Динамическое программирование: перемножение последовательности матриц

Александр Куликов

Онлайн-курс «Алгоритмы: теория и практика. Методы»
http://stepic.org/217

Перемножение последовательности матриц

Вход: последовательность n матриц $A_1, \ldots, A_n$, которые нужно перемножить.

Выход: порядок умножения, минимизирующий стоимость умножения.

Замечания

- Обозначим размеры матриц $A_1, \ldots, A_n$ через

$$m_0 \times m_1, m_1 \times m_2, \ldots, m_{n-1} \times m_n\,,$$

соответственно. То есть размер A_i есть $m_{i-1} \times m_i$.

- Умножение матриц не коммутативно (в общем случае, $A \times B \neq B \times A$), но ассоциативно:

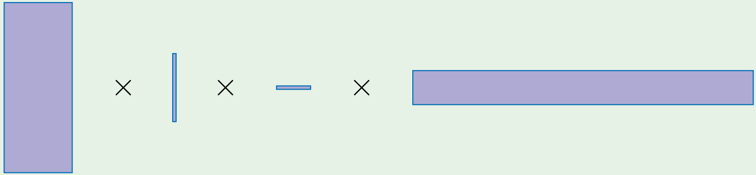
$$A \times (B \times C) = (A \times B) \times C\,.$$

- Значит, $A \times B \times C \times D$ может быть вычислено как

$$(A \times B) \times (C \times D) \text{ или } (A \times (B \times C)) \times D\,.$$

- Стоимость умножения двух матриц размеров $p \times q$ и $q \times r$ будем считать pqr.

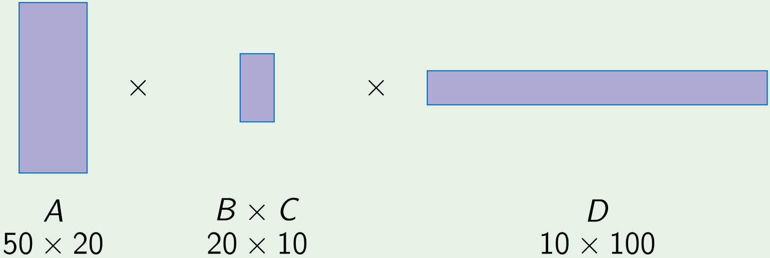
Пример: $A \times ((B \times C) \times D)$



A	B	C	D
50×20	20×1	1×10	10×100

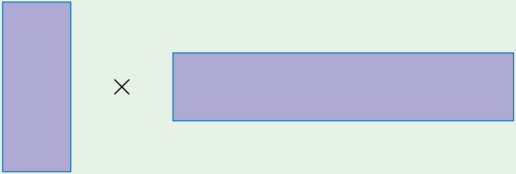
стоимость:

Пример: $A \times ((B \times C) \times D)$



стоимость: $20 \cdot 1 \cdot 10$

Пример: $A \times ((B \times C) \times D)$



A $\qquad$ $B \times C \times D$

50×20 $\qquad$ 20×100

стоимость: $20 \cdot 1 \cdot 10 + 20 \cdot 10 \cdot 100$

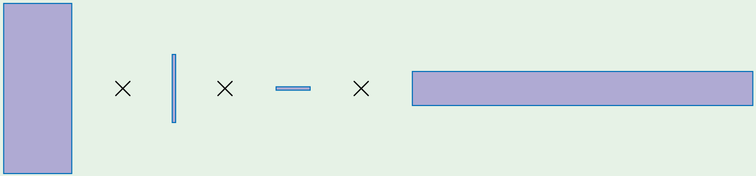
Пример: $A \times ((B \times C) \times D)$

$A \times B \times C \times D$

50×100

стоимость: $20 \cdot 1 \cdot 10 + 20 \cdot 10 \cdot 100 + 50 \cdot 20 \cdot 100 = 120\,200$

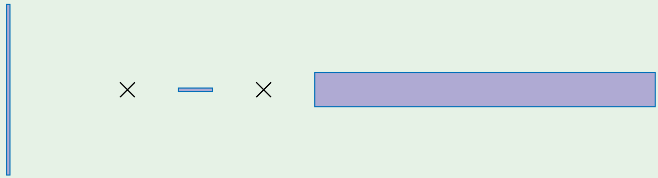
Пример: $(A \times B) \times (C \times D)$



A 50×20 B 20×1 C 1×10 D 10×100

стоимость:

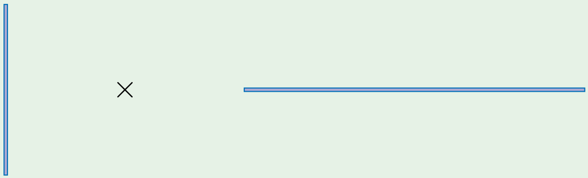
Пример: $(A \times B) \times (C \times D)$



