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 $p \times r$.

Multiplicação iterada de matrizes

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

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

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)

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

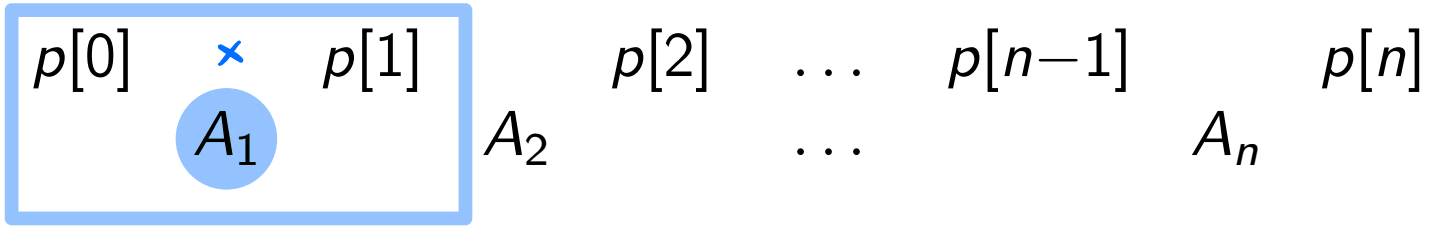
Número de multiplicações escalares $= p \cdot q \cdot r$

Multiplicação iterada

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

Multiplicação iterada

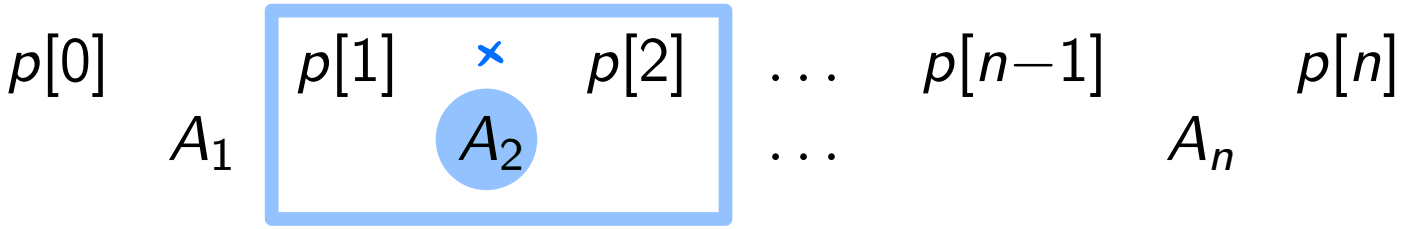
Problema: Encontrar número mínimo de multiplicações escalares necessário para calcular produto $A_1 A_2 \cdots A_n$.



cada A_i é $p[i-1] \times p[i]$, ou seja, $A_i[1\,..\,p[i-1], 1\,..\,p[i]]$

Multiplicação iterada

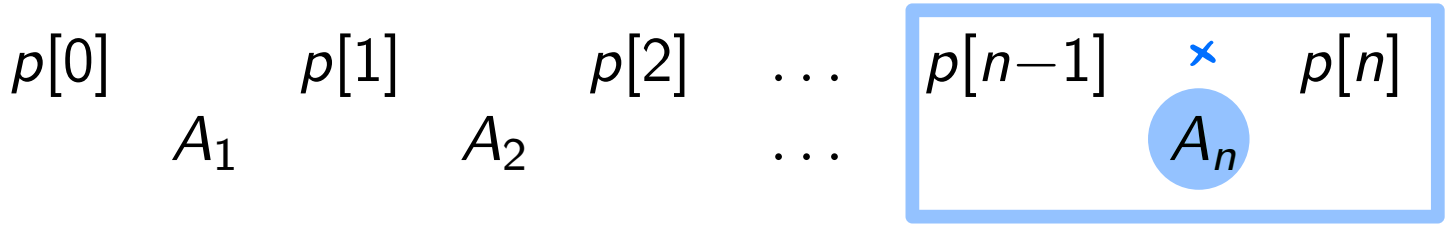
Problema: Encontrar número mínimo de multiplicações escalares necessário para calcular produto $A_1 A_2 \cdots A_n$.



cada A_i é $p[i-1] \times p[i]$, ou seja, $A_i[1\,..\,p[i-1], 1\,..\,p[i]]$

Multiplicação iterada

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



cada A_i é $p[i-1] \times p[i]$, ou seja, $A_i[1\,..\,p[i-1], 1\,..\,p[i]]$

Multiplicação iterada

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

$$p[0] \quad A_1 \quad p[1] \quad A_2 \quad p[2] \quad \ldots \quad p[n-1] \quad A_n \quad p[n]$$

cada A_i é $p[i-1] \times p[i]$, ou seja, $A_i[1\,..\,p[i-1], 1\,..\,p[i]]$

Exemplo: $A_1 \cdot A_2 \cdot A_3$

$$p[0]{=}10 \quad A_1 \quad p[1]{=}100 \quad A_2 \quad p[2]{=}5 \quad A_3 \quad p[3]{=}50$$

