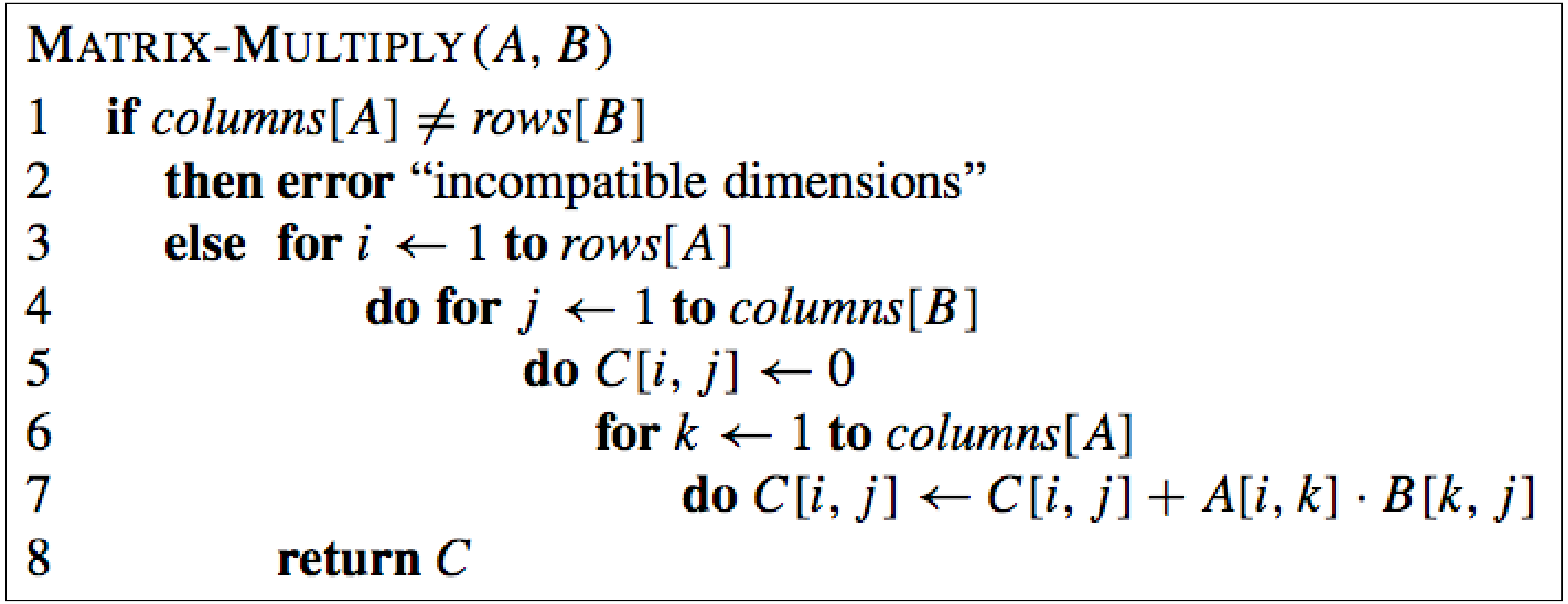
Multiplicação de Matrizes

Carlos Alberto Ynoguti

Jonas Lopes de Vilas Boas

Multiplicação de matrizes



```
MATRIX-MULTIPLY(A, B)
1  if columns[A] ≠ rows[B]
2     then error "incompatible dimensions"
3     else for i ← 1 to rows[A]
4              do for j ← 1 to columns[B]
5                     do C[i, j] ← 0
6                        for k ← 1 to columns[A]
7                            do C[i, j] ← C[i, j] + A[i, k] · B[k, j]
8           return C
```

Custo: se A é (**p** x **q**) e B é (**q** x **r**), então o número de vezes que a linha 7 será executada é **pqr**.

Problema

Multiplicar uma cadeia <A1,A2, ..., An> de matrizes.

Como a multiplicação de matrizes é associativa, podemos escolher quais queremos multiplicar primeiro.

Exemplo: A1 x A2 x A3 x A4

A1 x (A2 x (A3 x A4))

A1 x ((A2 x A3) x A4)

(A1 x (A2 x A3)) x A4

((A1 x A2) x A3) x A4

(A1 x A2) x (A3 x A4)

E daí?

