Programação dinâmica

CLRS 15.2–15.3

= "recursão–com–tabela"
= transformação inteligente de recursão em iteração

Multiplicação iterada de matrizes

Se A é $p \times q$ e B é $q \times r$ então AB é $p \times r$.

$(AB)[i,j] = \sum_k A[i,k]\,B[k,j]$

MULT-MAT (p, A, q, B, r)

```
para i ← 1 até p faça
    para j ← 1 até r faça
        AB[i,j] ← 0
        para k ← 1 até q faça
            AB[i,j] ← AB[i,j] + A[i,k] · B[k,j]
```

Número de multiplicações escalares $= p \cdot q \cdot r$

Multiplicação iterada

Problema: Encontrar número mínimo de multiplicações escalares necessário para calcular produto $A_1 A_2 \cdots A_n$.

$$\begin{array}{ccccccccc} p[0] & & p[1] & & p[2] & \dots & p[n-1] & & p[n] \\ & A_1 & & A_2 & & \dots & & A_n & \end{array}$$

cada A_i é $p[i-1] \times p[i]$ $(A_i[1\,..\,p[i-1], 1\,..\,p[i]])$

Exemplo: $A_1 \cdot A_2 \cdot A_3$

$$\begin{array}{ccccccc} 10 & A_1 & 100 & A_2 & 5 & A_3 & 50 \end{array}$$

$((A_1\, A_2)\, A_3)$	7500	multiplicações escalares
$(A_1\, (A_2\, A_3))$	75000	multiplicações escalares

Soluções ótimas contêm soluções ótimas

Se

$$(A_1 A_2)\,(A_3((A_4 A_5)A_6))$$

é ordem ótima de multiplicação então

$$(A_1 A_2) \quad \text{e} \quad (A_3((A_4 A_5)A_6))$$

também são ordens ótimas.

Soluções ótimas contêm soluções ótimas

Se

$$(A_1 A_2)\,(A_3((A_4 A_5) A_6))$$

é ordem ótima de multiplicação então

$$(A_1 A_2) \quad \text{e} \quad (A_3((A_4 A_5) A_6))$$

também são ordens ótimas.

Decomposição: $(A_i \cdots A_k)\,(A_{k+1} \cdots A_j)$

$m[i,j] =$ número mínimo de multiplicações escalares
para calcular $A_i \cdots A_j$

Recorrência

$m[i,j] =$ número mínimo de multiplicações escalares para calcular $A_i \cdots A_j$

se $i = j$ então $m[i,j] = 0$

se $i < j$ então

$$m[i,j] = \min_{i \leq k < j} \{\, m[i,k] + p[i-1]p[k]p[j] + m[k+1,j] \,\}$$

Exemplo:

$$m[3,7] = \min_{3 \leq k < 7} \{\, m[3,k] + p[2]p[k]p[7] + m[k+1,7] \,\}$$

Algoritmo recursivo

Recebe $p[i-1\,..\,j]$ e devolve $m[i,j]$

REC-MAT-CHAIN (p, i, j)

se $i = j$
  então devolva 0
$m[i,j] \leftarrow \infty$
para $k \leftarrow i$ **até** $j-1$ **faça**
  $q_1 \leftarrow$ REC-MAT-CHAIN (p, i, k)
  $q_2 \leftarrow$ REC-MAT-CHAIN $(p, k+1, j)$
  $q \leftarrow q_1 + p[i-1]p[k]p[j] + q_2$
  se $q < m[i,j]$
    então $m[i,j] \leftarrow q$
devolva $m[i,j]$

Consumo de tempo?

Consumo de tempo

A plataforma utilizada nos experimentos é um PC rodando Linux Debian ?.? com um processador Pentium II de 233 MHz e 128MB de memória RAM .

O programa foi compilado com o gcc versão ?? e opção de compilação "-O2".

n	3	6	10	20	25
tempo	$0.0s$	$0.0s$	$0.01s$	$201s$	$567m$

Consumo de tempo

$T(n) =$ número comparações entre q e $m[\star, \star]$
na linha 8 quando $n := j - i + 1$

$$
\begin{aligned}
T(1) &= 0 \\
T(n) &= \sum_{h=1}^{n-1} (T(h) + T(n-h) + 1) = 2\sum_{h=2}^{n-1} T(h) + (n-1) \\
&= 2(T(2) + \cdots + T(n-1)) + (n-1) \quad \text{para } n \geq 2
\end{aligned}
$$

Consumo de tempo

$T(n) =$ número comparações entre q e $m[\star, \star]$
na linha 8 quando $n := j - i + 1$

$$T(1) = 0$$

$$T(n) = \sum_{h=1}^{n-1} (T(h) + T(n-h) + 1) = 2\sum_{h=2}^{n-1} T(h) + (n-1)$$

$$= 2(T(2) + \cdots + T(n-1)) + (n-1) \text{ para } n \geq 2$$

Considere a mesma fórmula para $n - 1$:

$$T(n-1) = 2(T(2) + \cdots + T(n-2)) + (n-2)$$

e subtraia a primeira da segunda.

Consumo de tempo

$T(n) =$ número comparações entre q e $m[\star, \star]$
na linha 8 quando $n := j - i + 1$

$$T(n) \;=\; 2(T(2) + \cdots + T(n{-}1)) \;+\; (n-1)$$

Considere a mesma fórmula para $n-1$:

$$T(n-1) \;=\; 2(T(2) + \cdots + T(n{-}2)) \;+\; (n-2)$$

e subtraia a primeira da segunda:

Consumo de tempo

$T(n) =$ número comparações entre q e $m[\star, \star]$
na linha 8 quando $n := j - i + 1$

$$T(n) \;=\; 2(T(2) + \cdots + T(n{-}1)) \;+\; (n-1)$$

Considere a mesma fórmula para $n - 1$:

$$T(n-1) \;=\; 2(T(2) + \cdots + T(n{-}2)) \;+\; (n-2)$$

e subtraia a primeira da segunda:

$$T(n) - T(n-1) \;=\; 2\,T(n-1) + 1.$$

Logo $T(n) \;=\; 3\,T(n-1) + 1.$

Consumo de tempo

$T(n) =$ número comparações entre q e $m[\star,\star]$
na linha 8 quando $n := j - i + 1$

$$T(n) \;=\; 2(T(2) + \cdots + T(n-1)) \;+\; (n-1)$$

Considere a mesma fórmula para $n-1$:

$$T(n-1) \;=\; 2(T(2) + \cdots + T(n-2)) \;+\; (n-2)$$

e subtraia a primeira da segunda:

$$T(n) - T(n-1) \;=\; 2\,T(n-1) + 1.$$

Logo $T(n) \;=\; 3\,T(n-1) + 1.$

Fácil verificar que $T(n) \geq \frac{3^{n-1}-1}{2}$ para $n \geq 1$.