Multiplicação iterada

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

$$\begin{array}{ccccccccc} p_0 & & p_1 & & p_2 & \cdots & p_{n-1} & & p_n \\ & A_1 & & A_2 & & \cdots & & A_n & \end{array}$$

cada A_i é $p_{i-1} \times p_i$, ou seja, $A_i[1\,..\,p_{i-1}, 1\,..\,p_i]$

Exemplo: $A_1 \cdot A_2 \cdot A_3$

$$\begin{array}{ccccccccc} p_0{=}10 & & p_1{=}100 & & p_2{=}5 & & p_3{=}50 \\ & A_1 & & A_2 & & A_3 & \end{array}$$

Multiplicação iterada

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

$$\begin{array}{ccccccccc} p_0 & & p_1 & & p_2 & \cdots & p_{n-1} & & p_n \\ & A_1 & & A_2 & & \cdots & & A_n & \end{array}$$

cada A_i é $p_{i-1} \times p_i$, ou seja, $A_i[1\,..\,p_{i-1}, 1\,..\,p_i]$

Exemplo: $A_1 \cdot A_2 \cdot A_3$

$p_0{=}10$ A_1 $p_1{=}100$ A_2 $p_2{=}5$ A_3 $p_3{=}50$

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

$+\ 5000$

2500

Multiplicação iterada

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

$$\begin{array}{ccccccccc} p_0 & & p_1 & & p_2 & \cdots & p_{n-1} & & p_n \\ & A_1 & & A_2 & & \cdots & & A_n & \end{array}$$

cada A_i é $p_{i-1} \times p_i$, ou seja, $A_i[1\,..\,p_{i-1}, 1\,..\,p_i]$

Exemplo: $A_1 \cdot A_2 \cdot A_3$

$p_0{=}10 \quad A_1 \quad p_1{=}100 \quad A_2 \quad p_2{=}5 \quad A_3 \quad p_3{=}50$

$((A_1\ A_2)\ A_3)$	7500	multiplicações escalares
$(A_1\ (A_2\ A_3))$	75000	multiplicações escalares
	+ 25000	
	50000	

Soluções ótimas contêm soluções ótimas

Se $$((A_1A_2)A_3)(A_4((A_5A_6)A_7))$$

é ordem ótima de multiplicação, então

$$((A_1A_2)A_3) \quad \text{e} \quad (A_4((A_5A_6)A_7))$$

também são ordens ótimas.

Soluções ótimas contêm soluções ótimas

Se $$((A_1A_2)A_3)(A_4((A_5A_6)A_7))$$

é ordem ótima de multiplicação, então

$$((A_1A_2)A_3) \quad \text{e} \quad (A_4((A_5A_6)A_7))$$

também são ordens ótimas.

Argumento informal: Suponha por contradição que exista uma ordem melhor para $A_1 \cdot A_2 \cdot A_3$, digamos, $(A_1(A_2A_3))$.

Então $(A_1(A_2A_3))(A_4((A_5A_6)A_7))$ seria uma ordem para $A_1A_2 \cdots A_7$ "melhor que a ótima".

Soluções ótimas contêm soluções ótimas

Se $$((A_1A_2)A_3)(A_4((A_5A_6)A_7))$$

é ordem ótima de multiplicação, então

$$((A_1A_2)A_3) \quad \text{e} \quad (A_4((A_5A_6)A_7))$$

também são ordens ótimas.

Argumento "menos" informal: Suponha que

- $((A_1A_2)A_3)$ requer $m_{1,3}$ multiplicações,
- $(A_4((A_5A_6)A_7))$ requer $m_{4,7}$ multiplicações;

assim, a ordem ótima requer $m_{1,3} + p_0p_3p_7 + m_{4,7}$ multiplicações.

Se existe uma ordem $(A_4 \cdots A_7)$ com $m'_{4,7} < m_{4,7}$ multiplicações, então

$$((A_1A_2)A_3)(A_4 \cdots A_7)$$

requer $m_{1,3} + p_0p_3p_7 + m'_{4,7} < m_{1,3} + p_0p_3p_7 + m_{4,7}$ multiplicações, uma contradição.

Soluções ótimas contêm soluções ótimas

Se

$$((A_1A_2)A_3)(A_4((A_5A_6)A_7))$$

é ordem ótima de multiplicação então

$$((A_1A_2)A_3) \quad \text{e} \quad (A_4((A_5A_6)A_7))$$

também são ordens ótimas.

Soluções ótimas contêm soluções ótimas

Se

$$((A_1A_2)A_3)\,(A_4((A_5A_6)A_7))$$

é ordem ótima de multiplicação então

$$((A_1A_2)A_3) \quad \text{e} \quad (A_4((A_5A_6)A_7))$$

também são ordens ótimas.