Suponha que queremos calcular A1 x A2 x A3,

A1 (10 x 100), A2 (100 x 5), A3 (5 x 50)

Custos:

(A1 x A2) x A3: 7500 multiplicações escalares
A1 x (A2 x A3): 75000 multiplicações escalares

10 vezes!

Problema

Determinar a ordem em que estas matrizes devem ser multiplicadas de forma a minimizar o número de operações a serem realizadas.

Etapa 1:
estrutura de uma colocação ótima de parênteses

Sequência de matrizes a multiplicar

$$A_{i..j} = A_i \times A_{i+1} \times \cdots \times A_j$$

Queremos dividi-la da seguinte forma:

$$\underbrace{\underbrace{(A_i \times A_{i+1} \times \cdots \times A_k)}_{\text{custo 1}} \times \underbrace{(A_{k+1} \times A_{k+2} \times \cdots \times A_j)}_{\text{custo 2}}}_{\text{custo 3}}$$

de maneira a minimizar o número de operações

Etapa 2:
uma solução recursiva

m[i][j]: número mínimo de multiplicações para calcular $A_{i..j}$

Custo total: $A_{1..n}$=m[1][n]

Podemos definir m[i][j] recursivamente:

  i=j: m[i][i]=0, i=0,1,2,...,n

  i<j: m[i][k]+m[k+1][j]+p[i-1]p[k]p[j]

p armazena
as dimensões
das matrizes

Problema: não sabemos o ponto ótimo k

Etapa 2: uma solução recursiva

Fato: os valores de k vão de i até j-1

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

Cálculo dos custos ótimos

1. Custo para cadeias de 1 matriz:
   m[i][i]=0; i=1,...,n (multiplicação de cadeias de 1 matriz apenas -> nada a fazer).
2. Custo para cadeias de 2 matrizes:
   m[i][i+1]; i=1,2,...,n-1 (também é trivial)
3. Custo para cadeias de 3 matrizes:
   m[i][i+2]; i=1,2,...,n-2 (não tão trivial)

Custo para cadeias de 3 matrizes

A1 x A2 x A3

Duas opções:

- (A1 x A2) x A3 custo: m[1][2] + p[0]p[2]p[3]
- A1 x (A2 x A3) custo: p[0]p[1]p[3] + m[2][3]

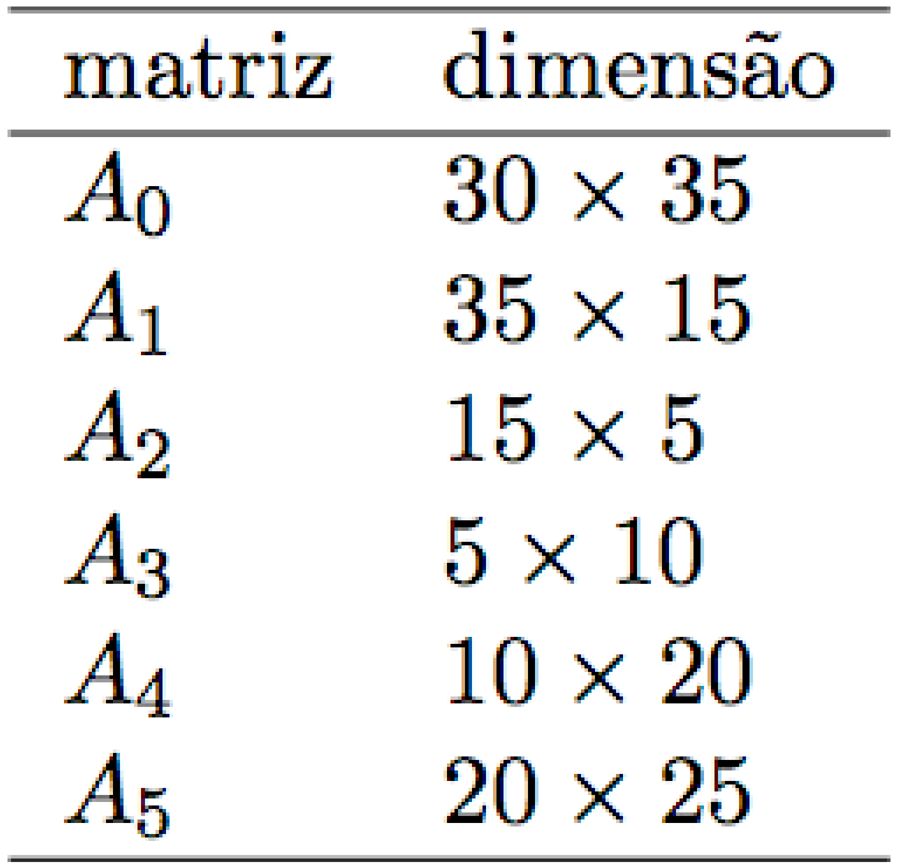
escolhemos a que der o menor custo

Se optarmos pela primeira, s[1][3]=2, senão s[1][3]=1 (ponto de corte)