Recorrência

n	1	2	3	4	5	6	7	8
$T(n)$	0	1	4	13	40	121	364	1093
$3^{n-1}-1$	0	2	8	26	80	242	728	2186

Prova: Para $n = 1$, $T(1) = 0 = (1-1)/2$.

Recorrência

n	1	2	3	4	5	6	7	8
$T(n)$	0	1	4	13	40	121	364	1093
$3^{n-1}-1$	0	2	8	26	80	242	728	2186

Prova: Para $n = 1$, $T(1) = 0 = (1-1)/2$.

Para $n \geq 2$,

$$
\begin{aligned}
T(n) &= 3T(n-1) + 1 \\
&\overset{\text{hi}}{=} 3\left(\frac{3^{n-1}-1}{2}\right) + 1 \\
&= \frac{3^n - 3}{2} + 1 \;=\; \frac{3^n - 3 + 2}{2} \\
&= \frac{3^n - 1}{2}.
\end{aligned}
$$

Conclusão

$T(n) \geq \frac{3^{n-1}-1}{2}$ para $n \geq 1$.

O consumo de tempo do algoritmo REC-MAT-CHAIN é $\Omega(3^n)$.

Resolve subproblemas muitas vezes

$p[0] = 10 \quad p[1] = 100 \quad p[2] = 5 \quad p[3] = 50$

```
REC-MAT-CHAIN(p, 1, 3)
  REC-MAT-CHAIN(p, 1, 1)
  REC-MAT-CHAIN(p, 2, 3)
    REC-MAT-CHAIN(p, 2, 2)
    REC-MAT-CHAIN(p, 3, 3)
  REC-MAT-CHAIN(p, 1, 2)
    REC-MAT-CHAIN(p, 1, 1)
    REC-MAT-CHAIN(p, 2, 2)
  REC-MAT-CHAIN(p, 3, 3)
```

Número mínimo de mults = 7500

Resolve subproblemas muitas vezes

```
REC-MAT-CHAIN(p, 1, 5)
  REC-MAT-CHAIN(p, 1, 1)
  REC-MAT-CHAIN(p, 2, 5)
    REC-MAT-CHAIN(p, 2, 2)
    REC-MAT-CHAIN(p, 3, 5)
      REC-MAT-CHAIN(p, 3, 3)
      REC-MAT-CHAIN(p, 4, 5)
        REC-MAT-CHAIN(p, 4, 4)
        REC-MAT-CHAIN(p, 5, 5)
      REC-MAT-CHAIN(p, 3, 4)
        REC-MAT-CHAIN(p, 3, 3)
        REC-MAT-CHAIN(p, 4, 4)
      REC-MAT-CHAIN(p, 5, 5)
    REC-MAT-CHAIN(p, 2, 3)
      REC-MAT-CHAIN(p, 2, 2)
      REC-MAT-CHAIN(p, 3, 3)
    REC-MAT-CHAIN(p, 4, 5)
      REC-MAT-CHAIN(p, 4, 4)
      REC-MAT-CHAIN(p, 5, 5)
    REC-MAT-CHAIN(p, 2, 4)
      REC-MAT-CHAIN(p, 2, 2)
      REC-MAT-CHAIN(p, 3, 4)
        REC-MAT-CHAIN(p, 3, 3)
        REC-MAT-CHAIN(p, 4, 4)
      REC-MAT-CHAIN(p, 2, 3)
        REC-MAT-CHAIN(p, 2, 2)
        REC-MAT-CHAIN(p, 3, 3)
      REC-MAT-CHAIN(p, 4, 4)
    REC-MAT-CHAIN(p, 5, 5)
  REC-MAT-CHAIN(p, 1, 2)
    REC-MAT-CHAIN(p, 1, 1)
    REC-MAT-CHAIN(p, 2, 2)
  REC-MAT-CHAIN(p, 3, 5)
    REC-MAT-CHAIN(p, 3, 3)
    REC-MAT-CHAIN(p, 4, 5)
      REC-MAT-CHAIN(p, 4, 4)
      REC-MAT-CHAIN(p, 5, 5)
    REC-MAT-CHAIN(p, 3, 4)
      REC-MAT-CHAIN(p, 3, 3)
      REC-MAT-CHAIN(p, 4, 4)
    REC-MAT-CHAIN(p, 5, 5)
  REC-MAT-CHAIN(p, 1, 3)
    REC-MAT-CHAIN(p, 1, 1)
    REC-MAT-CHAIN(p, 2, 3)
      REC-MAT-CHAIN(p, 2, 2)
      REC-MAT-CHAIN(p, 3, 3)
    REC-MAT-CHAIN(p, 1, 2)
      REC-MAT-CHAIN(p, 1, 1)
      REC-MAT-CHAIN(p, 2, 2)
    REC-MAT-CHAIN(p, 3, 3)
  REC-MAT-CHAIN(p, 4, 5)
    REC-MAT-CHAIN(p, 4, 4)
    REC-MAT-CHAIN(p, 5, 5)
  REC-MAT-CHAIN(p, 1, 4)
    REC-MAT-CHAIN(p, 1, 1)
    REC-MAT-CHAIN(p, 2, 4)
      REC-MAT-CHAIN(p, 2, 2)
      REC-MAT-CHAIN(p, 3, 4)
        REC-MAT-CHAIN(p, 3, 3)
        REC-MAT-CHAIN(p, 4, 4)
      REC-MAT-CHAIN(p, 2, 3)
        REC-MAT-CHAIN(p, 2, 2)
        REC-MAT-CHAIN(p, 3, 3)
      REC-MAT-CHAIN(p, 4, 4)
    REC-MAT-CHAIN(p, 1, 2)
      REC-MAT-CHAIN(p, 1, 1)
      REC-MAT-CHAIN(p, 2, 2)
    REC-MAT-CHAIN(p, 3, 4)
      REC-MAT-CHAIN(p, 3, 3)
      REC-MAT-CHAIN(p, 4, 4)
    REC-MAT-CHAIN(p, 1, 3)
      REC-MAT-CHAIN(p, 1, 1)
      REC-MAT-CHAIN(p, 2, 3)
        REC-MAT-CHAIN(p, 2, 2)
        REC-MAT-CHAIN(p, 3, 3)
      REC-MAT-CHAIN(p, 1, 2)
        REC-MAT-CHAIN(p, 1, 1)
        REC-MAT-CHAIN(p, 2, 2)
      REC-MAT-CHAIN(p, 3, 3)
    REC-MAT-CHAIN(p, 4, 4)
  REC-MAT-CHAIN(p, 5, 5)
```

Programação dinâmica

Cada subproblema

$$A_i \cdots A_j$$

é resolvido uma só vez.