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

$m[i,j] =$ nº mín. de multiplicações escalares p/ calcular $A_i \cdots A_j$

Recorrência

$m[i, j] =$ nº mín. de multiplicações escalares p/ calcular $A_i \cdots A_j$

se $i = j$ então $m[i, j] = 0$

Recorrência

p/ calcular $(A_i \cdots A_k)(A_{k+1} \cdots A_j)$

$m[i,j] = $ nº mín. de multiplicações escalares p/ calcular $A_i \cdots A_j$

se $i = j$ então $m[i,j] = 0$

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

p/ calcular $A_i \cdots A_k$

p/ calcular $A_{k+1} \cdots A_j$

Decomposição:

p_{i-1} $\cdots$ p_k $\cdots$ p_j

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

Recorrência

$m[i,j] =$ nº mín. de multiplicações escalares p/ calcular $A_i \cdots A_j$

se $i = j$ então $m[i,j] = 0$

se $i < j$ então $m[i,j] = \min\limits_{i \le k < j} \{\, m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \,\}$

Exemplo:

$$m[3,7] = \min_{3 \le k < 7} \{\, m[3,k] + p_2 p_k p_7 + m[k+1,7] \,\}$$

Prova de corretude da recorrência

$$\begin{array}{ccccccccc} p_0 & & p_1 & & p_2 & \cdots & p_{n-1} & & p_n \\ & A_1 & & A_2 & & \cdots & & A_n & \end{array}$$

$m[i,j] =$ nº mín. de multiplicações escalares p/ calcular $A_i \cdots A_j$

se $i = j$ então $m[i,j] = 0$

Trivialmente, se $i = j$, a expressão $A_i \cdots A_j$ é o produto formado por uma única matriz, A_i, que requer 0 multiplicações escalares.

Prova de corretude da recorrência

$$\begin{array}{ccccccccc} p_0 & & p_1 & & p_2 & \cdots & p_{n-1} & & p_n \\ & A_1 & & A_2 & & \cdots & & A_n & \end{array}$$

$m[i,j] =$ nº mín. de multiplicações escalares p/ calcular $A_i \cdots A_j$

Se $i < j$ então $m[i,j] \leq \min_{i \leq k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$:

Suponha que $k \in [i,j)$ atinge o 'min'.

Existe uma ordem para $(A_i \cdots A_k)$, que é uma matriz $p_{i-1} \times p_k$, com $m[i,k]$ multiplicações escalares.

Existe uma ordem para $(A_{k+1} \cdots A_j)$, que é uma matriz $p_k \times p_j$, com $m[k+1,j]$ multiplicações escalares.

Logo, a ordem $(A_i \cdots A_k)(A_{k+1} \cdots A_j)$ utiliza $m[i,k] + p_{i-1} p_k p_j + m[k+1,j]$ multiplicações escalares.

Prova de corretude da recorrência

$$\begin{array}{ccccccccc} p_0 & & p_1 & & p_2 & \cdots & p_{n-1} & & p_n \\ & A_1 & & A_2 & & \cdots & & A_n & \end{array}$$

$m[i,j] =$ nº mín. de multiplicações escalares p/ calcular $A_i \cdots A_j$

Se $i < j$ então $m[i,j] \geq \min_{i \leq k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$:

Suponha que uma ordem ótima separe $A_i \cdots A_j$ em $(A_i \cdots A_k)$ e $(A_{k+1} \cdots A_j)$ para algum $k \in [i,j)$, usando $x \geq m[i,k]$ mults para $(A_i \cdots A_k)$ e $y \geq m[k+1,j]$ mults para $(A_{k+1} \cdots A_j)$.

Prova de corretude da recorrência

$$\begin{array}{ccccccccc} p_0 & & p_1 & & p_2 & \cdots & p_{n-1} & & p_n \\ & A_1 & & A_2 & & \cdots & & A_n & \end{array}$$

$m[i,j] =$ nº mín. de multiplicações escalares p/ calcular $A_i \cdots A_j$

Se $i < j$ então $m[i,j] \geq \min_{i \leq k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$:

Suponha que uma ordem ótima separe $A_i \cdots A_j$ em $(A_i \cdots A_k)$ e $(A_{k+1} \cdots A_j)$ para algum $k \in [i,j)$, usando $x \geq m[i,k]$ mults para $(A_i \cdots A_k)$ e $y \geq m[k+1,j]$ mults para $(A_{k+1} \cdots A_j)$.
Logo,

$$\begin{aligned} m[i,j] &= x + p_{i-1} p_k p_j + y \\ &\geq m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \\ &\geq \min_{i \leq k' < j} \{ m[i,k'] + p_{i-1} p_{k'} p_j + m[k'+1,j] \} \end{aligned}$$

Prova de corretude da recorrência

$$\begin{array}{ccccccccc} p_0 & & p_1 & & p_2 & \cdots & p_{n-1} & & p_n \\ & A_1 & & A_2 & & \cdots & & A_n & \end{array}$$

$m[i,j] =$ nº mín. de multiplicações escalares p/ calcular $A_i \cdots A_j$

Se $i < j$ então $m[i,j] \geq \min_{i \leq k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$:

Suponha que uma ordem ótima separe $A_i \cdots A_j$ em $(A_i \cdots A_k)$ e $(A_{k+1} \cdots A_j)$ para algum $k \in [i,j)$, usando $x \geq m[i,k]$ mults para $(A_i \cdots A_k)$ e $y \geq m[k+1,j]$ mults para $(A_{k+1} \cdots A_j)$.
Total: $x + p_{i-1} p_k p_j + y$ mults.

