Matrix-Chain Multiplication

Multiplying non-square matrices:

* A is $p \times \boxed{q}$, B is $\boxed{q} \times r$ → must be equal

* AB is $p \times r$ whose (i, j) entry is $\sum a_{ik} b_{kj}$

Computing AB takes $p*q*r$ scalar multipli-cations and $p(q-1)r$ scalar additions.

Suppose we have a sequence of matrices to multiply. <u>What is the best order?</u>

Given a sequence of matrices $A_1, A_2, A_3 \dots A_n$

then, Compute $C = A_1 \cdot A_2 \cdot A_3 \dots A_n$

Different way to compute C

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

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

Matrix multiplication is associative

* So output will be the same

However, time cost can be very different

Example : Suppose we have 3 matrices

$A_1 = 10 * 30$

$A_2 = 30 * 5$

$A_3 = 5 * 60$

$A = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}$ $B = \begin{bmatrix} j & m \\ k & n \\ l & o \end{bmatrix}$

2×3 (p q) $\quad$ 3×2 (q r)

$C = AB = \begin{bmatrix} aj+bk+cl & am+bn+co \\ dj+ek+fl & dm+en+fo \end{bmatrix}$

2×2 (p r)

* Number of scalar Multiplication $= p * q * r = 2 * 3 * 2 = 12$

$* \; C = \underbrace{(A_1 A_2)}_{(10*30*5)} \underbrace{A_3}_{0}$

$[\;]_{10 \times 5} \quad [\;]_{5 \times 60}$

$= (10 * 30 * 5) + 0 + (10 * 5 * 60)$

$= 1500 + 0 + 3000$

$= 4500$

$* \; C = \underbrace{A_1}_{0} \underbrace{(A_2 A_3)}_{(30*5*60)}$

$[\;]_{10 \times 30} \quad [\;]_{30 \times 60}$

$= 0 + (30 * 5 * 60) + (10 * 30 * 60)$

$= 0 + 9000 + 18000$

$= 27000$

So, parenthesize is occured for minimum cost

$\Rightarrow (A_1 A_2) A_3$ [which is 4500]

Goal: Fully parenthesize the product in a way that minimizes the number of scalar multiplications.

A possible Solution

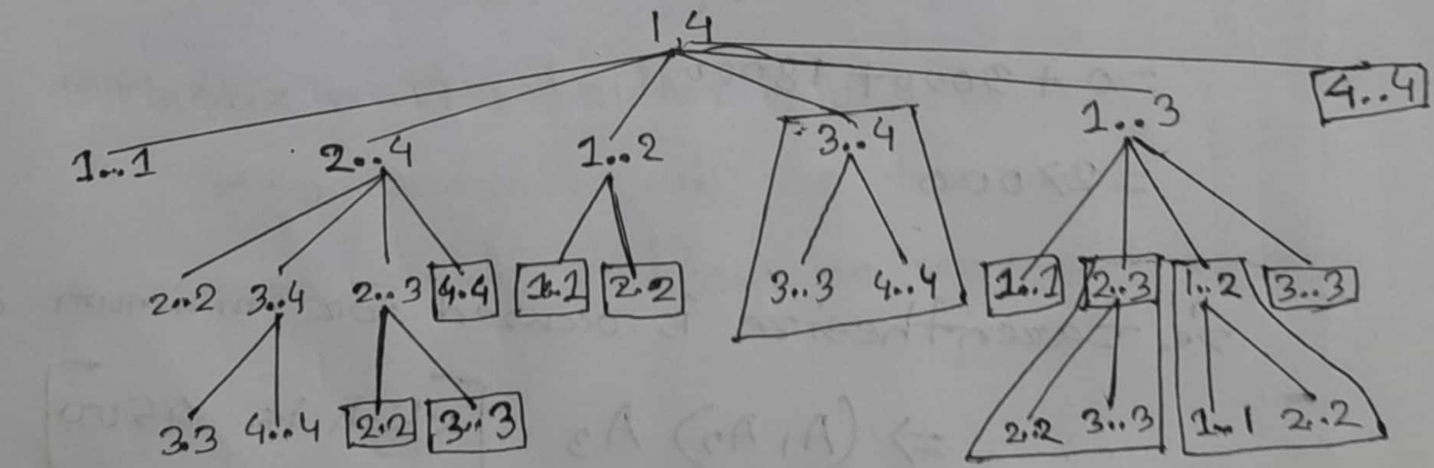
* Try all possibilities and choose the best one.
* Drawback is there are too many of them (Exponential in the num. of matrices to be multiplied
* the number of parenthesizations is