Em que ordem calcular os componentes da tabela m?

Para calcular $m[2, 6]$ preciso de ...

Programação dinâmica

Cada subproblema

$$A_i \cdots A_j$$

é resolvido uma só vez.

Em que ordem calcular os componentes da tabela m?

Para calcular $m[2,6]$ preciso de ...

$m[2,2]$, $m[2,3]$, $m[2,4]$, $m[2,5]$ e de
$m[3,6]$, $m[4,6]$, $m[5,6]$, $m[6,6]$.

Programação dinâmica

Cada subproblema

$$A_i \cdots A_j$$

é resolvido uma só vez.

Em que ordem calcular os componentes da tabela m?

Para calcular $m[2,6]$ preciso de ...

$m[2,2]$, $m[2,3]$, $m[2,4]$, $m[2,5]$ e de
$m[3,6]$, $m[4,6]$, $m[5,6]$, $m[6,6]$.

Calcule todos os $m[i,j]$ com $j-i+1=2$,
depois todos com $j-i+1=3$,
depois todos com $j-i+1=4$,
etc.

Programação dinâmica

	1	2	3	4	5	6	7	8	j
1	0								
2		0	$\star$	$\star$	$\star$	??			
3			0			$\star$			
4				0		$\star$			
5					0	$\star$			
6						0			
7							0		
8								0	
i									

Simulação

$p[0]{=}10$ $p[1]{=}10$ $p[2]{=}20$ $p[3]{=}30$ $p[4]{=}10$ $p[5]{=}15$ $p[6]{=}30$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

i \ j	1	2	3	4	5	6
1	0	??				
2		0				
3			0			
4				0		
5					0	
6						0

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000					
2		0					
3			0				
4				0			
5					0		
6						0	

$m[1,1] + p[1{-}1]p[1]p[2] + m[1{+}1,2] {=} 0 {+} 2000 {+} 0 {=} 2000$

Simulação

$p[0]{=}10$ $p[1]{=}10$ $p[2]{=}20$ $p[3]{=}30$ $p[4]{=}10$ $p[5]{=}15$ $p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000					
2		0	??				
3			0				
4				0			
5					0		
6						0	
i							

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000					
2		0	6000				
3			0				
4				0			
5					0		
6						0	

$m[2,2] + p[2-1]p[2]p[3] + m[2+1,3] = 0 + 6000 + 0 = 6000$

Simulação

$p[0]{=}10$ $p[1]{=}10$ $p[2]{=}20$ $p[3]{=}30$ $p[4]{=}10$ $p[5]{=}15$ $p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000					
2		0	6000				
3			0	??			
4				0			
5					0		
6						0	
i							

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000					
2		0	6000				
3			0	6000			
4				0			
5					0		
6						0	

$$m[3,3] + p[3-1]p[3]p[4] + m[3+1,4] = 0 + 6000 + 0 = 6000$$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

i \ j	1	2	3	4	5	6
1	0	2000				
2		0	6000			
3			0	6000		
4				0	??	
5					0	
6						0

Simulação

$p[0]=10 \quad p[1]=10 \quad p[2]=20 \quad p[3]=30 \quad p[4]=10 \quad p[5]=15 \quad p[6]=30$

	1	2	3	4	5	6
1	0	2000				
2		0	6000			
3			0	6000		
4				0	4500	
5					0	
6						0

j

$m[4,4] + p[4-1]p[4]p[5] + m[4+1,5] = 0 + 4500 + 0 = 4500$

Simulação

$p[0]{=}10$ $p[1]{=}10$ $p[2]{=}20$ $p[3]{=}30$ $p[4]{=}10$ $p[5]{=}15$ $p[6]{=}30$

i \ j	1	2	3	4	5	6
1	0	2000				
2		0	6000			
3			0	6000		
4				0	4500	
5					0	??
6						0

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000					
2		0	6000				
3			0	6000			
4				0	4500		
5					0	4500	
6						0	

$m[5,5] + p[5-1]p[5]p[6] + m[5+1,6] = 0 + 4500 + 0 = 4500$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	??				
2		0	6000				
3			0	6000			
4				0	4500		
5					0	4500	
6						0	
i							

Simulação

$p[0]=10 \quad p[1]=10 \quad p[2]=20 \quad p[3]=30 \quad p[4]=10 \quad p[5]=15 \quad p[6]=30$

	1	2	3	4	5	6
1	0	2000	9000			
2		0	6000			
3			0	6000		
4				0	4500	
5					0	4500
6						0

j

$m[1,1] + p[1-1]p[1]p[3] + m[1+1,3] = 0 + 3000 + 6000 = 9000$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000				
2		0	6000				
3			0	6000			
4				0	4500		
5					0	4500	
6						0	

$m[1,2] + p[1-1]p[2]p[3] + m[2+1,3] = 2000 + 6000 + 0 = 8000$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

$i \backslash j$	1	2	3	4	5	6
1	0	2000	8000			
2		0	6000	??		
3			0	6000		
4				0	4500	
5					0	4500
6						0

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6
1	0	2000	8000			
2		0	6000	8000		
3			0	6000		
4				0	4500	
5					0	4500
6						0

j

$$m[2,2] + p[2-1]p[2]p[4] + m[2+1,4] = 0 + 2000 + 6000 = 8000$$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000				
2		0	6000	8000			
3			0	6000			
4				0	4500		
5					0	4500	
6						0	