Logo, $x = m[i,k]$ e $y = m[k+1,j]$. Portanto,

$$\begin{aligned} m[i,j] &= m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \\ &\geq \min_{i \leq k' < j} \{ m[i,k'] + p_{i-1} p_{k'} p_j + m[k'+1,j] \} \end{aligned}$$

Algoritmo recursivo

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

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

```
REC-MAT-CHAIN (p, i, j)
 1  se i = j
 2      então devolva 0
 3  m[i, j] ← ∞
 4  para k ← i até j − 1 faça
 5      q1 ← REC-MAT-CHAIN (p, i, k)
 6      q2 ← REC-MAT-CHAIN (p, k + 1, j)
 7      q ← q1 + p[i − 1]p[k]p[j] + q2
 8      se q < m[i, j]
 9          então m[i, j] ← q
10  devolva m[i, j]
```

Algoritmo recursivo

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

```
REC-MAT-CHAIN (p, i, j)
 1  se i = j
 2      então devolva 0
 3  m[i, j] ← ∞
 4  para k ← i até j − 1 faça
 5      q1 ← REC-MAT-CHAIN (p, i, k)
 6      q2 ← REC-MAT-CHAIN (p, k + 1, j)
 7      q ← q1 + p[i − 1]p[k]p[j] + q2
 8      se q < m[i, j]
 9          então m[i, j] ← q
10  devolva m[i, j]
```

Corretude: imediata da corretude da recorrência

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

$$T(1) = 0$$

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

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

Consumo de tempo

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

$$T(1) = 0$$

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

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

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

Exercício: prove que $T(n) = \frac{3^{n-1}-1}{2}$ para $n \geq 1$.

Conclusão

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

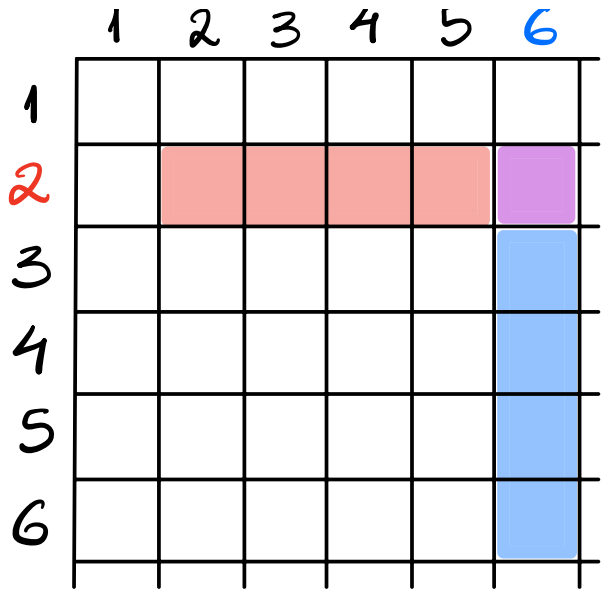
$$A_i \cdots A_j$$

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

$$A_i \cdots A_j$$

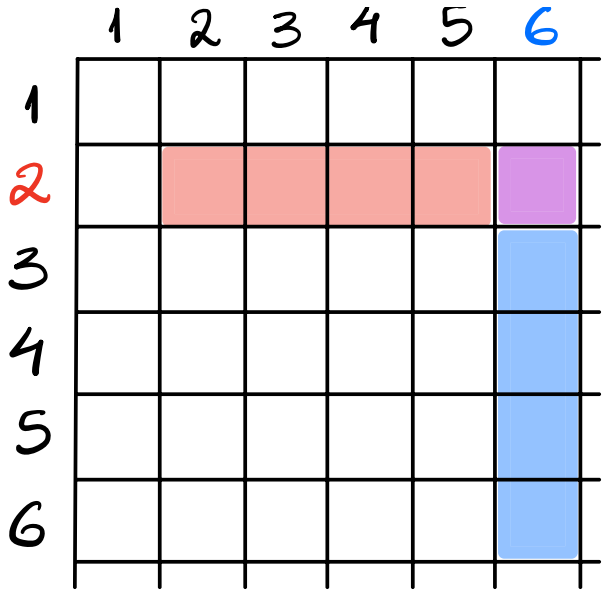
é resolvido uma só vez.

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

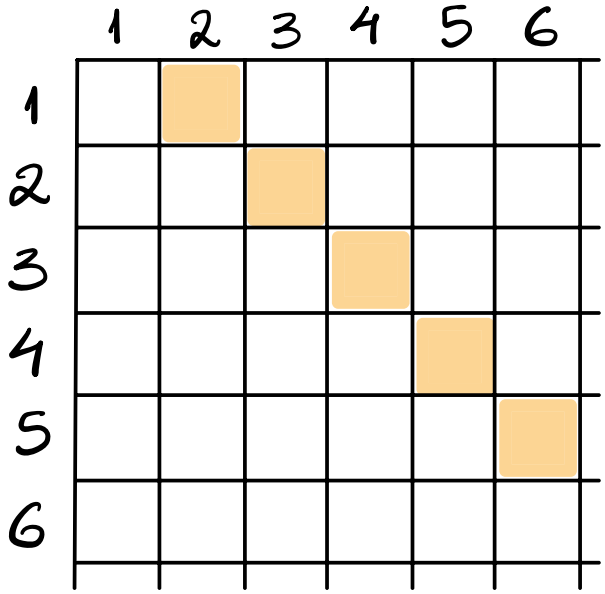
$$A_i \cdots A_j$$

é resolvido uma só vez.

