طراحی و تحلیل الگوریتم ها

دکتر امیر لکی زاده
استادیار گروه مهندسی کامپیوتر دانشگاه قم

2. Matrix-chain Multiplication

The ***matrix-chain multiplication problem*** can be stated as follows: given a chain $\langle A_1, A_2, \ldots, A_n \rangle$ of n matrices, where for $i = 1, 2, \ldots, n$, matrix A_i has dimension $p_{i-1} \times p_i$, fully parenthesize the product $A_1 A_2 \cdots A_n$ in a way that minimizes the number of scalar multiplications.

2. Matrix-chain Multiplication

Counting the number of parenthesizations

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

Catalan numbers, which grows as $\Omega(4^n / n^{3/2})$

Brute-Force :

2. Matrix-chain Multiplication

$A = \langle A_1, A_2, \ldots, A_n \rangle$

$P = \langle p_0, p_1, p_2, \ldots, p_n \rangle$

$A_i = \langle p_{i-1}, p_i \rangle$

$A_{i..j}$

$m[i,j]$

$A = \langle A_1, A_2, A_3 \rangle$

$P = \langle p_0, p_1, p_2, p_3 \rangle = \langle 10, 100, 5, 50 \rangle$

$A_{i..j}$

2. Matrix-chain Multiplication

$A_{i..j}$ ابعاد

$A_{i..j}$ $A_{i..k}$ $A_{k+1..j}$

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

$$m[i, i] = 0 \text{ for } i = 1, 2, \ldots, n$$

2. Matrix-chain Multiplication

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

$s[i,j]$

$m[1,n]$

Brute-Force:

2. Matrix-chain Multiplication

نحوه محاسبه ماتریس m:

سطری، ستونی و یا ...

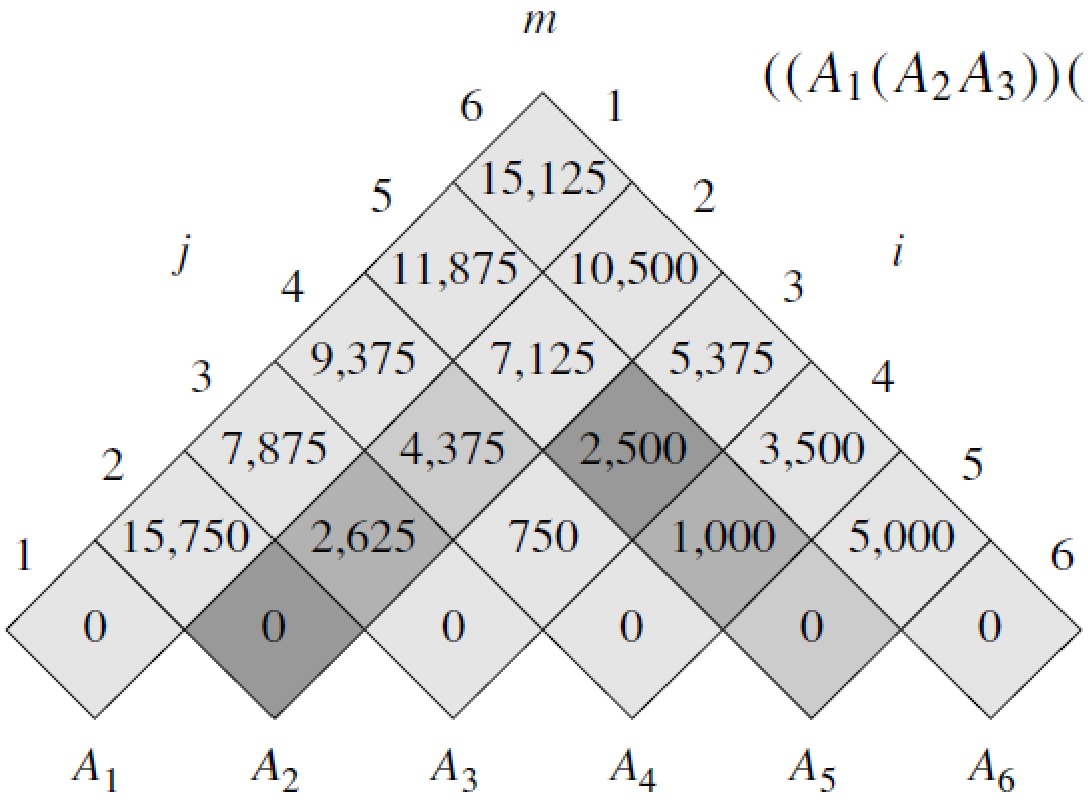
2. Matrix-chain Multiplication

```
MATRIX-CHAIN-ORDER(p)
n ← length[p] − 1
for i ← 1 to n
     do m[i, i] ← 0
for l ← 2 to n                  ▷ l is the chain length.
     do for i ← 1 to n − l + 1
             do j ← i + l − 1
                m[i, j] ← ∞
                for k ← i to j − 1
                     do q ← m[i, k] + m[k + 1, j] + p_{i−1} p_k p_j
                        if q < m[i, j]
                           then m[i, j] ← q
                                s[i, j] ← k
return m and s
```