$A \times B$ — 50×1; C — 1×10; D — 10×100

стоимость: $50 \cdot 20 \cdot 1$

Пример: $(A \times B) \times (C \times D)$



$A \times B$
50×1

$C \times D$
1×100

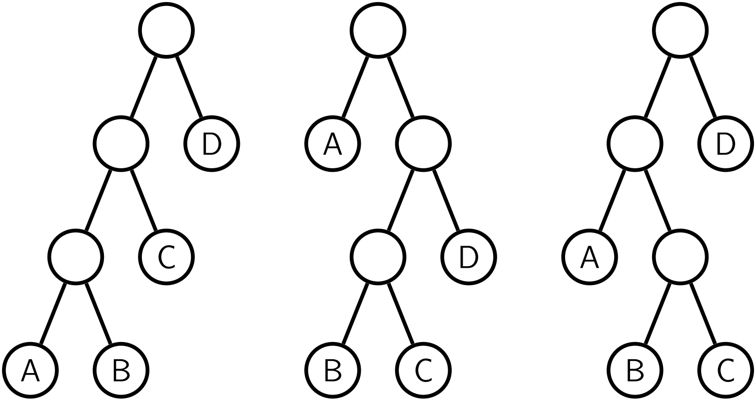
стоимость: $50 \cdot 20 \cdot 1 + 1 \cdot 10 \cdot 100$

Пример: $(A \times B) \times (C \times D)$

$A \times B \times C \times D$
50×100

стоимость: $50 \cdot 20 \cdot 1 + 1 \cdot 10 \cdot 100 + 50 \cdot 1 \cdot 100 = 7\,000$

Порядки как строго двоичные деревья



$((A \times B) \times C) \times D \quad A \times ((B \times C) \times D) \quad (A \times (B \times C)) \times D$

Подзадачи и рекуррентное соотношение

- Для $1 \le i \le j \le n$, пусть

  $$D[i,j] = \text{мин. стоимость вычисления } A_i \times A_{i+1} \times \ldots \times A_j \,.$$

- Корень поддерева разбивает его на два поддерева: $A_i \times \ldots \times A_k$ и $A_{k+1} \times \ldots \times A_j$ (для некоторого $i \le k < j$).
- Рекуррентное соотношение:

$$D[i,j] = \min_{i \le k < j} \{ D[i,k] + D[k+1,j] + m_{i-1} \cdot m_k \cdot m_j \} \,.$$

Дин. прог. сверху вниз

Инициализация

создать таблицу $D[1 \ldots n, 1 \ldots n] \leftarrow [\infty, \ldots, \infty]$

Дин. прог. сверху вниз

Инициализация

создать таблицу $D[1 \ldots n, 1 \ldots n] \leftarrow [\infty, \ldots, \infty]$

Функция $\textsc{MatrixMultTD}(i, j)$

если $D[i,j] = \infty$:
  если $i = j$: $D[i,j] \leftarrow 0$
  иначе:
    для k от i до $j-1$:
      $\ell \leftarrow \textsc{MatrixMultTD}(i, k)$
      $r \leftarrow \textsc{MatrixMultTD}(k+1, j)$
      $D[i,j] \leftarrow \min(D[i,j], \ell + r + m_{i-1} m_k m_j)$
вернуть $D[i,j]$

Дин. прог. сверху вниз

Инициализация

создать таблицу $D[1\dots n, 1\dots n] \leftarrow [\infty, \dots, \infty]$

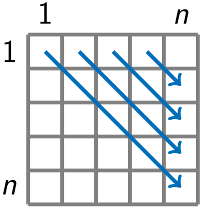
Функция $\textsc{MatrixMultTD}(i, j)$

если $D[i, j] = \infty$:
  если $i = j$: $D[i, j] \leftarrow 0$
  иначе:
    для k от i до $j - 1$:
      $\ell \leftarrow \textsc{MatrixMultTD}(i, k)$
      $r \leftarrow \textsc{MatrixMultTD}(k + 1, j)$
      $D[i, j] \leftarrow \min(D[i, j], \ell + r + m_{i-1} m_k m_j)$
вернуть $D[i, j]$

Время работы: $O(n^3)$.

Порядок подзадач

- Хотим идти от меньших подзадач к бо́льшим.
- Размером подзадачи естественно считать требующееся количество умножений: $j - i$.
- Возможный порядок:



Дин. прог. снизу вверх

Функция $\textsc{MatrixMultBU}(m_0, m_1, \ldots, m_n)$

```
создать массив D[1...n, 1...n] ← [∞, ..., ∞]
для i от 1 до n:
  D[i, i] ← 0
для s от 1 до n − 1:
  для i от 1 до n − s:
    j ← i + s
    для k от i до j − 1:
      D[i, j] ← min(D[i, j], D[i, k] + D[k + 1, j] + m_{i−1} m_k m_j)
вернуть D[1, n]
```

Дин. прог. снизу вверх

Функция $\textsc{MatrixMultBU}(m_0, m_1, \ldots, m_n)$

```
создать массив D[1...n, 1...n] ← [∞, ..., ∞]
для i от 1 до n:
   D[i, i] ← 0
для s от 1 до n − 1:
   для i от 1 до n − s:
      j ← i + s
      для k от i до j − 1:
         D[i, j] ← min(D[i, j], D[i, k] + D[k + 1, j] + m_{i−1} m_k m_j)
вернуть D[1, n]
```

Время работы: $O(n^3)$.

Пример

$$m_0 = 50,\ m_1 = 20,\ m_2 = 1,\ m_3 = 10,\ m_4 = 100$$

	1	2	3	4
1	0	1000	1500	7000
2		0	200	3000
3			0	1000
4				0