Em que ordem calcular os componentes da tabela m?

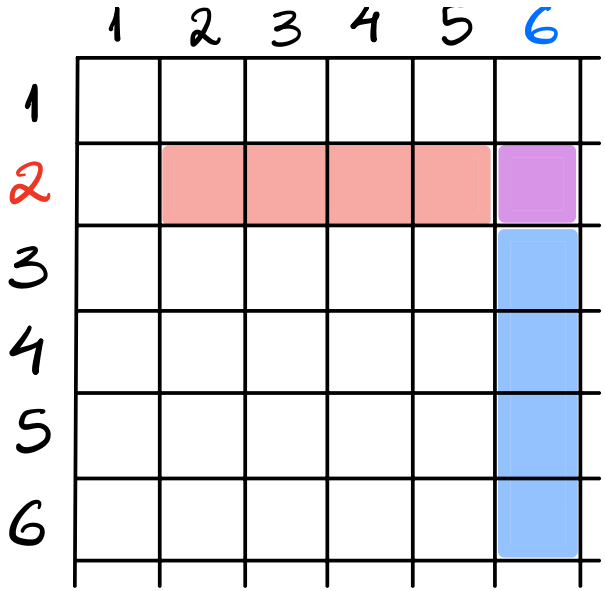
Para calcular $m[2,6]$ preciso de ...

$m[2,2]$, $m[2,3]$, $m[2,4]$, $m[2,5]$ 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



Cada subproblema

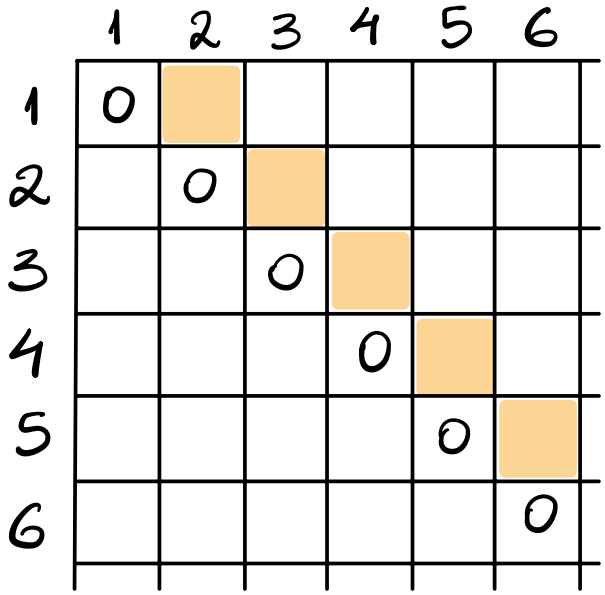
$$A_i \cdots A_j$$

é resolvido uma só vez.

Em que ordem calcular os componentes da tabela m?

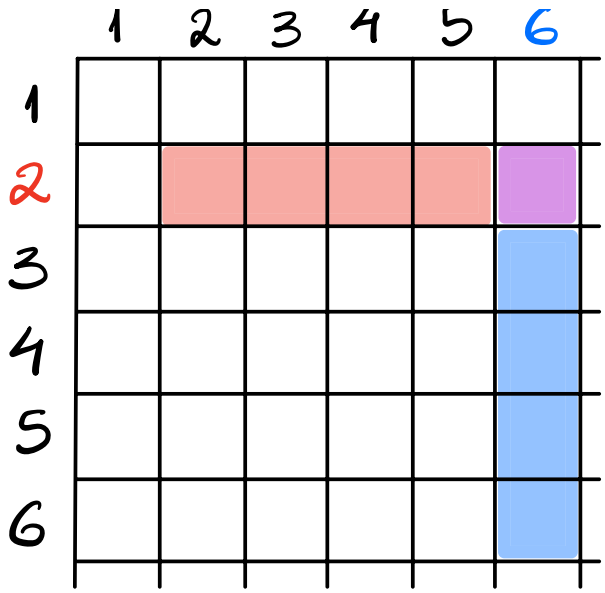
Para calcular $m[2,6]$ preciso de ...

$m[2,2]$, $m[2,3]$, $m[2,4]$, $m[2,5]$ 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



Cada subproblema

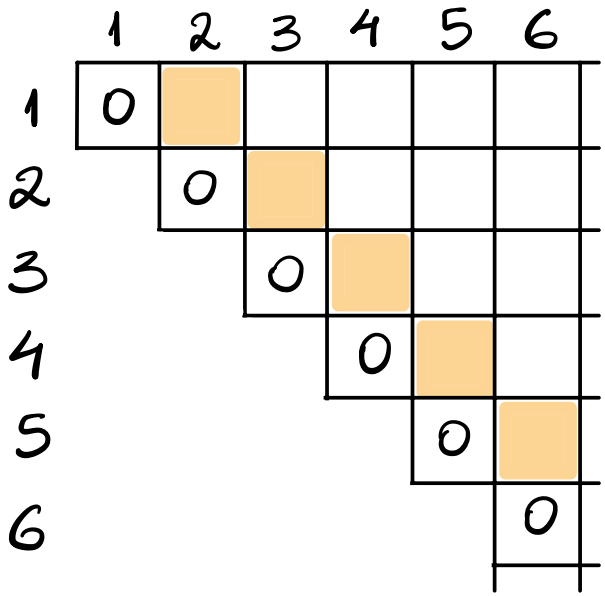
$$A_i \cdots A_j$$

é resolvido uma só vez.