$m[2,3] + p[2-1]p[3]p[4] + m[3+1,4]{=}6000{+}3000{+}0{=}9000$

Simulação

$p[0]{=}10$ $p[1]{=}10$ $p[2]{=}20$ $p[3]{=}30$ $p[4]{=}10$ $p[5]{=}15$ $p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000				
2		0	6000	8000			
3			0	6000	??		
4				0	4500		
5					0	4500	
6						0	
i							

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000				
2		0	6000	8000			
3			0	6000	13500		
4				0	4500		
5					0	4500	
6						0	

$m[3,3] + p[3-1]p[3]p[5] + m[3+1,5] {=} 0 {+} 9000 {+} 4500 {=} 13500$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000				
2		0	6000	8000			
3			0	6000	9000		
4				0	4500		
5					0	4500	
6						0	

$m[3,4] + p[3-1]p[4]p[5] + m[4+1,5]{=}6000{+}3000{+}0{=}9000$

Simulação

$p[0]{=}10$ $p[1]{=}10$ $p[2]{=}20$ $p[3]{=}30$ $p[4]{=}10$ $p[5]{=}15$ $p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000				
2		0	6000	8000			
3			0	6000	9000		
4				0	4500	??	
5					0	4500	
6						0	
i							

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000				
2		0	6000	8000			
3			0	6000	9000		
4				0	4500	13500	
5					0	4500	
6						0	

$$m[4,4] + p[4{-}1]p[4]p[6] + m[4{+}1,6]{=}0{+}9000{+}4500{=}13500$$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000				
2		0	6000	8000			
3			0	6000	9000		
4				0	4500	13500	
5					0	4500	
6						0	

$m[4,5] + p[4-1]p[5]p[6] + m[5+1,6] = 4500 + 13500 + 0 = 18000$

Simulação

$p[0]{=}10$ $p[1]{=}10$ $p[2]{=}20$ $p[3]{=}30$ $p[4]{=}10$ $p[5]{=}15$ $p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	??			
2		0	6000	8000			
3			0	6000	9000		
4				0	4500	13500	
5					0	4500	
6						0	
i							

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000			
2		0	6000	8000			
3			0	6000	9000		
4				0	4500	13500	
5					0	4500	
6						0	

$$m[1,1] + p[1-1]p[1]p[4] + m[1+1,4] = 0 + 1000 + 8000 = 9000$$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000			
2		0	6000	8000			
3			0	6000	9000		
4				0	4500	13500	
5					0	4500	
6						0	

$$m[1,2] + p[1-1]p[2]p[4] + m[2+1,4] = 2000 + 2000 + 6000 = 10000$$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6 j
1	0	2000	8000	9000		
2		0	6000	8000		
3			0	6000	9000	
4				0	4500	13500
5					0	4500
6						0

$m[1,3] + p[1{-}1]p[3]p[4] + m[3{+}1,4] {=} 8000{+}3000{+}0{=}11000$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

i \ j	1	2	3	4	5	6
1	0	2000	8000	9000		
2		0	6000	8000	??	
3			0	6000	9000	
4				0	4500	13500
5					0	4500
6						0

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000			
2		0	6000	8000	12000		
3			0	6000	9000		
4				0	4500	13500	
5					0	4500	
6						0	

$m[2,2] + p[2-1]p[2]p[5] + m[2+1,5] = 0 + 3000 + 9000 = 12000$

Simulação

$p[0]=10 \quad p[1]=10 \quad p[2]=20 \quad p[3]=30 \quad p[4]=10 \quad p[5]=15 \quad p[6]=30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000			
2		0	6000	8000	12000		
3			0	6000	9000		
4				0	4500	13500	
5					0	4500	
6						0	

$m[2,3] + p[2-1]p[3]p[5] + m[3+1,5] = 6000 + 4500 + 4500 = 15000$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000			
2		0	6000	8000	9500		
3			0	6000	9000		
4				0	4500	13500	
5					0	4500	
6						0	

$$m[2,4] + p[2-1]p[4]p[5] + m[4+1,5] = 8000 + 1500 + 0 = 9500$$

Simulação

$p[0]=10$ $p[1]=10$ $p[2]=20$ $p[3]=30$ $p[4]=10$ $p[5]=15$ $p[6]=30$

i \ j	1	2	3	4	5	6
1	0	2000	8000	9000		
2		0	6000	8000	9500	
3			0	6000	9000	??
4				0	4500	13500
5					0	4500
6						0

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000			
2		0	6000	8000	9500		
3			0	6000	9000	31500	
4				0	4500	13500	
5					0	4500	
6						0	

$m[3,3] + p[3{-}1]p[3]p[6] + m[3{+}1,6]{=}0{+}18000{+}13500{=}31500$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000			
2		0	6000	8000	9500		
3			0	6000	9000	16500	
4				0	4500	13500	
5					0	4500	
6						0	

$m[3,4] + p[3{-}1]p[4]p[6] + m[4{+}1,6]{=}6000{+}6000{+}4500{=}16500$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000			
2		0	6000	8000	9500		
3			0	6000	9000	16500	
4				0	4500	13500	
5					0	4500	
6						0	

$m[3,5] + p[3-1]p[5]p[6] + m[5+1,6] = 9000 + 9000 + 0 = 18000$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

i \ j	1	2	3	4	5	6
1	0	2000	8000	9000	??	
2		0	6000	8000	9500	
3			0	6000	9000	16500
4				0	4500	13500
5					0	4500
6						0

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000	11000		
2		0	6000	8000	9500		
3			0	6000	9000	16500	
4				0	4500	13500	
5					0	4500	
6						0	

$m[1,1] + p[1{-}1]p[1]p[5] + m[1{+}1,5] {=} 0 {+} 1500 {+} 9500 {=} 11000$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000	11000		
2		0	6000	8000	9500		
3			0	6000	9000	16500	
4				0	4500	13500	
5					0	4500	
6						0	

$m[1,2] + p[1-1]p[2]p[5] + m[2+1,5] = 2000 + 3000 + 9000 = 14000$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000	11000		
2		0	6000	8000	9500		
3			0	6000	9000	16500	
4				0	4500	13500	
5					0	4500	
6						0	