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

* The soln recurrence is $\Omega(2^n)$

No. of parenthesizations: Exponential.

Overlapping Subproblem



Suppose we have 3 matrices:

$$A_1 \times A_2 \times A_3$$

$$\begin{array}{ccc} 2\ 3 & 3\ 4 & 4\ 2 \\ d_0\ d_1 & d_1\ d_2 & d_2\ d_3 \end{array}$$

$(A_1 \times A_2) \times A_3$

2 3   3 4   4 2

$\rightarrow 2*3*4 \quad 0$

[ ] 2 4   [ ] 4 2

$\rightarrow 2*3*4+0+2*4*2$

Total $\Rightarrow$ 40

$A_1 \times (A_2 \times A_3)$

2 3   3 4   4 2

$\rightarrow 0 \quad 3*4*2$

[ ] 2 3   [ ] 3 2

$\rightarrow 0+3*4*2+2*3*2$

total $\Rightarrow$ 36

Using Dynamic Programming: M[ ][ ]

$$A_1 \times A_2 \times A_3$$

$(A_1 \times A_2) \times A_3$

M[1 2] = 24   M[3 3] = 0

[ ] 2 4   [ ] 4 2

$$M[1\ 3] = M[1\ 2] + M[3\ 3] + \overset{d_0\ d_2\ d_3}{2*4*2}$$

$$= 24+0+16$$

$$= 40$$

$A_1 \times (A_2 \times A_3)$

M[1 1] = 0   M[2 3] = 24

[ ] 2 3   [ ] 3 2

$$M[1\ 3] = M[1\ 1] + M[2\ 3] + \overset{d_0\ d_1\ d_3}{2*3*2}$$

$$= 0+24+12$$

$$= 36$$

So,

$$M[i\ j] = M[i\ k] + M[k+1, j] + d_{i-1} d_k d_j$$

$$M[i\,j] = \min_{i \le k < j} \left\{ M[i,k] + M[k+1,j] + d_{i-1} d_k d_j \right.$$

$$A_1 \times A_2 \times A_3 \times A_4$$

$$\underset{d_0}{3} \quad \underset{d_1}{2} \qquad \underset{d_1}{2} \quad \underset{d_2}{4} \qquad \underset{d_2}{4} \quad \underset{d_3}{2} \qquad \underset{d_3}{2} \quad \underset{d_4}{5}$$

$$\overset{i\ \ j}{M[1\ 4]} = \min \begin{cases} k=1 & \underline{M[1\,1]} + M[2,4] + d_0 d_1 d_4 \Rightarrow A_1(A_2A_3A_4) \\ k=2 & M[1\,2] + M[3.4] + d_0 d_2 d_4 \Rightarrow (A_1A_2)(A_3A_4) \\ k=3 & M[1\ 3] + \underline{M[4\ 4]} + d_0 d_3 d_4 \Rightarrow (A_1A_2A_3)A_4 \end{cases}$$

$$\overset{i\ \ j}{M[2\,4]} = \min \begin{cases} k=2 & \underline{M[2\,2]} + M[3\,4] + d_1 d_2 d_4 \Rightarrow A_2(A_3A_4) \\ k=3 & M[2,3] + \underline{M[4\ 4]} + d_1 d_3 d_4 \Rightarrow (A_2A_3)A_4 \end{cases}$$