Cálculo dos custos ótimos

4. Vamos nesta toada até chegar ao caso de multiplicação de n matrizes.

Exemplo



matriz	dimensão
A_0	30×35
A_1	35×15
A_2	15×5
A_3	5×10
A_4	10×20
A_5	20×25

Determinar a melhor forma de calcular $A_0 \times A_1 \times A_2 \times A_3 \times A_4 \times A_5$

Solução

Vetor das dimensões das matrizes: $p = \{30, 35, 15, 5, 10, 20, 25\}$

j

i	0	1	2	3	4	5
0	0					
1		0				
2			0			
3				0		
4					0	
5						0

m

j

i	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

s

Primeiro, definimos como 0 todas os custos de multiplicação de uma matriz (m [i][j]=0 para i=j)

Segundo, calculamos os custos de multiplicação das combinações de duas matrizes.

Solução

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

Para k = i .. j-1

$$p = \{30, 35, 15, 5, 10, 20, 25\}$$

Para k = 0

m[0][1] = m[0][0] + m[1][1] + p[-1]*p[0]*p[1]
m[0][1] = 0 + 0 + 30*35*15 = 15750

j

	0	1	2	3	4	5
0	0	15750				
1		0				
2			0			
3				0		
4					0	
5						0

i

m

j

	0	1	2	3	4	5
0		0				
1						
2						
3						
4						
5						

i

s

$(A_0) * (A_1) * A_2 * A_3 * A_4 * A_5$

Corte: 0 1 2 3 4

Segundo, calculamos os custos de multiplicação das combinações de duas matrizes.

Solução

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

Para k = i .. j-1

Para k = 1

$$p = \{30, 35, 15, 5, 10, 20, 25\}$$

m[1][2] = m[1][1] + m[2][2] + p[0]*p[1]*p[2]

m[1][2] = 0 + 0 + 35*15*5 = 2625

j

i

	0	1	2	3	4	5
0	0	15750				
1		0	2625			
2			0			
3				0		
4					0	
5						0

m

j

i

	0	1	2	3	4	5
0		0				
1			1			
2						
3						
4						
5						

s

$A_0 * (A_1) * (A_2) * A_3 * A_4 * A_5$

Corte: 0 1 2 3 4

Segundo, calculamos os custos de multiplicação das combinações de duas matrizes.

Solução

$$m[i][j] = \begin{cases} 0, & i = j \\ \min\limits_{i \leq k < j} \{m[i][k] + m[k+1][j] + p_{i-1}p_kp_j\}, & i < j \end{cases}$$

Para k = i .. j-1

$$p = \{30, 35, 15, 5, 10, 20, 25\}$$

Para k = 2

m[2][3] = m[2][2] + m[3][3] + p[1]*p[2]*p[3]
m[2][3] = 0 + 0 + 15*5*10 = 750

j

i	0	1	2	3	4	5
0	0	15750				
1		0	2625			
2			0	750		
3				0		
4					0	
5						0

m

j

i	0	1	2	3	4	5
0		0				
1			1			
2				2		
3						
4						
5						

s

$A_0 * A_1 * (A_2) \downarrow (A_3) * A_4 * A_5$

Corte: 0 1 2 3 4

Segundo, calculamos os custos de multiplicação das combinações de duas matrizes.