Em que ordem calcular os componentes da tabela m?

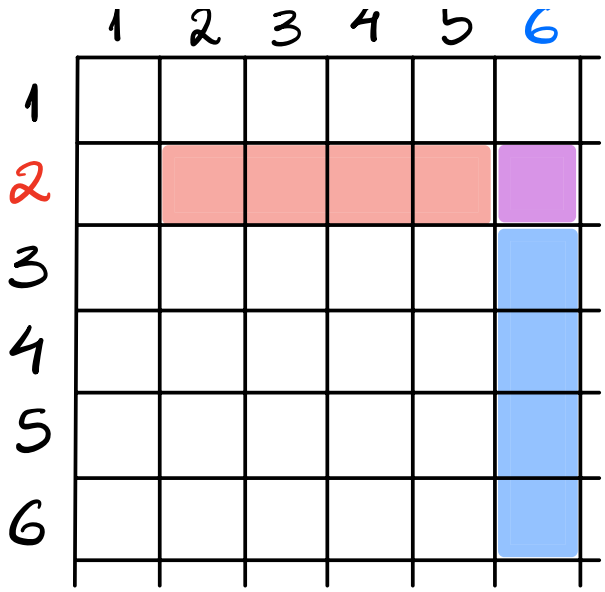
Para calcular $m[2,6]$ preciso de ...

$m[2,2]$, $m[2,3]$, $m[2,4]$, $m[2,5]$ 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



Cada subproblema

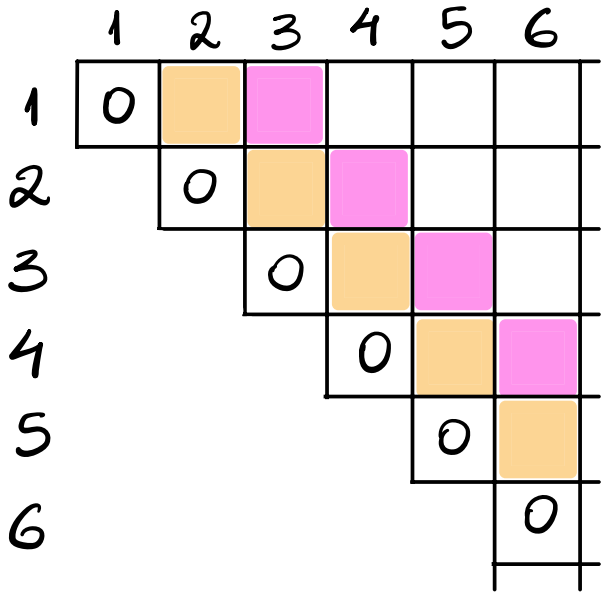
$$A_i \cdots A_j$$

é resolvido uma só vez.

Em que ordem calcular os componentes da tabela m?

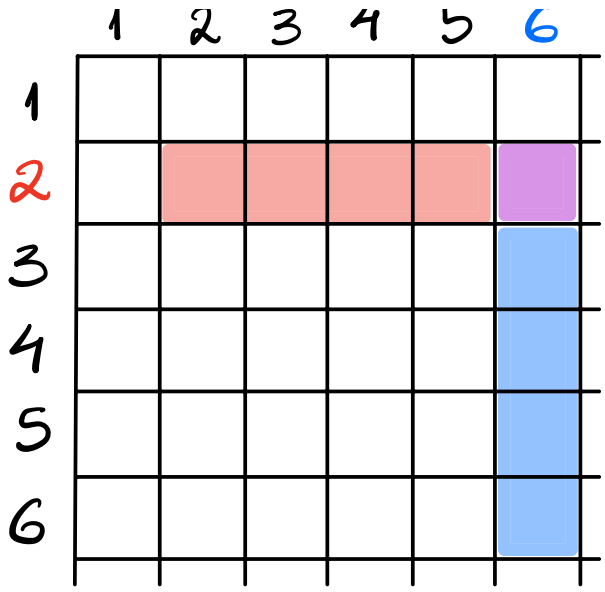
Para calcular $m[2,6]$ preciso de ...

