m[1,5] = m[1,1] + m[\overset{1420}{2,5}] + 5 \cdot 10 \cdot 50 = 4920$

$\hookrightarrow m[2,5] = m[2,2] + m[\overset{930}{3,5}] + 10 \cdot 3 \cdot 50 = 1420$

$\hookrightarrow m[3,5] = m[3,3] + m[\overset{1800}{4,5}] + \overset{p_2}{3} \cdot \overset{p_3}{12} \cdot \overset{p_5}{50} = 4800$

$= m[\overset{180}{3,4}] + [5,5] + 3 \cdot 5 \cdot 50 = 930$

$\hookrightarrow \quad = m[2,3] + m[4,5] + 10 \cdot 12 \cdot 50 = 9360$

$= m[2,4] + m[5,5] + 10 \cdot 5 \cdot 50 = 2830$

$= m[\overset{150}{1,2}] + m[\overset{930}{3,5}] + 5 \cdot 3 \cdot 50 = 1830$

$= m[\overset{330}{1,3}] + m[\overset{1800}{4,5}] + 5 \cdot 12 \cdot 50 = 6330$

$= m[\overset{480}{1,4}] + m[5,5] + 5 \cdot 5 \cdot 50 = 1750$

Pg 699 25.2-4

To prove that updating in place gives the correct answer, we need to show that the term $\min(d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1})$ is the same as $\min(d_{ij}, d_{ik} + d_{kj})$

- $d_{ij} = d_{ij}^{k-1}$ since k is in the outermost loop.
- $d_{ik} = d_{ik}^{k-1}$ or $d_{ik} = d_{ik}^{k}$

In the former case, there is no problem. In the latter case, we can note that $d_{ik}^{k} = d_{ik}^{k-1}$ because going through vertex k through all intermediate vertices in set $\{1 \dots k\}$ is the same as having all ~~t~~ intermediate vertices in set $\{1 \dots k-1\}$. The same logic applies to d_{kj}

Albert Millan Truyols (2019280366)

p.389 15.3-3

Optimal substructure: optimal solutions to a problem incorporate optimal solutions to related subproblems, which may be solved independently.

Yes, it exhibits optimal substructure property. Maximizing is no different from minimizing. The optimal solution can be computed by dividing the array A_1, ..., A_n between A_k and A_(k+1) and commiting further divisions that yield the most expensive operation on each side. That is, finding the optimal solution at each subproblem, each being solved independently.

p.396 15.4-4

a) To compute any given row from the 'c' table, only the current and previous rows are needed, hence only these two should be stored in memory at a single time. The following modification can therefore be made to compute the 'c' table:

- Store in two in two arrays the length of min(m, n), one for the previous row and another for the current-row, holding the rows of c.
- Initialize the previous row array with zeros and compute the current-row from left to right.
- Copy current-row into previous-row array and compute the new current-row
- Repeat previous step until there are no more rows to compute.

b) To compute a given c[i, j], only entries c[i-1,j], c[i-1,j-1] and c[i,j-1] are required. Therefore, it can free up entry-byentry those from the previous row which will never need again, reducing the space requirement to min(m,n).