15.2 Matrix-chain multiplication problem

Let $A = (a_{ij})$ be an $\ell \times m$ matrix and $B = (b_{ij})$ an $m \times n$ matrix.

The product $X = AB$ then is the $\ell \times n$ matrix $X = (x_{ij})$, where

$$x_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj}.$$

Thus, multiplying A and B takes ℓmn scalar multiplications.

Now, suppose we have a third matrix C_{ij} which is an $n \times p$ matrix.
To compute the product ABC, we can

- (1) Find AB then multiply by C — $\ell mn + \ell np$ multiplications, or
- (2) Find BC then multiply by A — $mnp + \ell mp$ multiplications.

Which is best?

15.2 Matrix-chain multiplication (continued)

More generally, we have a chain $\langle A_1, A_2, \ldots, A_n \rangle$ of n matrices, where A_i has dimension $p_{i-1} \times p_i$ for $i = 1, 2, \ldots, n$. We want to compute $A_1 A_2 \ldots A_n$ using the least total number of scalar multiplications.

Each order of multiplication corresponds to a parenthesisation. For example, we can compute $A_1 A_2 A_3 A_4$ in 5 ways:

1. $(A_1 A_2)(A_3 A_4)$
2. $((A_1 A_2) A_3) A_4$
3. $(A_1 (A_2 A_3)) A_4$
4. $A_1 ((A_2 A_3) A_4)$
5. $A_1 (A_2 (A_3 A_4))$ (Question: how many ways for n matrices?)

15.2 Matrix-chain multiplication (continued)

Denote by the number of alternative parenthesizations of a sequence of n matrix multiplication by $P(n)$. Then

$$P(n) = \begin{cases} 1 & \text{if } n = 1, \\ \sum_{k=1}^{n-1} P(k) \cdot P(n-k) & \text{if } n \geq 2. \end{cases}$$

$P(n) = \Omega(4^n / n^{3/2})$, which is usually referred to as *the Catalan number*.

15.2 Matrix-chain multiplication (continued)

1. Characterize the structure of the optimal solution

Let $A_{i..j}$ denote the product of $(j-i+1)$ matrices $A_iA_{i+1}\cdots A_{j-1}A_j$ with $i \le j$.
The dimensions of A_i is $p_{i-1} \times p_i$.

As a special case where $A_{i..i} = A_i$, we aim to compute $A_{1..n}$.

Consider where the last multiplication is performed in computing $A_{1..n}$. For some k $(1 \le k \le n-1)$, we have computed $A_{1..k}$ and $A_{k+1..n}$,
we then multiply them to get $A_{1..n}$, using an extra $p_0p_kp_n$ scalar multiplications.

If this is the best way (least total scalar multiplications), then we must have obtained $A_{1..k}$ and $A_{k+1..n}$ using the least total scalar multiplications. However, we don't know the best k, so we have to consider all the possibilities $1 \le k \le n-1$ and take the best.

2. Formulate the recurrence for the optimal solution

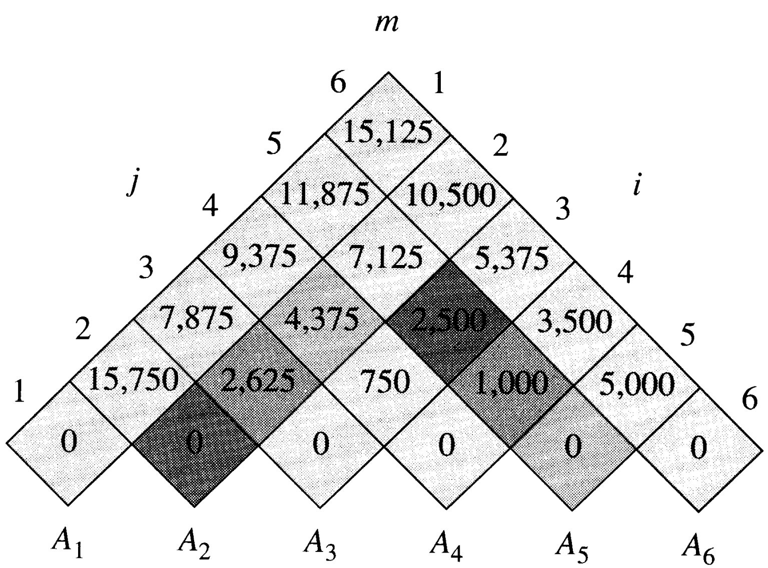
Let $m[i, j]$ be the least number of scalar multiplications needed to compute $A_{i..j}$.

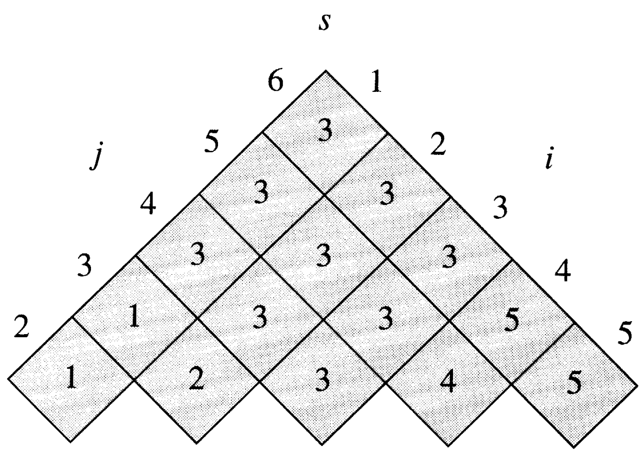
$$m[i,i] = 0, \qquad (1 \leq i \leq n)$$

$$m[i,j] = \min_{i \leq k < j} \{ m[i,k] + m[k+1,j] + p_{i-1}p_k p_j \}, \qquad (1 \leq i < j \leq n)$$

Example (p376): $A_1A_2A_3A_4A_5A_6$. Dimensions:

A_1: 30×35, A_2: 35×15, A_3: 15×5, A_4: 5×10, A_5: 10×20, A_6: 20×25





Example (p376): $A_1 A_2 A_3 A_4 A_5 A_6$.