Solução

$$m[i][j] = \begin{cases} 0, & i = j \\ \min\limits_{i \leq k < j} \{m[i][k] + m[k+1][j] + p_{i-1}p_kp_j\}, & i < j \end{cases}$$

Para k = i .. j-1

$$p = \{30, 35, 15, 5, 10, 20, 25\}$$

Para k = 3

m[3][4] = m[3][3] + m[4][4] + p[2]*p[3]*p[4]
m[3][4] = 0 + 0 + 5*10*20 = 1000

j

i	0	1	2	3	4	5
0	0	15750				
1		0	2625			
2			0	750		
3				0	1000	
4					0	
5						0

m

j

i	0	1	2	3	4	5
0		0				
1			1			
2				2		
3					3	
4						
5						

s

$A_0 * A_1 * A_2 * (A_3) \downarrow * (A_4) * A_5$

Corte: 0 1 2 3 4

Segundo, calculamos os custos de multiplicação das combinações de duas matrizes.

Solução

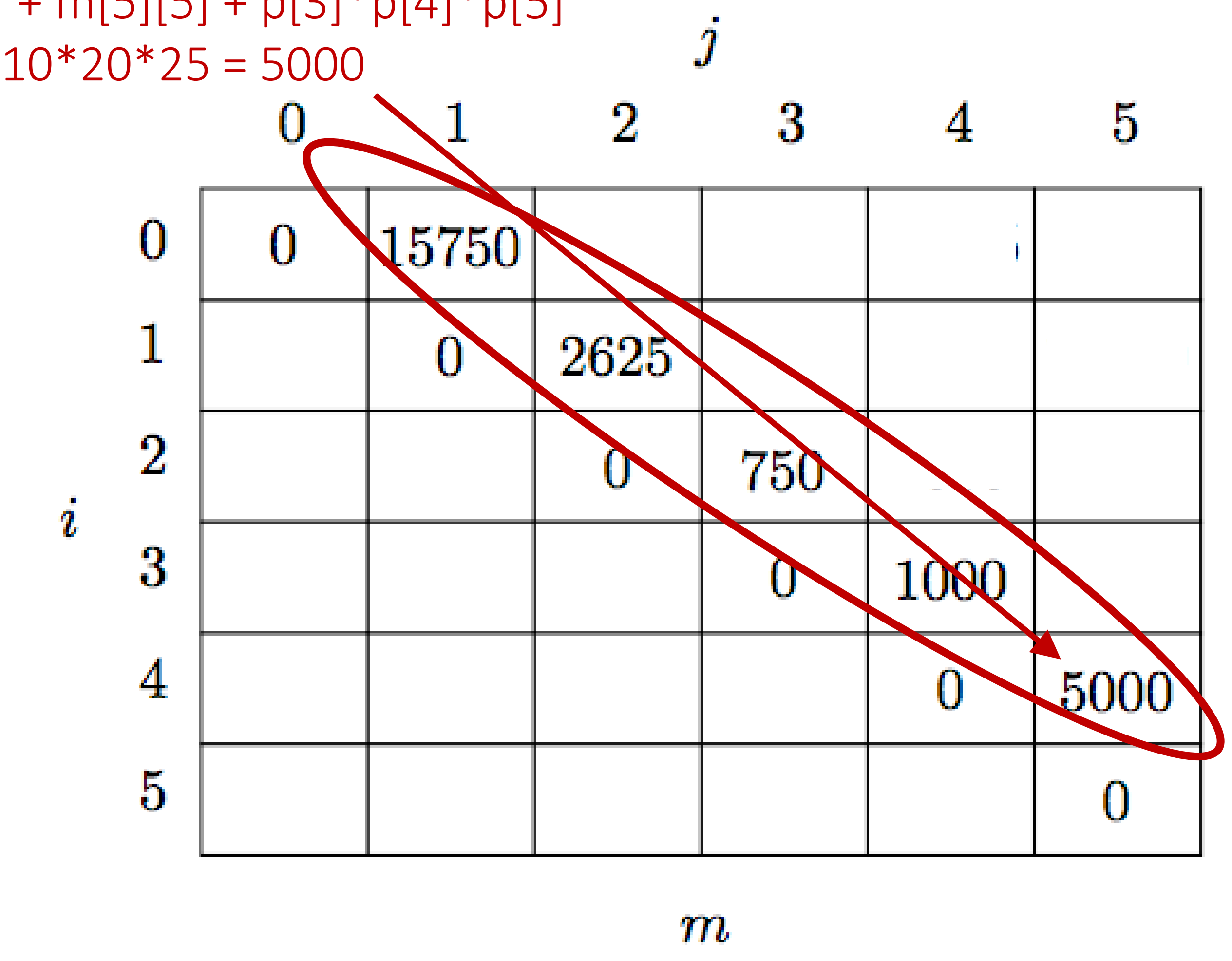
$$m[i][j] = \begin{cases} 0, & i = j \\ \min\limits_{i \leq k < j} \{m[i][k] + m[k+1][j] + p_{i-1}p_kp_j\}, & i < j \end{cases}$$