$m[2,2]$, $m[2,3]$, $m[2,4]$, $m[2,5]$ 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



Cada subproblema

$$A_i \cdots A_j$$

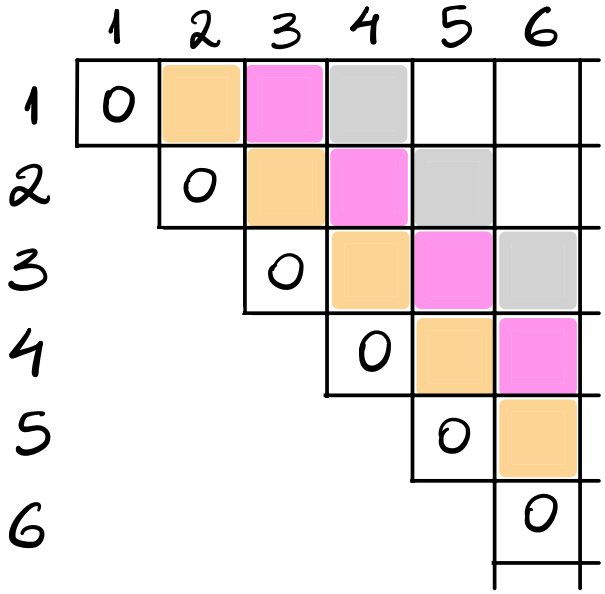
é resolvido uma só vez.

Em que ordem calcular os componentes da tabela m?

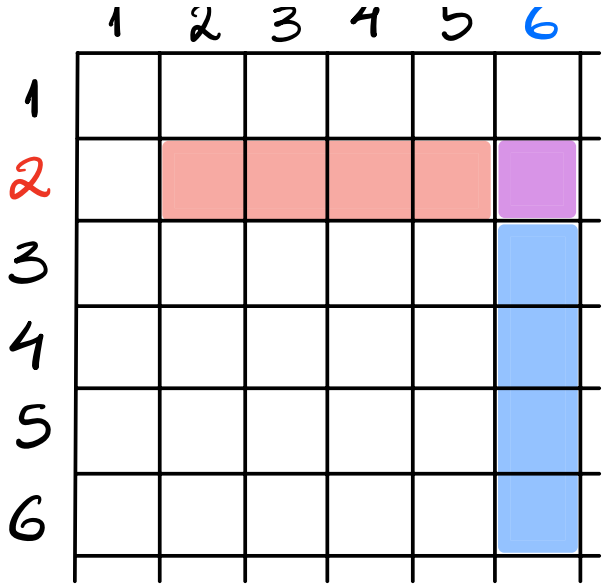
Para calcular $m[2,6]$ preciso de ...

$m[2,2]$, $m[2,3]$, $m[2,4]$, $m[2,5]$ 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



Cada subproblema

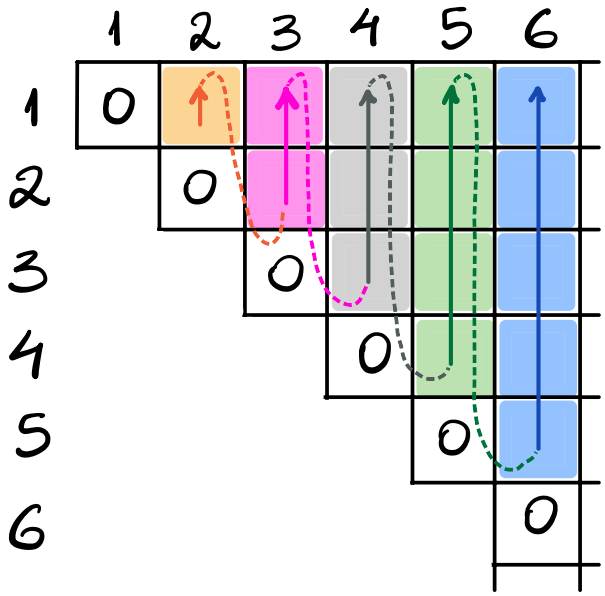
$$A_i \cdots A_j$$

é resolvido uma só vez.

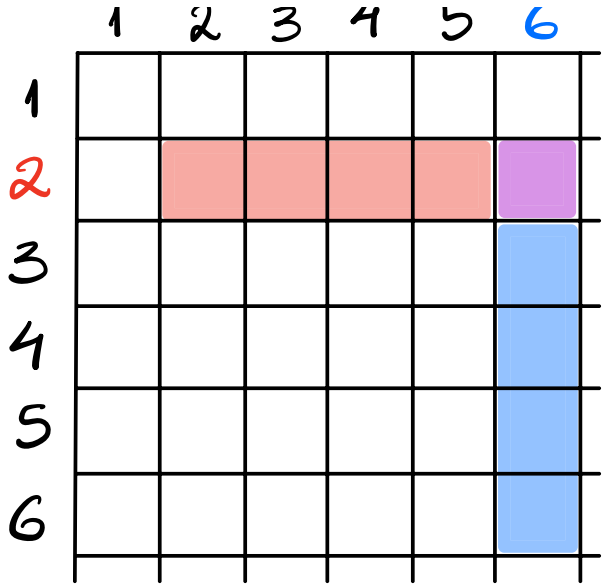
Em que ordem calcular os componentes da tabela m?