$m[1,3] + p[1-1]p[3]p[5] + m[3+1,5] = 8000 + 4500 + 4500 = 17000$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000	10500		
2		0	6000	8000	9500		
3			0	6000	9000	16500	
4				0	4500	13500	
5					0	4500	
6						0	

$$m[1,4] + p[1-1]p[4]p[5] + m[4+1,5] = 9000 + 1500 + 0 = 10500$$

Simulação

$p[0]{=}10$ $p[1]{=}10$ $p[2]{=}20$ $p[3]{=}30$ $p[4]{=}10$ $p[5]{=}15$ $p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000	10500		
2		0	6000	8000	9500	??	
3			0	6000	9000	16500	
4				0	4500	13500	
5					0	4500	
6						0	
i							

Simulação

$p[0]=10 \quad p[1]=10 \quad p[2]=20 \quad p[3]=30 \quad p[4]=10 \quad p[5]=15 \quad p[6]=30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000	10500		
2		0	6000	8000	9500	22500	
3			0	6000	9000	16500	
4				0	4500	13500	
5					0	4500	
6						0	

$m[2,2] + p[2-1]p[2]p[6] + m[2+1,6] = 0 + 6000 + 16500 = 22500$

Simulação

$p[0]{=}10$ $p[1]{=}10$ $p[2]{=}20$ $p[3]{=}30$ $p[4]{=}10$ $p[5]{=}15$ $p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000	10500		
2		0	6000	8000	9500	22500	
3			0	6000	9000	16500	
4				0	4500	13500	
5					0	4500	
6						0	

$m[2,3] + p[2{-}1]p[3]p[6] + m[3{+}1,6]{=}6000{+}9000{+}13500{=}28500$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000	10500		
2		0	6000	8000	9500	15500	
3			0	6000	9000	16500	
4				0	4500	13500	
5					0	4500	
6						0	

$$m[2,4] + p[2-1]p[4]p[6] + m[4+1,6] = 8000+3000+4500=15500$$

Simulação

$p[0]=10$ $p[1]=10$ $p[2]=20$ $p[3]=30$ $p[4]=10$ $p[5]=15$ $p[6]=30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000	10500		
2		0	6000	8000	9500	14000	
3			0	6000	9000	16500	
4				0	4500	13500	
5					0	4500	
6						0	

$m[2,5] + p[2-1]p[5]p[6] + m[5+1,6] = 9500 + 4500 + 0 = 14000$

Simulação

$p[0]{=}10$ $p[1]{=}10$ $p[2]{=}20$ $p[3]{=}30$ $p[4]{=}10$ $p[5]{=}15$ $p[6]{=}30$

i \ j	1	2	3	4	5	6
1	0	2000	8000	9000	10500	??
2		0	6000	8000	9500	14000
3			0	6000	9000	16500
4				0	4500	13500
5					0	4500
6						0

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000	10500	17000	
2		0	6000	8000	9500	14000	
3			0	6000	9000	16500	
4				0	4500	13500	
5					0	4500	
6						0	

$m[1,1] + p[1{-}1]p[1]p[6] + m[1{+}1,6]{=}0{+}3000{+}14000{=}17000$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000	10500	17000	
2		0	6000	8000	9500	14000	
3			0	6000	9000	16500	
4				0	4500	13500	
5					0	4500	
6						0	

$m[1,2] + p[1{-}1]p[2]p[6] + m[2{+}1,6]{=}2000{+}6000{+}16500{=}24500$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000	10500	17000	
2		0	6000	8000	9500	14000	
3			0	6000	9000	16500	
4				0	4500	13500	
5					0	4500	
6						0	

$m[1,3] + p[1-1]p[3]p[6] + m[3+1,6] = 8000+9000+13500 = 30500$

Simulação

$p[0]{=}10$ $p[1]{=}10$ $p[2]{=}20$ $p[3]{=}30$ $p[4]{=}10$ $p[5]{=}15$ $p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000	10500	16500	
2		0	6000	8000	9500	14000	
3			0	6000	9000	16500	
4				0	4500	13500	
5					0	4500	
6						0	

$m[1,4] + p[1{-}1]p[4]p[6] + m[4{+}1,6]{=}9000{+}3000{+}4500{=}16500$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

	1	2	3	4	5	6	j
1	0	2000	8000	9000	10500	15000	
2		0	6000	8000	9500	14000	
3			0	6000	9000	16500	
4				0	4500	13500	
5					0	4500	
6						0	

$m[1,5] + p[1-1]p[5]p[6] + m[5+1,6]{=}10500{+}4500{+}0{=}15000$

Simulação

$p[0]{=}10 \quad p[1]{=}10 \quad p[2]{=}20 \quad p[3]{=}30 \quad p[4]{=}10 \quad p[5]{=}15 \quad p[6]{=}30$

i \ j	1	2	3	4	5	6
1	0	2000	8000	9000	10500	15000
2		0	6000	8000	9500	14000
3			0	6000	9000	16500
4				0	4500	13500
5					0	4500
6						0

Algoritmo de programação dinâmica

Recebe $p[0\,..\,n]$ e devolve $m[1,n]$.

MATRIX-CHAIN-ORDER (p, n)

para $i \leftarrow 1$ **até** n **faça**
    $m[i,i] \leftarrow 0$
para $\ell \leftarrow 2$ **até** n **faça**
    para $i \leftarrow 1$ **até** $n - \ell + 1$ **faça**
        $j \leftarrow i + \ell - 1$
        $m[i,j] \leftarrow \infty$
        para $k \leftarrow i$ **até** $j - 1$ **faça**
            $q \leftarrow m[i,k] + p[i-1]p[k]p[j] + m[k{+}1,j]$
            se $q < m[i,j]$
                então $m[i,j] \leftarrow q$
devolva $m[1,n]$

Correção e consumo de tempo

Linhas 3–10: tratam subcadeias $A_i \cdots A_j$ de comprimento ℓ

Correção e consumo de tempo

Linhas 3–10: tratam subcadeias $A_i \cdots A_j$ de comprimento ℓ

Consumo de tempo: ???