Para k = i .. j-1

$$p = \{30, 35, 15, 5, 10, 20, 25\}$$

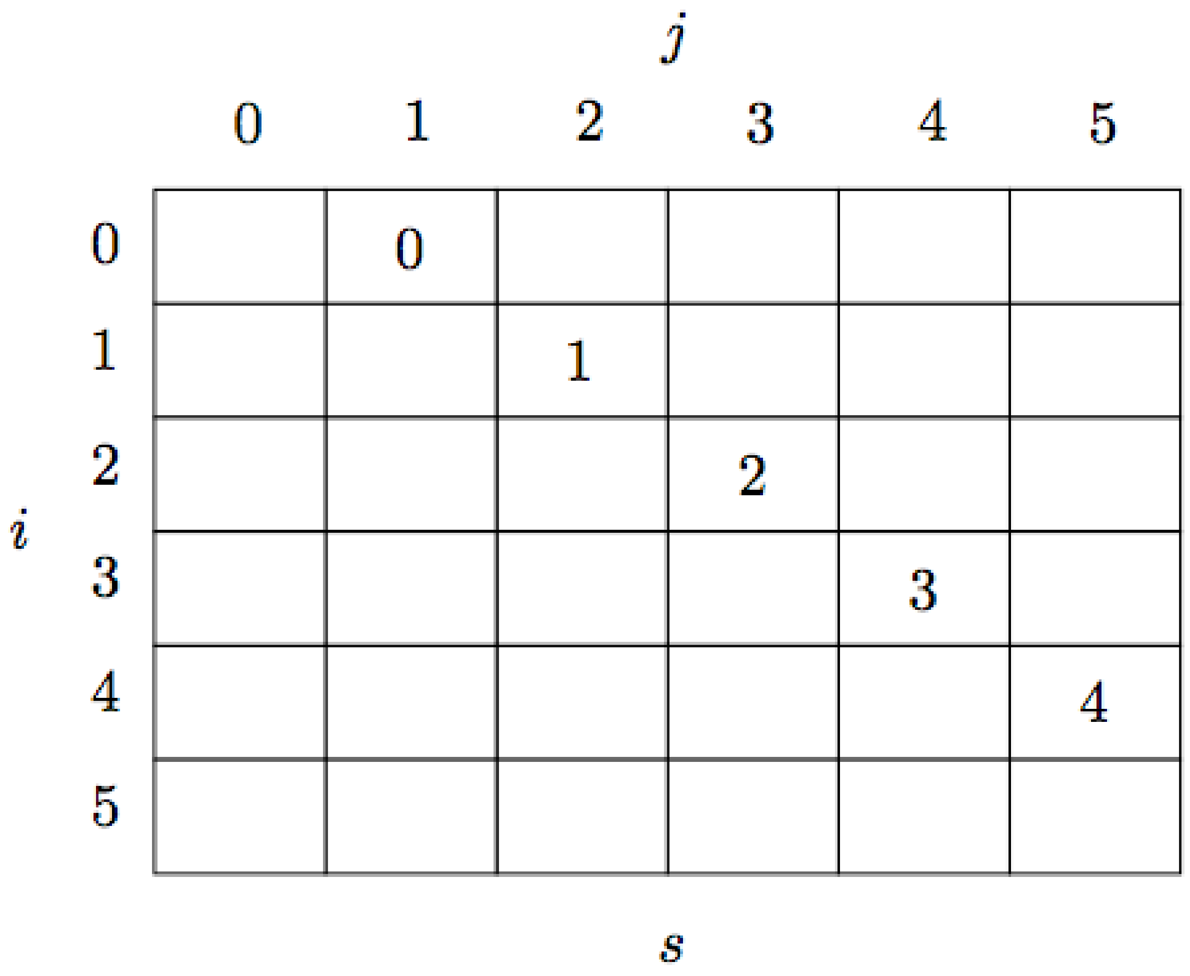
Para k = 4

m[4][5] = m[4][4] + m[5][5] + p[3]*p[4]*p[5]
m[4][5] = 0 + 0 + 10*20*25 = 5000



	0	1	2	3	4	5
0	0	15750				
1		0	2625			
2			0	750		
3				0	1000	
4					0	5000
5						0

m



	0	1	2	3	4	5
0		0				
1			1			
2				2		
3					3	
4						4
5						

s

$A_0 * A_1 * A_2 * A_3 * (A_4) * (A_5)$

Corte: 0 1 2 3 4

Terceiro, calculamos os custos de multiplicação das combinações de três matrizes.