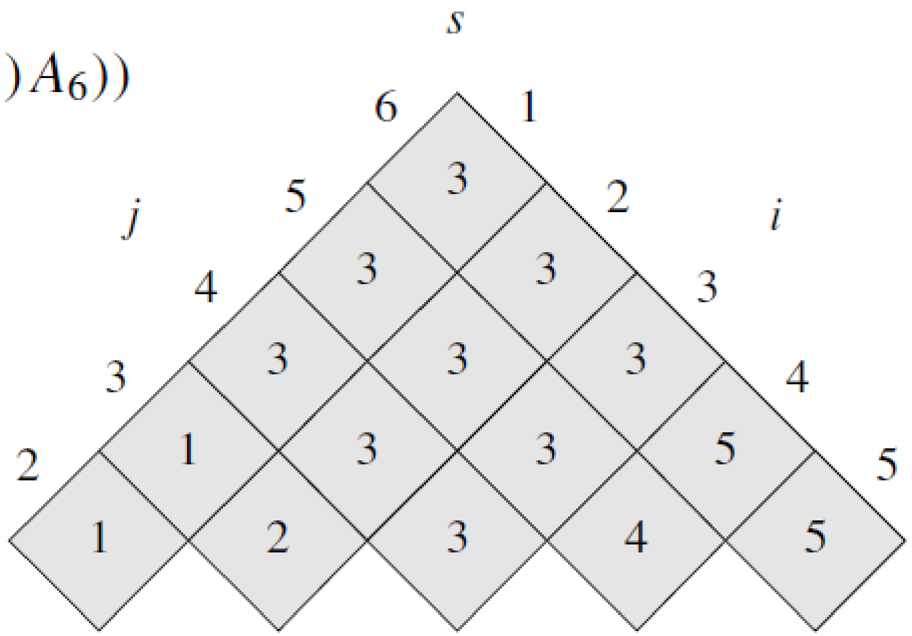
2. Matrix-chain Multiplication

matrix	dimension
A_1	30×35
A_2	35×15
A_3	15×5
A_4	5×10
A_5	10×20
A_6	20×25

2. Matrix-chain Multiplication



$((A_1(A_2A_3))((A_4A_5)A_6))$



$$m[2,5] = \min \begin{cases} m[2,2] + m[3,5] + p_1p_2p_5 = 0 + 2500 + 35 \cdot 15 \cdot 20 & = 13000\,, \\ m[2,3] + m[4,5] + p_1p_3p_5 = 2625 + 1000 + 35 \cdot 5 \cdot 20 & = 7125\,, \\ m[2,4] + m[5,5] + p_1p_4p_5 = 4375 + 0 + 35 \cdot 10 \cdot 20 & = 11375 \end{cases}$$

$$= 7125\,.$$

2. Matrix-chain Multiplication

```
PRINT-OPTIMAL-PARENS(s, i, j)
if i = j
    then print "A"_i
    else print "("
         PRINT-OPTIMAL-PARENS(s, i, s[i, j])
         PRINT-OPTIMAL-PARENS(s, s[i, j] + 1, j)
         print ")"
```

2. Matrix-chain Multiplication

15.2-2

Give a recursive algorithm MATRIX-CHAIN-MULTIPLY(A, s, i, j) that actually performs the optimal matrix-chain multiplication, given the sequence of matrices $\langle A_1, A_2, \ldots, A_n \rangle$, the s table computed by MATRIX-CHAIN-ORDER, and the indices i and j. (The initial call would be MATRIX-CHAIN-MULTIPLY$(A, s, 1, n)$.)

15.2-4

Let $R(i, j)$ be the number of times that table entry $m[i, j]$ is referenced while computing other table entries in a call of MATRIX-CHAIN-ORDER. Show that the total number of references for the entire table is

$$\sum_{i=1}^{n} \sum_{j=i}^{n} R(i, j) = \frac{n^3 - n}{3} .$$