Correção e consumo de tempo

Linhas 3–10: tratam subcadeias $A_i \cdots A_j$ de comprimento ℓ

Consumo de tempo: $\mathrm{O}(n^3)$ (três loops encaixados)

Correção e consumo de tempo

Linhas 3–10: tratam subcadeias $A_i \cdots A_j$ de comprimento ℓ

Consumo de tempo: $\mathrm{O}(n^3)$ (três loops encaixados)

Curioso verificar que consumo de tempo é $\Omega(n^3)$:

Número de execuções da linha 8:

Correção e consumo de tempo

Linhas 3–10: tratam subcadeias $A_i \cdots A_j$ de comprimento ℓ

Consumo de tempo: $\mathrm{O}(n^3)$ (três loops encaixados)

Curioso verificar que consumo de tempo é $\Omega(n^3)$:

Número de execuções da linha 8:

ℓ	i	execs linha 8
2	$1,\ldots,n-1$	$(n-1)\cdot 1$
3	$1,\ldots,n-2$	$(n-2)\cdot 2$
4	$1,\ldots,n-3$	$(n-3)\cdot 3$
$\ldots$	$\ldots$	$\ldots$
$n-1$	$1,2$	$2\cdot(n-2)$
n	1	$1\cdot(n-1)$
	total	$\sum_{h=1}^{n-1} h(n-h)$

Consumo de tempo

Para $n \geq 6$, $\sum_{h=1}^{n-1} h(n-h) =$

$$
\begin{aligned}
&= n\sum_{h=1}^{n-1} h - \sum_{h=1}^{n-1} h^2 \\
&= n\tfrac{1}{2}n(n-1) - \tfrac{1}{6}(n-1)n(2n-1) \quad \text{(CLRS p.1060)} \\
&\geq \tfrac{1}{2}n^2(n-1) - \tfrac{1}{6}2n^3 \\
&\geq \tfrac{1}{2}n^2\tfrac{5n}{6} - \tfrac{1}{3}n^3 \\
&= \tfrac{5}{12}n^3 - \tfrac{1}{3}n^3 \\
&= \tfrac{1}{12}n^3.
\end{aligned}
$$

Consumo de tempo é $\Omega(n^3)$

Conclusão

O consumo de tempo do algoritmo MATRIX-CHAIN-ORDER é $\Theta(n^3)$.

Versão recursiva eficiente

MEMOIZED-MATRIX-CHAIN-ORDER (p, n)

para $i \leftarrow 1$ **até** n **faça**
    para $j \leftarrow 1$ **até** n **faça**
        $m[i, j] \leftarrow \infty$
devolva LOOKUP-CHAIN $(p, 1, n)$

Versão recursiva eficiente

LOOKUP-CHAIN (p, i, j)

se $m[i, j] < \infty$
    então devolva $m[i, j]$
se $i = j$
    então $m[i, j] \leftarrow 0$
    senão para $k \leftarrow i$ **até** $j - 1$ **faça**
            $q \leftarrow$ LOOKUP-CHAIN (p, i, k)
                $+\ p[i-1]p[k]p[j]$
                $+$ LOOKUP-CHAIN $(p, k+1, j)$
            se $q < m[i, j]$
                então $m[i, j] \leftarrow q$
devolva $m[i, j]$

Ingredientes de programação dinâmica

- Subestrutura ótima: soluções ótimas contém soluções ótimas de subproblemas.
- Subestrutura: decomponha o problema em subproblemas menores e, com sorte, mais simples.
- *Bottom-up*: combine as soluções dos problemas menores para obter soluções dos maiores.
- Tabela: armazene as soluções dos subproblemas em uma tabela, pois soluções dos subproblemas são consultadas várias vezes.
- Número de subproblemas: para a eficiência do algoritmo é importante que o número de subproblemas resolvidos seja ‘pequeno’.
- Memoized: versão *top-down*, recursão com tabela.

Exercício

O algoritmo MATRIX-CHAIN-ORDER determina o número mínimo de multiplicações escalares necessário para calcular produto $A_1 A_2 \cdots A_n$.

Na aula, mencionamos uma maneira de obter uma parentização ótima a partir dos cálculos feitos, usando para isso um dado a mais que podemos guardar no decorrer do algoritmo.

Faça os ajustes sugeridos na aula, de modo a guardar esse dado extra, e devolvê-lo junto com o valor $m[1, n]$.

Faça uma rotina que recebe a informação extra armazenada pelo algoritmo acima e imprime uma parentização ótima das matrizes $A_1 A_2 \cdots A_n$.

3. Algoritmo p/ Valor Ótimo (Bottom-up, não recursivo)

MCM-PD(p, n)

Criar matriz $m[0 \ldots n,\ 0 \ldots n]$
para $i \leftarrow 1$ **até** n: $\quad m[i,i] \leftarrow 0$
para $\ell \leftarrow 2$ **até** n:
    para $i \leftarrow 1$ **até** $n - \ell + 1$: $\quad j \leftarrow i + \ell - 1$
        $m[i,j] \leftarrow \infty$
        para $k \leftarrow i$ **até** $j - 1$
            $q \leftarrow m[i,k] + m[k+1,j] + p_{i-1} \cdot p_k \cdot p_j$
            se $(m[i,j] > q)$ **então**
                $m[i,j] \leftarrow q$
retorne $m[1,n]$

Tempo $\Theta(n^3)$

3. Algoritmo p/ Valor Ótimo (Bottom-up, não recursivo)

MCM-PD(p, n)

Criar matriz $m[0 \dots n,\ 0 \dots n]$
para $i \leftarrow 1$ **até** n: $\quad m[i, i] \leftarrow 0$
para $i \leftarrow n - 1$ **até** 1 **(dec):**
    para $j \leftarrow i + 1$ **até** n:
        $m[i, j] \leftarrow \infty$
        para $k \leftarrow i$ **até** $j - 1$
            $q \leftarrow m[i, k] + m[k + 1, j] + p_{i-1} \cdot p_k \cdot p_j$
            se $(m[i, j] > q)$ **então**
                $m[i, j] \leftarrow q$
retorne $m[1, n]$

Tempo $\Theta(n^3)$

3. Algoritmo p/ Valor Ótimo (Bottom-up, não recursivo)

MCM-PD(p, n)

Criar matriz $m[0 \dots n,\ 0 \dots n]$
para $i \leftarrow 1$ **até** n: $\quad m[i,i] \leftarrow 0$
para $j \leftarrow 2$ **até** n**:**
    para $i \leftarrow j-1$ **até** 1 **(dec):**
        $m[i,j] \leftarrow \infty$
        para $k \leftarrow i$ **até** $j-1$
            $q \leftarrow m[i,k] + m[k+1,j] + p_{i-1} \cdot p_k \cdot p_j$
            se $(m[i,j] > q)$ **então**
                $m[i,j] \leftarrow q$
retorne $m[1,n]$