So, Generate the value lower to upper

$$\overset{i\ j}{M[1\,2]} = \min_{k=1} \left\{ M[1\,1] + c[2\,2] + d_0 d_1 d_2 \right.$$

$$= 0 + 0 + 3 * 2 * 4$$

$$= 24$$

$$\overset{i\ j}{M[2\,3]} = \min_{k=2} \left\{ M[2\,2] + M[3\,3] + d_1 d_2 d_3 \right.$$

$$= 0 + 0 + 2 * 4 * 2$$

$$= 16$$

M

	1	2	3	4
1	0	24		
2		0	16	
3			0	
4				0

S

	1	2	3	4
1		1		
2			2	
3				
4				

A_1		A_2		A_3		A_4	
3	2	2	4	4	2	2	5
d_0	d_1	d_1	d_2	d_2	d_3	d_3	d_4

$$M[i,j] = \min_{i \le k < j} \left\{ M[i,k] + M[k+1,j] + d_{i-1} * d_k * d_j \right\}$$

$$M[3,4] = \min_{k=3} \left\{ M[3\,3] + M[4\,4] + d_2 d_3 d_4 \right.$$

$$= 0 + 0 + 4 * 2 * 5 = 40$$

$$M[1\,3] = \min \begin{cases} k=1 & M[1\,1] + M[2\,3] + d_0 d_1 d_3 \\ k=2 & M[1\,2] + M[3\,3] + d_0 d_2 d_3 \end{cases}$$

$$= \min \begin{cases} k=1 & 0 + 16 + 3 * 2 * 2 = 28 \\ k=2 & 24 + 0 + 3 * 4 * 2 = 48 \end{cases}$$

$$M[2,4] = \min \begin{cases} k=2 & M[2,2] + M[3\,4] + d_1 d_2 d_4 \\ k=3 & M[2\,3] + M[4\,4] + d_1 d_3 d_4 \end{cases}$$

$$= \min \begin{cases} k=2 & 0 + 40 + 2 * 4 * 5 = 80 \\ k=3 & 16 + 0 + 2 * 2 * 5 = 36 \end{cases}$$

$$M[1\,4] = \min \begin{cases} k=1 & M[1\,1] + M[2\,4] + d_0 d_1 d_4 \\ k=2 & M[1\,2] + M[3\,4] + d_0 d_2 d_4 \\ k=3 & M[1\,3] + M[4\,4] + d_0 d_3 d_4 \end{cases}$$

$$= \min \begin{cases} k=1 & 0 + 36 + 3 * 2 * 5 = 66 \\ k=2 & 24 + 40 + 3 * 4 * 5 = 124 \\ k=3 & 28 + 0 + 3 * 2 * 5 = 58 \end{cases}$$

M	1	2	3	4
1	0	24	28	58
2		0	16	36
3			0	40
4				0

S	1	2	3	4
1		1	1	3
2			2	3
3				3
4				

From 'S' table (Parenthesization):

$$A_1 A_2 A_3 A_4 \xrightarrow[S[1\,4]]{} (A_1 A_2 A_3)(A_4) \xrightarrow[S[1\,3]]{} ((A_1)(A_2 A_3))(A_4)$$

Algorithm:

```
Matrix_Chain_Order(d)
    n = d.length - 1
    let m[1..n, 1..n] and s[1..n, 1..n] be new tables
    For i = 1 to n
        m[i,i] = 0
    For l = 2 to n
        For i = 1 to n-l+1
            j = ~~i+l-1~~ i+l-1
            m[i,j] = ∞
            For k = i to j-1
                q = m[i,k] + m[k+1,j] + d[i-1] d[k] d[j]
                if m[i,j] > q
                    m[i,j] = q
                    s[i,j] = k
    return m and s
```

TC: $O(n^3)$

SC: $O(n^2)$

Problem:

Leetcode - 312. Burst Balloons

UVA - 442. Matrix Chain Multiplication