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

$$A_i \cdots A_j$$

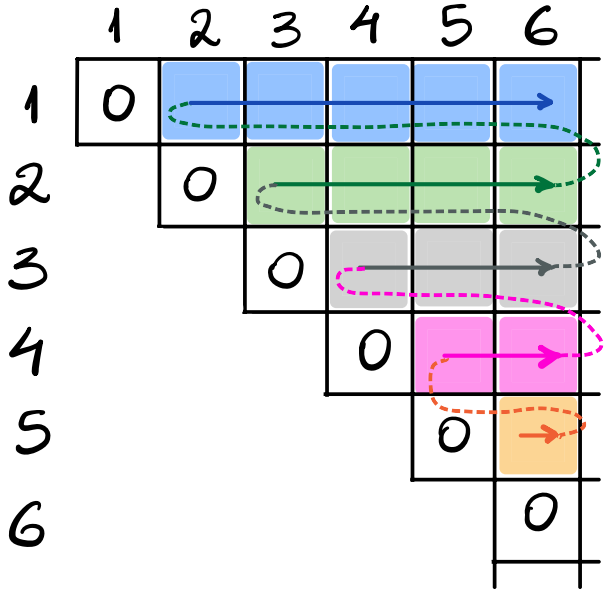
é resolvido uma só vez.

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

$m[3,6]$, $m[4,6]$, $m[5,6]$, $m[6,6]$.



Simulação

$$\text{se } i < j \text{ então } m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$$

$$\underset{A_1}{p_0=10} \quad \underset{A_2}{p_1=10} \quad \underset{A_3}{p_2=20} \quad \underset{A_4}{p_3=30} \quad \underset{A_5}{p_4=10} \quad \underset{A_6}{p_5=15} \quad \underset{}{p_6=30}$$

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

j

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

p_0=10 p_1=10 p_2=20 p_3=30 p_4=10 p_5=15 p_6=30

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

j

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

i \ 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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

i \ j	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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

j

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

j

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

j

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

i

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

j

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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	
i							

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

j

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

j

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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	
i							

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	1	2	3	4	5	6
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

j

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	1	2	3	4	5	6
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

j

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	1	2	3	4	5	6
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

j

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	1	2	3	4	5	6
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

j

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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

p_0=10 p_1=10 p_2=20 p_3=30 p_4=10 p_5=15 p_6=30

A_1 (A_2 A_3 A_4 A_5 A_6)

i \ j	1	2	3	4	5	6
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

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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	
i							

Simulação

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

$(\; A_1 \quad (A_2 \quad A_3 \quad A_4 \quad A_5 \quad A_6) \;)$

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

	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

se $i < j$ então $m[i,j] = \min_{i \le k < j} \{ m[i,k] + p_{i-1} p_k p_j + m[k+1,j] \}$

$p_0{=}10 \quad p_1{=}10 \quad p_2{=}20 \quad p_3{=}30 \quad p_4{=}10 \quad p_5{=}15 \quad p_6{=}30$

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

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

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

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

MATRIX-CHAIN-ORDER (p, n)

```
 1  para i ← 1 até n faça
 2      m[i, i] ← 0
 3  para ℓ ← 2 até n faça          ▷ ℓ é comprimento da cadeia A_i ··· A_j
 4      para i ← 1 até n faça      ▷ FIXME
 5          j ← i + ℓ − 1          ▷ A_{i+0} ··· A_{i+ℓ−1} tem comprimento ℓ
 6          m[i, j] ← ∞
 7          para k ← i até j − 1 faça
 8              q ← m[i, k] + p[i − 1]p[k]p[j] + m[k+1, j]
 9              se q < m[i, j]
10                  então m[i, j] ← q
11  devolva m[1, n]
```

Algoritmo de programação dinâmica

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

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

MATRIX-CHAIN-ORDER (p, n)

```
 1  para i ← 1 até n faça
 2      m[i, i] ← 0
 3  para ℓ ← 2 até n faça          ▷ ℓ é comprimento da cadeia A_i ··· A_j
 4      para i ← 1 até n − ℓ + 1 faça      ▷ FIXED!
 5          j ← i + ℓ − 1          ▷ A_{i+0} ··· A_{i+ℓ−1} tem comprimento ℓ
 6          m[i, j] ← ∞
 7          para k ← i até j − 1 faça
 8              q ← m[i, k] + p[i − 1]p[k]p[j] + m[k+1, j]
 9              se q < m[i, j]
10                  então m[i, j] ← q
11  devolva m[1, n]
```

Consumo de tempo

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

Consumo de tempo

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

Consumo de tempo: ???

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)

Consumo de tempo

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

Consumo de tempo: $O(n^3)$ (três loops encaixados)

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

Versão recursiva com memoização

```
MEMOIZED-MATRIX-CHAIN-ORDER (p, n)
1   para i ← 1 até n faça
2       para j ← 1 até n faça
3           m[i, j] ← ∞
4   devolva LOOKUP-CHAIN (p, 1, n)

LOOKUP-CHAIN (p, i, j)
 1   se m[i, j] < ∞
 2       então devolva m[i, j]
 3   se i = j
 4       então m[i, j] ← 0
 5   para k ← i até j − 1 faça
 6       q ← LOOKUP-CHAIN (p, i, k) + p[i−1]p[k]p[j]
 7                        +  LOOKUP-CHAIN (p, k+1, j)
 8       se q < m[i, j]
 9         então m[i, j] ← q
10   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 $A_1 A_2 \cdots A_n$.

Exercícios

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

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