Tempo $\Theta(n^3)$

4. Algoritmo p/ obter uma Solução Ótima (Bottom-up)

MCM-PD(p, n)

Criar matriz $m[0 \ldots n,\ 0 \ldots n]$
para $i \leftarrow 1$ **até** n: $m[i,i] \leftarrow 0$; $R[i,i] \leftarrow i$
para $i \leftarrow n-1$ **até** 1 (dec):
    para $j \leftarrow i+1$ **até** n:
        $m[i,j] \leftarrow \infty$
        para $k \leftarrow i$ **até** $j-1$
            $q \leftarrow m[i,k] + m[k+1,j] + p_{i-1} \cdot p_k \cdot p_j$
            se $(m[i,j] > q)$ **então**
                $m[i,j] \leftarrow q$; $R[i,j] \leftarrow k$
retorne $m[1,n]$ **e** R

Tempo $\Theta(n^3)$

4b. Algoritmo p/ escrever a Solução Ótima (recursivo)

Print-OPT(i, j, R)

se $i = j$ **então print** "A"$_i$; **retorne**
$k \leftarrow R[i, j]$
print "("
Print-OPT$(\ i\ ,\ k,\ R)$
Print-OPT$(k{+}1,\ j,\ R)$
print ")"

Exemplo: $((A_1(A_2A_3))((A_4A_5)A_6))$

Tempo $\Theta(n)$

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	∞	∞	∞	∞	∞
10×3	-	0	∞	∞	∞	∞
3×12	-	-	0	∞	∞	∞
12×5	-	-	-	0	∞	∞
5×50	-	-	-	-	0	∞
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	∞	∞	∞	∞	∞
10×3	-	0	∞	∞	∞	∞
3×12	-	-	0	∞	∞	∞
12×5	-	-	-	0	∞	∞
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	∞	∞	∞	∞	∞
10×3	-	0	∞	∞	∞	∞
3×12	-	-	0	∞	∞	∞
12×5	-	-	-	0	3000 [4]	∞
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	∞	∞	∞	∞	∞
10×3	-	0	∞	∞	∞	∞
3×12	-	-	0	∞	∞	∞
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	∞	∞	∞	∞	∞
10×3	-	0	∞	∞	∞	∞
3×12	-	-	0	180 [3]	∞	∞
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	∞	∞	∞	∞	∞
10×3	-	0	∞	∞	∞	∞
3×12	-	-	0	180 [3]	930 [4]	∞
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	∞	∞	∞	∞	∞
10×3	-	0	∞	∞	∞	∞
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	∞	∞	∞	∞	∞
10×3	-	0	360 [2]	∞	∞	∞
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	∞	∞	∞	∞	∞
10×3	-	0	360 [2]	330 [2]	∞	∞
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	∞	∞	∞	∞	∞
10×3	-	0	360 [2]	330 [2]	2830 [4]	∞
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	∞	∞	∞	∞	∞
10×3	-	0	360 [2]	330 [2]	2830 [4]	1950 [2]
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	∞	∞	∞	∞
10×3	-	0	360 [2]	330 [2]	2830 [4]	1950 [2]
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	∞	∞	∞
10×3	-	0	360 [2]	330 [2]	2830 [4]	1950 [2]
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	∞	∞
10×3	-	0	360 [2]	330 [2]	2830 [4]	1950 [2]
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	1655 [4]	∞
10×3	-	0	360 [2]	330 [2]	2830 [4]	1950 [2]
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	1655 [4]	2010 [2]
10×3	-	0	360 [2]	330 [2]	2830 [4]	1950 [2]
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	∞	∞	∞	∞	∞
10×3	-	0	∞	∞	∞	∞
3×12	-	-	0	∞	∞	∞
12×5	-	-	-	0	∞	∞
5×50	-	-	-	-	0	∞
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	∞	∞	∞	∞
10×3	-	0	360 [2]	∞	∞	∞
3×12	-	-	0	180 [3]	∞	∞
12×5	-	-	-	0	3000 [4]	∞
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	∞	∞	∞
10×3	-	0	360 [2]	330 [2]	∞	∞
3×12	-	-	0	180 [3]	930 [4]	∞
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	∞	∞
10×3	-	0	360 [2]	330 [2]	2830 [4]	∞
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	1655 [4]	∞
10×3	-	0	360 [2]	330 [2]	2830 [4]	1950 [2]
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	1655 [4]	2010 [2]
10×3	-	0	360 [2]	330 [2]	2830 [4]	1950 [2]
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	∞	∞	∞	∞	∞
10×3	-	0	∞	∞	∞	∞
3×12	-	-	0	∞	∞	∞
12×5	-	-	-	0	∞	∞
5×50	-	-	-	-	0	∞
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	∞	∞	∞	∞
10×3	-	0	∞	∞	∞	∞
3×12	-	-	0	∞	∞	∞
12×5	-	-	-	0	∞	∞
5×50	-	-	-	-	0	∞
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	∞	∞	∞	∞
10×3	-	0	360 [2]	∞	∞	∞
3×12	-	-	0	∞	∞	∞
12×5	-	-	-	0	∞	∞
5×50	-	-	-	-	0	∞
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	∞	∞	∞
10×3	-	0	360 [2]	∞	∞	∞
3×12	-	-	0	∞	∞	∞
12×5	-	-	-	0	∞	∞
5×50	-	-	-	-	0	∞
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	∞	∞	∞
10×3	-	0	360 [2]	∞	∞	∞
3×12	-	-	0	180 [3]	∞	∞
12×5	-	-	-	0	∞	∞
5×50	-	-	-	-	0	∞
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	∞	∞	∞
10×3	-	0	360 [2]	330 [2]	∞	∞
3×12	-	-	0	180 [3]	∞	∞
12×5	-	-	-	0	∞	∞
5×50	-	-	-	-	0	∞
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	∞	∞
10×3	-	0	360 [2]	330 [2]	∞	∞
3×12	-	-	0	180 [3]	∞	∞
12×5	-	-	-	0	∞	∞
5×50	-	-	-	-	0	∞
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	∞	∞
10×3	-	0	360 [2]	330 [2]	∞	∞
3×12	-	-	0	180 [3]	∞	∞
12×5	-	-	-	0	3000 [4]	∞
5×50	-	-	-	-	0	∞
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	∞	∞
10×3	-	0	360 [2]	330 [2]	∞	∞
3×12	-	-	0	180 [3]	930 [4]	∞
12×5	-	-	-	0	3000 [4]	∞
5×50	-	-	-	-	0	∞
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	∞	∞
10×3	-	0	360 [2]	330 [2]	2830 [4]	∞
3×12	-	-	0	180 [3]	930 [4]	∞
12×5	-	-	-	0	3000 [4]	∞
5×50	-	-	-	-	0	∞
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	1655 [4]	∞
10×3	-	0	360 [2]	330 [2]	2830 [4]	∞
3×12	-	-	0	180 [3]	930 [4]	∞
12×5	-	-	-	0	3000 [4]	∞
5×50	-	-	-	-	0	∞
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	1655 [4]	∞
10×3	-	0	360 [2]	330 [2]	2830 [4]	∞
3×12	-	-	0	180 [3]	930 [4]	∞
12×5	-	-	-	0	3000 [4]	∞
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	1655 [4]	∞
10×3	-	0	360 [2]	330 [2]	2830 [4]	∞
3×12	-	-	0	180 [3]	930 [4]	∞
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	1655 [4]	∞
10×3	-	0	360 [2]	330 [2]	2830 [4]	∞
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	1655 [4]	∞
10×3	-	0	360 [2]	330 [2]	2830 [4]	1950 [2]
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	1655 [4]	2010 [2]
10×3	-	0	360 [2]	330 [2]	2830 [4]	1950 [2]
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Exercício CLRS 15.2-1