Solução

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

Para k = i .. j-1

$$p = \{30, 35, 15, 5, 10, 20, 25\}$$

Para k = 0

m[0][2] = m[0][0] + m[1][2] + p[-1]*p[0]*p[2]
m[0][2] = 0 + 2625 + 30*35*5 = 2625 + 5250 = **7875**

j

	0	1	2	3	4	5
0	0	15750	7875			
1		0	2625			
2			0	750		
3				0	1000	
4					0	5000
5						0

i

m

j

	0	1	2	3	4	5
0		0	0			
1			1			
2				2		
3					3	
4						4
5						

i

s

Para k = 1

m[0][2] = m[0][1] + m[2][2] + p[-1]*p[1]*p[2]
m[0][2] = 15750 + 0 + 30*15*5 = 15750 + 5250 = **18000**

$(A_0) * (A_1 * A_2) * A_3 * A_4 * A_5$

Corte: 0 1 2 3 4

Terceiro, calculamos os custos de multiplicação das combinações de três matrizes.

Solução

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

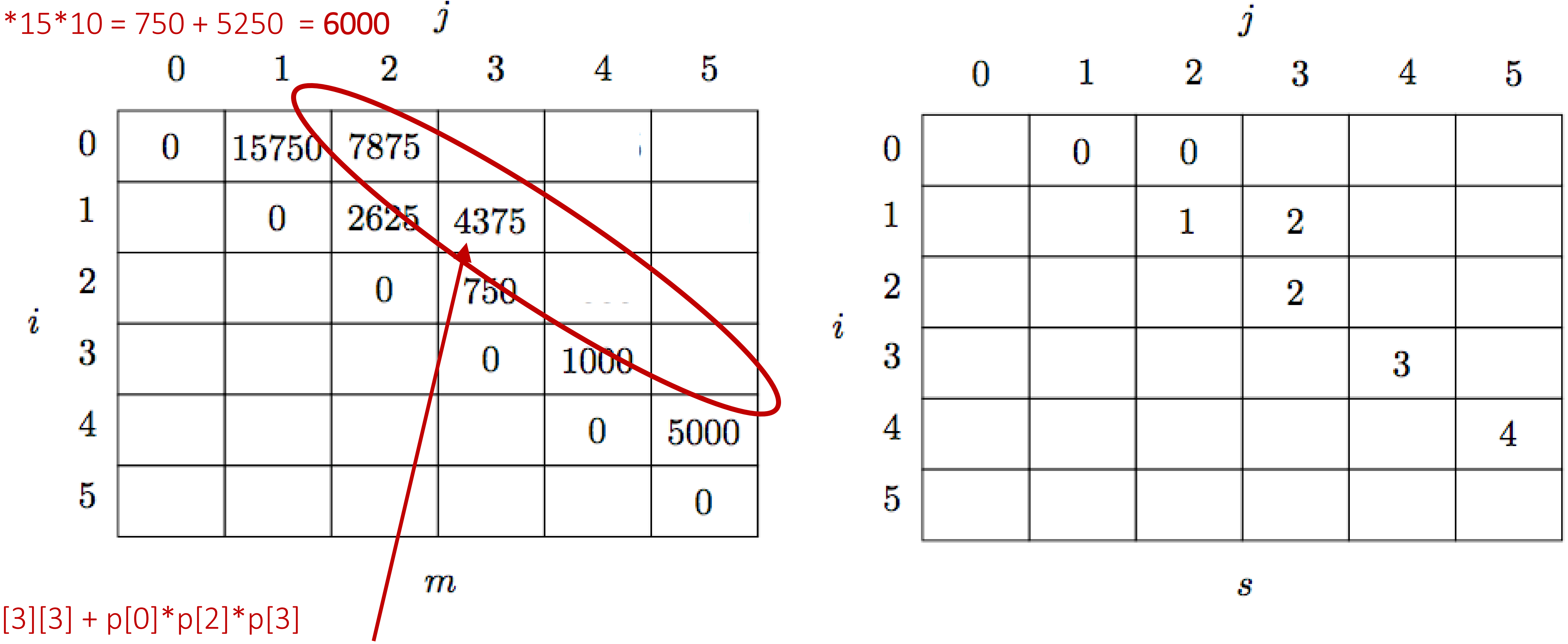
Para k = i .. j-1

$$p = \{30, 35, 15, 5, 10, 20, 25\}$$

Para k = 1

m[1][3] = m[1][1] + m[2][3] + p[0]*p[1]*p[3]

