

$$A_1 A_2 \dots A_n$$

$$A_1 A_2 A_3 A_4$$

$$(A_1 A_2)(A_3 A_4)$$

$$(A_1(A_2(A_3 A_4)))$$

$$A_1 \quad p \times q$$

$$A_2 \quad q \times r$$

$$A_1 A_2 = A \quad p \times r, \quad p \times q \times r \text{ multiplicaciones}$$

Multiplicación Matrices (A, B)

Sea C de A. filas x B. columnas

$$C \leftarrow 0;$$

for $i=1$ a A. filas
 for $j=1$ a B. columnas
 for $k=1$ a A. columnas
 $C_{ij} = C_{ij} + a_{ik} \cdot b_{kj}$

return C.

$$A_1, A_2, A_3$$

$$(10 \times 100, 100 \times 5, 5 \times 50)$$

$$P_1 = 10$$

$$P_2 = 100$$

$$P_3 = 5$$

$$P_4 = 50$$

A_2 dimension $P_2 \times P_3$

$$(A_1 A_2) A_3$$

$$10 \times 100 \times 5 = 5000$$

$$\text{-----}$$

$$10 \times 5 \times 50 = 2500$$

} 7500

$$A_1 (A_2 A_3)$$

$$100 \times 5 \times 50 = 25000$$

$$\text{-----}$$

$$\rightarrow 10 \times 100 \times 50 = 50000$$

} 75000

$$A_1 - \dots - A_j$$

$$A_1 - A_k A_{k+1} - A_j$$

$$\text{-----}$$

$P(k) \neq$ diferentes parentizaciones de $A_1 - A_k$

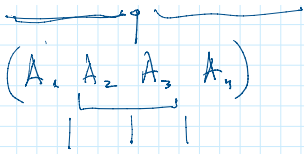
$$P(n) = \begin{cases} 1, & n=1 \\ \sum_{k=1}^{n-1} P(k)P(n-k) \end{cases}$$

$\Omega(2^n)$

- 1 - Encontrar una estructura de la solución óptima
- 2 - Refinar el valor óptimo recursivamente
- 3 - Computar el valor óptimo
- 4 - Construir la solución óptima

$$A_1 \longrightarrow A_n$$

$$(A_1 - A_k)(A_{k+1} - A_n)$$



$A_i \text{ --- } A_j$

$P_{i-1} \times P_i$ dimensions A_i

$$m[i, j] = m[i, k] + m[k+1, j] + P_{i-1} \times P_k \times P_j$$

$$i > 1$$

$$j \leq n$$

$$i \leq j$$

$$OPT[i, j] = \begin{cases} 0 \\ \min_{i \leq k < j} (OPT(i, k) + OPT(k+1, j) + P_{i-1} P_k P_j) \end{cases}$$

Solution recursive

Recursive Mat (P)

for $i = 1$ to n
 $m[i, i] = 0$

for $l = 2$ to n

for $i = 1$ to $n - l + 1$
 $j = i + l - 1$

$$m[i, j] = \infty$$

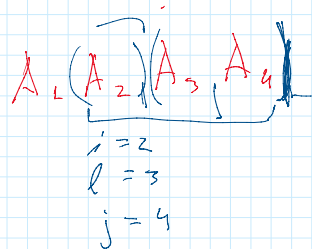
for $k = i$ to $j - 1$

$$q = m[i, k] + m[k+1, j] + P_{i-1} P_k P_j$$

if $q < m[i, j]$

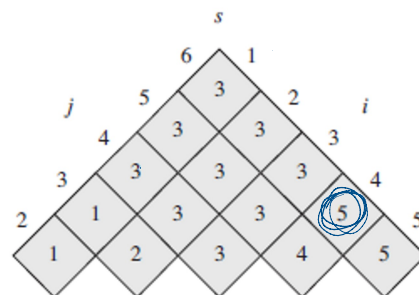
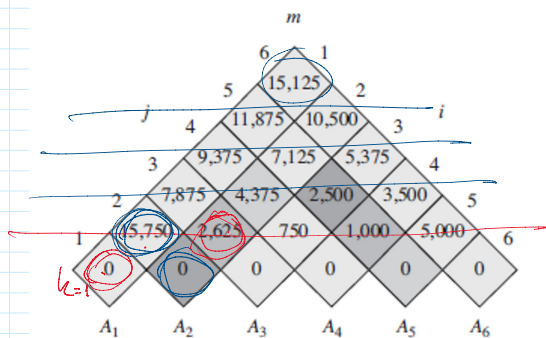
$$m[i, j] = q$$

$$s[i, j] = k$$



return m, s ;

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



Print Optimal(s, i, j)

if i == j
 print "A"

else

→ print "("

Print Optimal(s, i, s[i, j])

Print Optimal(s, s[i, j]+1, j)

Print ")"

(A₁ (A₂ A₃))(A₄ A₅)A₆)

i = 1, j = 6

i = 1, j = 3

i = 4, j = 6