A_1: $p_0 \times p_1$, A_2: $p_1 \times p_2$, A_3: $p_2 \times p_3$, A_4: $p_3 \times p_4$, A_5: $p_4 \times p_5$, A_6: $p_5 \times p_6$

A_1: 30×35, A_2: 35×15, A_3: 15×5, A_4: 5×10, A_5: 10×20, A_6: 20×25

We here calculate $A_3 A_4 A_5$ as follows,
where $i = 3$, $j = 5$, $p_{i-1} = p_2 = 15$, $p_i = p_3 = 5$, $p_4 = 10$, and $p_j = p_5 = 20$.

$m[3,4] = 15 \times 5 \times 10 = 750$; $s[3,4] = 3$;
$m[4,5] = 5 \times 10 \times 20 = 1,000$; $s[4,5] = 4$;

$$\begin{aligned}
m[3,5] &= \min_{i=3 \le k < j=5} \{m[i,k] + m[k+1,j] + p_{i-1} p_k p_j\} \\
&= \min\{m[3,3] + m[4,5] + p_2 \times p_3 \times p_5,\ \ m[3,4] + m[5,5] + p_2 \times p_4 \times p_5\} \\
&= \min\{m[3,3] + m[4,5] + 15 \times 5 \times 20,\ \ m[3,4] + m[5,5] + 15 \times 10 \times 20\} \\
&= \min\{0 + 1,000 + 1,500,\ 750 + 0 + 3,000\} \\
&= \min\{2,500,\ 3,750\} \\
&= 2,500 \qquad (1)
\end{aligned}$$

$s[3,5] = 3$, as $k = 3$ the value of $m[3,5]$ is the minimum one.

Example (p376): $A_1A_2A_3A_4A_5A_6$. Dimensions:

A_1: 30×35, A_2: 35×15, A_3: 15×5, A_4: 5×10, A_5: 10×20, A_6: 20×25

A_1: $p_0 \times p_1$, A_2: $p_1 \times p_2$, A_3: $p_2 \times p_3$, A_4: $p_3 \times p_4$, A_5: $p_4 \times p_5$, A_6: $p_5 \times p_6$

$$
\begin{aligned}
m[2,5] &= \min_{i=2 \leq k < j=5} \{m[i,k] + m[k+1,j] + p_{i-1}p_kp_j\} \\
&= \min\{m[2,2] + m[3,5] + p_1 \times p_2 \times p_5, m[2,3] + m[4,5] + p_1 \times p_3 \times p_5, \\
&\quad m[2,4] + m[5,5] + p_1 \times p_4 \times p_5\} \\
&= \min\{0 + 2500 + 35 \cdot 15 \cdot 20, 2625 + 1000 + 35 \cdot 5 \cdot 20, 4375 + 0 + 35 \cdot 10 \cdot 20\} \\
&= \min\{13,000, 7125, 11,375\} \\
&= 7125 \qquad (2)
\end{aligned}
$$

$s[2,5] = 3$, as $k = 3$ the value of $m[2,5]$ is the minimum one.

15.2 Matrix-chain multiplication (continued)

3. Algorithm for the recurrence

Matrix_Chain_Order$(p[])$

$n \leftarrow length[p] - 1;$
for $i \leftarrow 1$ to n do
    $m[i,i] \leftarrow 0$
for $l \leftarrow 2$ to n do
    for $i \leftarrow 1$ to $n - l + 1$ do
        $j \leftarrow i + l - 1;$
        $m[i,j] \leftarrow \infty;$
        for $k \leftarrow i$ to $j - 1$ do
            $q \leftarrow m[i,k] + m[k+1,j] + p_{i-1} * p_k * p_j;$
            if $q < m[i,j]$ then $m[i,j] \leftarrow q$; $s[i,j] \leftarrow k;$
return m and s.

What is the running time of the proposed algorithm?

15.2 Matrix-chain multiplication (continued)

4. Construct the optimal solution

Having a matrix $s[1..n, 1..n]$ that stores the indices of the computation of $m[i, j]$s, we now construct the optimal solution for matrix multiplication.

The following procedure is used to construct an optimal solution to a partial solution of the product of matrices $A_i \dots A_j$ with $j > i$.

Matrix_Chain_Multiply(A, s, i, j)

if $j > i$
    then $X \leftarrow$ Matrix_Chain_Multiply$(A, s, i, s[i, j])$
        $Y \leftarrow$ Matrix_Chain_Multiply$(A, s, s[i, j] + 1, j)$
        return $X \times Y$
    else return A_i

Thus, the optimal solution of the problem is obtained by calling **Matrix_Chain_Multiply**$(A, s, 1, n)$.