The Matrix Chain Multiplication Problem

Source: CLRS and Dasgupta et al.

Let X and Y be two matrices of dimensions $(P_0 \times P_1)$ and $(P_1 \times P_2)$. The multiplying X with Y takes $P_0 \times P_1 \times P_2$ multiplications.

Now, matrix multiplication is not commutative, i.e., $A \times B \neq B \times A$, but it is Associative, i.e. $A \times (B \times C) = (A \times B) \times C$

For example consider four matrices A, B, C, and D, where A is a 30×1, B is 1×40, C is 40×10, and D is 10×25

Let's try a few possible orderings for $A \times B \times C \times D$ using parentheses:

$$(((A \times B) \times C) \times D) = \underset{30 \times 1 \times 40}{A \times B} + \underset{30 \times 40 \times 10}{(A \times B) \times C} + \underset{30 \times 10 \times 25}{((A \times B) \times C) \times D}$$

Total # of scalar multiplications: 20,700.

But there are other possibilities (arrangements)!

The problem is not matrix multiplication but about the number of scalar multiplications when several matrices are multiplied in chain.

Now, another possibility is

$$(A \times B) \times (C \times D)$$

$$30 \times 1 \times 40 + 40 \times 10 \times 25$$
$$+ 30 \times 40 \times 25 = 41,200$$

Signature No.

Another possibility

$$A \times ((B \times C) \times D)$$

and we'll have

$$\underbrace{1 \times 40 \times 10}_{B \times C} + \underbrace{1 \times 10 \times 25}_{(B \times C) \times D} + \underbrace{30 \times 1 \times 25}_{A \times ((B \times C) \times D)}$$

$$= 1400 \text{ multiplications!}$$

A huge performance gain compared to 41,200 multiplication operations!

The order imposed by parentheses makes the difference.

How do we find the best-possible ordering to minimize the # of scalar multiplications

Problem: let $M(i, j)$ be the minimum (optimal) no. of [scalar] multiplications necessary to compute a product of k matrices

i.e., $\prod_{i=1}^{k} M_k$ in chain

where the order of the matrix M_p is $r_p \times r_{p+1}$

Two key observations:-

1) The outermost parenthesis partitions a chain of matrices $\underbrace{(i, j)}_{\text{subproblems}}$ at some k

2) The optimal parenthesizing has an optimal ordering (not necessarily unique) on either side of k

Optimal substructure!

Date

<u>We're ready to solve it using DP:</u>

<u>Using CLRS notations</u>:-

Let $A_1, A_2, \dots, A_n$ be a chain of matrices (matrix chain). A chain product of matrices is fully-parenthesized, if it's either a <u>single</u> or a product of <u>two</u> fully parenthesized matrix products.

[Consider the problem: Given n nodes, how many different binary trees can be constructed?]

[We're talking about the same thing here!]

<u>Problem</u>: Given a chain $\langle A_1, A_2, \dots, A_n \rangle$ of n matrices, with the i^{th} matrix A_i having dimensions $P_{i-1} \times P_i$ $(1 \leq i \leq n)$, find a fully-parenthesized chain of product ~~to~~ <u>minimize</u> the total # of scalar multiplications.

<u>Constraint</u>: We can't alter the ordering in the chain (non-commutative).

Let's check <u>Brute-force</u>:- How many different orderings exist given a chain of n $(n \geq 1)$ matrices?

Date

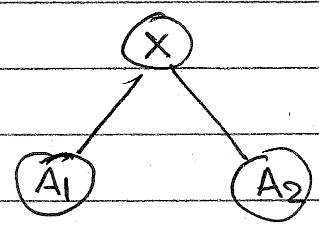
lets each matrix A_i be a node in a full binary tree.

So, for $n=1$

# of possible orderings is 1. (A_1) ; A_1

$n=2$:

$\underbrace{P(n=2)}_{\text{\# of possible orderings}} = 1$, $(A_1 \times A_2)$;