m[1][3] = 0 + 750 + 35*15*10 = 750 + 5250 = **6000**



j

i	0	1	2	3	4	5
0	0	15750	7875			
1		0	2625	4375		
2			0	750		
3				0	1000	
4					0	5000
5						0

m

j

i	0	1	2	3	4	5
0		0	0			
1			1	2		
2				2		
3					3	
4						4
5						

s

Para k = 2

m[1][3] = m[1][2] + m[3][3] + p[0]*p[2]*p[3]

m[0][2] = 2625 + 0 + 35*5*10 = 2625 + 1750 = **4375**

$(A_0)*(A_1 * A_2) \downarrow * (A_3)* A_4 * A_5$

Corte: 0 1 2 3 4

Solução

Terceiro, calculamos os custos de multiplicação das combinações de três matrizes.

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

Para k = i .. j-1

$$p = \{30, 35, 15, 5, 10, 20, 25\}$$

Para k = 2

m[2][4] = m[2][2] + m[3][4] + p[1]*p[2]*p[4]
m[2][4] = 0 + 1000 + 15*5*20 = 1000 + 1500 = **2500**

j

i	0	1	2	3	4	5
0	0	15750	7875			
1		0	2625	4375		
2			0	750	2500	
3				0	1000	
4					0	5000
5						0

m

j

i	0	1	2	3	4	5
0		0	0			
1			1	2		
2				2	2	
3					3	
4						4
5						

s

Para k = 3

m[2][4] = m[2][3] + m[4][4] + p[1]*p[3]*p[4]
m[2][4] = 750 + 0 + 15*10*20 = 750 + 3000 = **3750**

$$(A_0)*(A_1 * A_2) \downarrow *(A_3 * A_4)* A_5$$

Corte: 0 1 2 3 4

Terceiro, calculamos os custos de multiplicação das combinações de três matrizes.

Solução

$$m[i][j] = \begin{cases} 0, & i = j \\ \min\limits_{i \leq k < j} \{m[i][k] + m[k+1][j] + p_{i-1}p_kp_j\}, & i < j \end{cases}$$

Para k = i .. j-1

$$p = \{30, 35, 15, 5, 10, 20, 25\}$$

Para k = 3

m[3][5] = m[3][3] + m[4][5] + p[2]*p[3]*p[5]

m[3][5] = 0 + 5000 + 5*10*25 = 5000 + 1250 = **6250**

j

i

	0	1	2	3	4	5
0	0	15750	7875			
1		0	2625	4375		
2			0	750	2500	
3				0	1000	3500
4					0	5000
5						0

m

j

i

	0	1	2	3	4	5
0		0	0			
1			1	2		
2				2	2	
3					3	4
4						4
5						

s

Para k = 4

m[3][5] = m[3][4] + m[5][5] + p[2]*p[4]*p[5]

m[3][5] = 1000 + 0 + 5*20*25 = 1000 + 2500 = **3500**

$(A_0)*(A_1 * A_2) *(A_3 * A_4) * (A_5)$

Corte: 0 1 2 3 4

Continuando, calculamos os custos de multiplicação das combinações de para todos os conjuntos de matrizes, aproveitando os resultados anteriores.