Encontre a maneira ótima de fazer a multiplicação iterada das seis matrizes cujas dimensões são (5, 10, 3, 12, 5, 50, 6).

Solução:

.	5×10	10×3	3×12	12×5	5×50	50×6
5×10	0	150 [1]	330 [2]	405 [2]	1655 [4]	2010 [2]
10×3	-	0	360 [2]	330 [2]	2830 [4]	1950 [2]
3×12	-	-	0	180 [3]	930 [4]	1770 [4]
12×5	-	-	-	0	3000 [4]	1860 [4]
5×50	-	-	-	-	0	1500 [5]
50×6	-	-	-	-	-	0

Solução: $A_1 \times \ldots \times A_6 \quad = \quad (\ (A_1 A_2) \times ((A_3 A_4)(A_5 A_6))\ )$

Obs: $A_1 \times \ldots \times A_5 \quad = \quad (\ ((A_1 A_2)(A_3 A_4)) \times A_5\ )$

Exercício CLRS 15.3-4

Considere o seguinte algoritmo para determinar a melhor ordem de multiplicação de matrizes $A_1 \times A_2 \times \ldots \times A_n$ de dimensões $p_0, p_1, \ldots, p_n$: primeiro escolha k que minimize p_k; depois determine recursivamente as ordens de multiplicação de $A_1 \times \ldots \times A_k$ e $A_{k+1} \times \ldots \times A_n$. Esse algoritmo produz uma ordem que minimiza o número total de multiplicações escalares? E se k for escolhido de modo a maximizar p_k?

Solução:

- Contraexemplo Mínimo p_k: $A_1(1 \times 1)$, $A_2(1 \times 2)$ e $A_3(2 \times 100)$.
- $(A_1(A_2A_3))$: 100+200=300 multiplicações escalares (min p_k)
- $((A_1A_2)A_3)$: 2+200=202 multiplicações escalares (OPT)

- Contraexemplo Máximo p_k: $A_1(1 \times 100)$, $A_2(100 \times 1)$ e $A_3(1 \times 1)$.
- $(A_1(A_2A_3))$: 100+100=200 multiplicações escalares (max p_k)
- $((A_1A_2)A_3)$: 100+1=101 multiplicações escalares (OPT)

Exercício CLRS 15.2-6

Mostre que são necessários exatamente $n-1$ pares de parênteses para especificar exatamente a ordem de multiplicação de $A_1 \times \ldots \times A_n$

Solução: Indução

- **Caso base:** $n = 1$: Uma matriz A_1, zero par de parêntesis. OK
- **Caso base:** $n = 2$: Duas matrizes $(A_1 A_2)$, um par de parêntesis. OK
- **H.I.:** Fixe $n > 2$ e suponha valer para todo n^o $< n$ de matrizes.
- **P.I.:** Vamos provar que vale para n matrizes.
- Toda multiplicação de $A_1 \times \ldots \times A_n$ será da forma:
- $(\; A_1 \ldots A_k \;\; \times \;\; A_{k+1} \ldots A_n)$
- Pela H.I., a 1^o e a 2^o parte usam respectivamente $k-1$ e $n-k-1$ pares de parêntesis.
- $Total = (k-1) + (n-k-1) + 1 \;\; = \;\; n-1$ pares de parêntesis.

Exercícios

Exercício 13.A [CLRS 15.2-1]
Encontre a maneira ótima de fazer a multiplicação iterada das matrizes cujas dimensões são $(5, 10, 3, 12, 5, 50, 6)$.

Exercício 13.B [CLRS 15.2-5]
Mostre que são necessários exatamente $n - 1$ pares de parênteses para especificar exatamente a ordem de multiplicação de $A_1 \cdot A_2 \cdots A_n$.

Exercício 13.C [CLRS 15.3-2]
Desenhe a árvore de recursão para o algoritmo Merge-Sort aplicado a um vetor de 16 elementos. Por que a técnica de programação dinâmica não é capaz de acelerar o algoritmo?

Mais exercícios

Exercício 13.D [CLRS 15.3-5 expandido]
Considere o seguinte algoritmo para determinar a ordem de multiplicação de uma cadeia de matrizes $A_1, A_2, \ldots, A_n$ de dimensões $p_0, p_1, \ldots, p_n$: primeiro, escolha k que minimize p_k; depois, determine recursivamente as ordens de multiplicação de $A_1, \ldots, A_k$ e $A_{k+1}, \ldots, A_n$. Esse algoritmo produz uma ordem que minimiza o número total de multiplicações escalares? E se k for escolhido de modo a maximizar p_k? E se k for escolhido de modo a minimizar p_k?

Exercício 13.E
Prove que o número de execuções da linha 9 em MATRIX-CHAIN-ORDER é $\mathrm{O}(n^3)$.