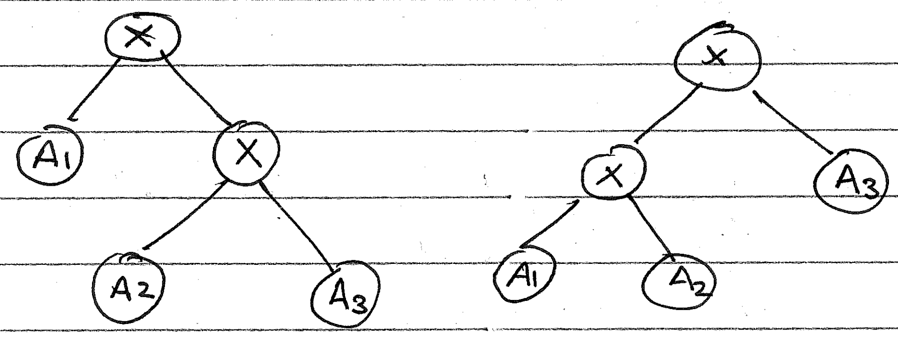


$n=3$

$P(n=3) = 2$. $(A_1 \times A_2 \times A_3)$

: $(A_1 \times (A_2 \times A_3))$ and $((A_1 \times A_2) \times A_3)$

and corresponding full binary trees



let's do that for $n=4$ $(A_1 \times A_2 \times A_3 \times A_4)$

then, we have :

$$\left.\begin{array}{l}
(A_1 \times (A_2 \times (A_3 \times A_4))) \\
(A_1 \times ((A_2 \times A_3) \times A_4)) \\
((A_1 \times A_2) \times (A_3 \times A_4)) \\
((A_1 \times (A_2 \times A_3)) \times A_4) \\
(((A_1 \times A_2) \times A_3) \times A_4)
\end{array}\right\} P(n=4) = 5$$

any pattern?

Signature ______ No. ______

Date

<u>Claim</u>: For $P(n \geq 2)$, a fully parenthesized matrix chain product is a product of <u>two</u> fully-parenthesized matrix subproducts

$\underbrace{\hspace{4em}}_{\text{a recursive approach!}}$

optimal substructure!

So, our resulting recurrence will be: as follows:

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

Aka Catalan numbers!

all k-splits in the chain

eg.

$$P(n=3) = \sum_{k=1}^{3-1} P(k) \cdot P(3-k)$$

$$= \sum_{k=1}^{2} P(k) \cdot P(3-k)$$

$$= \underbrace{P(1)}_{=1} \cdot \underbrace{P(2)}_{=1} + \underbrace{P(2)}_{=1} \cdot \underbrace{P(1)}_{=1} = 2$$

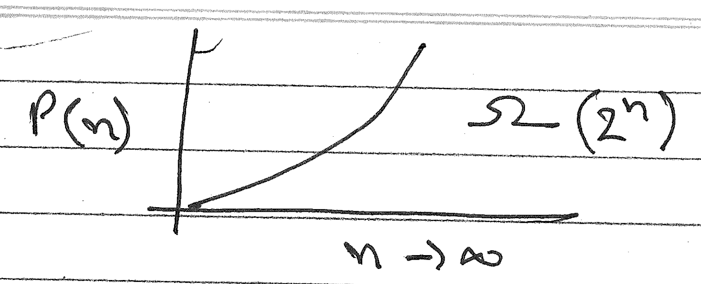
$$P(n=4) = \sum_{k}^{4-1} P(k) \cdot P(4-k)$$

$$= P(1) \cdot P(3) + P(2) \cdot P(2) + P(3) \cdot P(1)$$

$$= 5$$

The first few Catalan numbers are
1, 1, 2, 5, 14, 42, 132, 429

Signature ______ No. ______



$\Rightarrow$ Brute-Force is really Bad!

Finding the optimal substructure

Suppose, to have an optimal parenthesizing of a matrix product chain, we split the chain at some k $(1 \leq k \leq n)$.

Given a chain of matrices $A_i, A_{i+1}, \ldots, A_j$, we split it at some k, s.t., we have two subproblems $(i \leq k \leq j)$

$$(A_i A_{i+1} \ldots A_k) \text{ and } (A_{k+1} A_{k+2} \ldots A_j)$$

Now, if k optimally splits the chain, then both subproblems have an optimal substructure.

let $m[i, j]$ be the minimum # of scalar multiplications (recall LCS) to compute the product of $(j-i)$ matrices in the chain $A_i A_{i+1} A_{i+2} \ldots A_j$

What is the smallest subproblem with an optimal substructure?

When $i = j$, $m[i, j] = 0$; just a single matrix (No multiplications needed)

Signature ______ No. ______

Date

When $i < j$:

$$m[i,j] = \underbrace{m[i,k]}_{A_i A_{i+1} \ldots A_k} + m\underbrace{[k+1,j]}_{A_{k+1} \ldots A_j} + P_{i-1} P_k . P_j$$

where k [which k?] optimally splits the chain.