Solução

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

Para k = i .. j-1

$$p = \{30, 35, 15, 5, 10, 20, 25\}$$

Para k = 0

m[0][3] = m[0][0] + m[1][3] + p[-1]*p[0]*p[3]
m[0][3] = 0 + 4375 + 30*35*10 = **14875**

j

i	0	1	2	3	4	5
0	0	15750	7875	9375		
1		0	2625	4375		
2			0	750	2500	
3				0	1000	3500
4					0	5000
5						0

m

j

i	0	1	2	3	4	5
0		0	0	2		
1			1	2		
2				2	2	
3					3	4
4						4
5						

s

Para k = 1

m[0][3] = m[0][1] + m[2][3] + p[-1]*p[1]*p[3]
m[0][3] = 15750 + 750 + 4500 = **21000**

Para k = 2

m[0][3] = m[0][2] + m[3][3] + p[-1]*p[2]*p[3]
m[0][3] = 7875 + 0 + 1500 = **9375**

$$((A_0)*(A_1 * A_2))\downarrow*((A_3 * A_4) * (A_5))$$

Corte: 0 1 2 3 4

Solução

$$p = \{30, 35, 15, 5, 10, 20, 25\}$$

j

i

	0	1	2	3	4	5
0	0	15750	7875	9375	11875	15125
1		0	2625	4375	7125	10500
2			0	750	2500	5375
3				0	1000	3500
4					0	5000
5						0

m

j

i

	0	1	2	3	4	5
0		0	0	2	2	2
1			1	2	2	2
2				2	2	2
3					3	4
4						4
5						

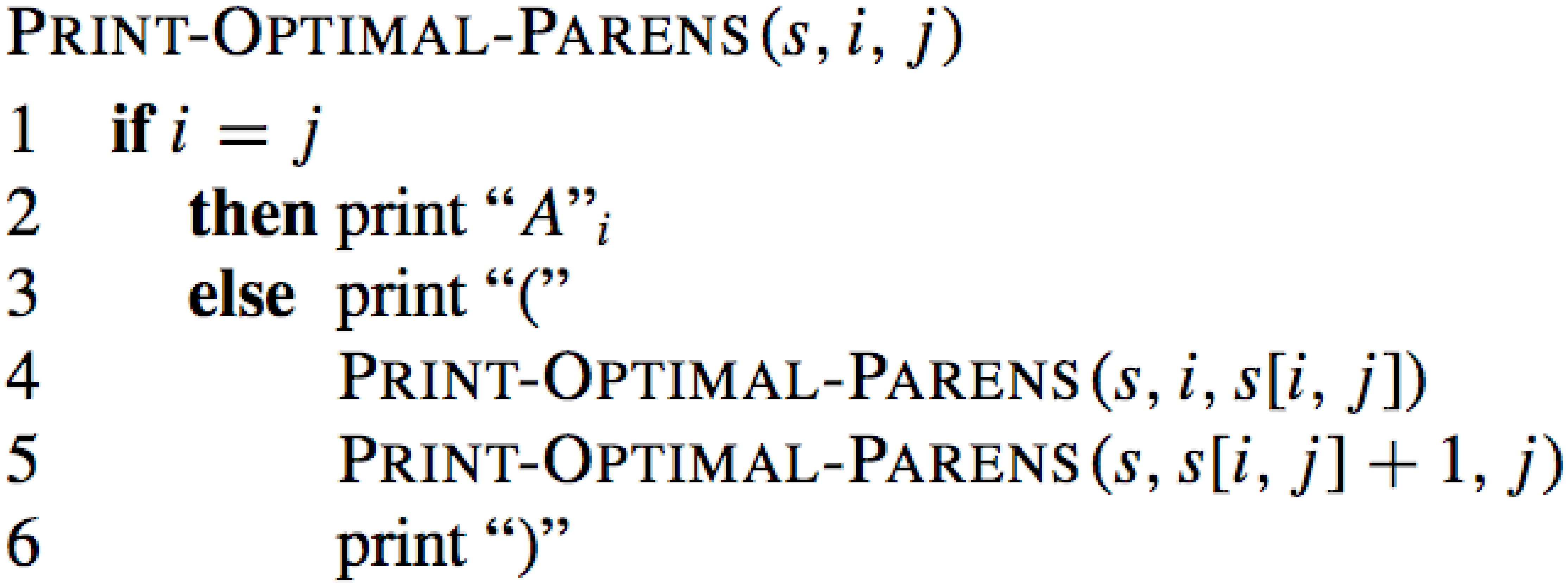
s

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

matriz	dimensão
A_0	30×35
A_1	35×15
A_2	15×5
A_3	5×10
A_4	10×20
A_5	20×25

```
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 $n$            $\triangleright$ $l$ is the chain length.
    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 $s$
```

Determinando a ordem ótima



```
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 ")"
```

Para o exemplo anterior, a saída seria:

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