So, like other DP problems,

$$m[i,j] = \begin{cases} 0, & i=j \\ \min\limits_{1 \le k \le j} m[i,k] + m[k+1,j] + P_{i-1} P_k . P_j ; & i<j \end{cases}$$

We'll fill the $\underbrace{\text{table}}_{2D}$ in $O(n^2)$ with a total cost of $\Theta(n^3)$.

Recall,

$$\underbrace{\underbrace{\underbrace{m[i,k]}_{\underbrace{A_i \times A_{i+1} \times \ldots \times A_k}_{(P_{i-1} \times P_i) \quad (P_{k-1} \times P_k)}}}_{(P_{i-1} \times P_k)} + \underbrace{\underbrace{m[k+1,j]}_{\underbrace{A_{k+1} \ldots A_j}_{(P_k \times P_{k+1}) \quad (P_{j-1} \times P_j)}}}_{(P_k \times P_j)}}_{P_{i-1} P_k . P_j}$$

[There's a relation b/w Catalan Numbers and Binomial Coefficients]

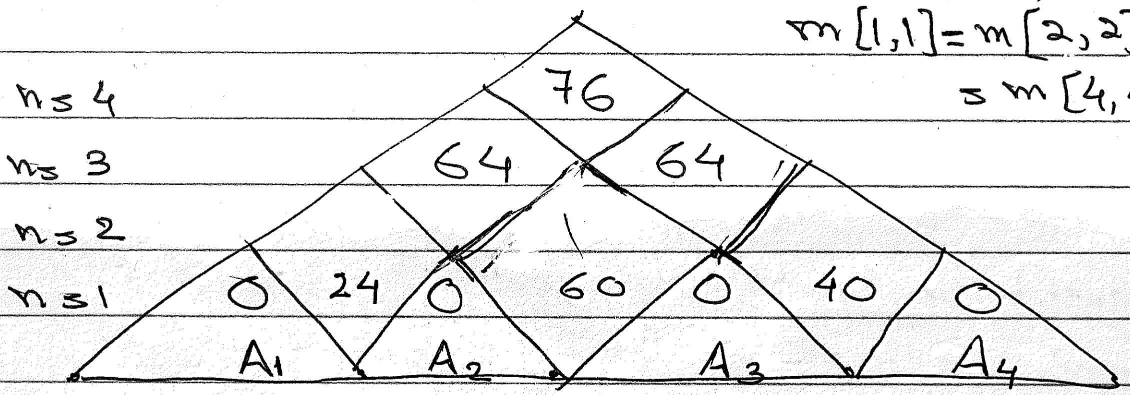
Pg 07

Date

Example: $\underset{\substack{2\times 3 \\ P_0\times P_1}}{A_1} \times \underset{\substack{3\times 4 \\ P_1\times P_2}}{A_2} \times \underset{\substack{4\times 5 \\ P_2\times P_3}}{A_3} \times \underset{\substack{5\times 2 \\ P_3\times P_4}}{A_4}$ There are 4 (four) smallest subproblems A_1, A_2, A_3, A_4

and so,

$m[1,1] = m[2,2] = m[3,3] = m[4,4] = 0$



For m=2

$$A_1 ! A_2 := \underbrace{m[1,2]}_{\substack{m[1,1]+m[2,2] \\ =0 \quad =0}} + P_0 P_1 P_2 = 0 + 2.3.4 = 24$$

likewise

$$A_2 ! A_3 = \underbrace{m[2,3]}_{\substack{m[2,2]+m[3,3] \\ =0 \quad =0}} + P_1 P_2 . P_3 = 0 + 3.4.5 = 60$$

and

$$A_3 ! A_4 = \underbrace{m[3,4]}_{m[3,3]+m[4,4]} + P_2 . P_3 . P_4 = 0 + 4.5.2 = 40$$

[for m=2, there was only one possible splits so no need for min]

For n=3

$$A_1 A_2 A_3 \begin{cases} \underset{k=1}{A_1 ! A_2 A_3} \\ \underset{k=2}{A_1 A_2 ! A_3} \end{cases} = \min \begin{cases} m[1,1] + m[2,3] + P_0 P_1 P_3, \\ m[1,2] + m[3,3] + P_0 P_2 P_3 \end{cases}$$

$$= \min(0 + 60 + 30, 24 + 0 + 40)$$
$$= 64$$

Fill the rest of the table in the Δ.

Do you remember Pascal's Δ? The Sierpenski Δ?

Signature ____ No. ____