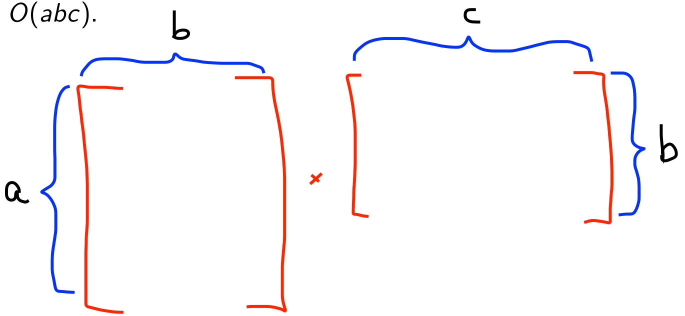
Graduate Algorithms.
Georgia Institute of Technology.

Sequencing DP.

Matrix multiplication

Input: a sequence of matrices $A_1, A_2, \dots, A_n$.
Output: the optimal way to compute the product $A_1 A_2 \dots A_n$.

Recall: a matrix of size $a \times b$ can be multiplied by a matrix of size $b \times c$ in time $O(abc)$.



Matrix multiplication

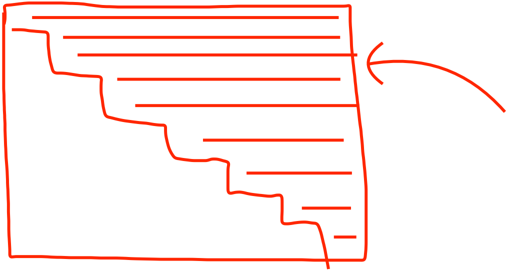
Input: a sequence of $\cancel{\text{matrices } A_1, A_2, \dots, A_n}$ natural numbers $m_0, m_1, \dots, m_n$ where $A_i \in M_{m_{i-1} \times m_i}(\mathbb{R})$.
Output: the optimal way to compute the product $A_1 A_2 \dots A_n$.

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

Table definition

$T[i,j] =$ min cost for calculating the product $A_i A_{i+1} \dots A_j$

Note: only need $i \leq j$ since the order matters.

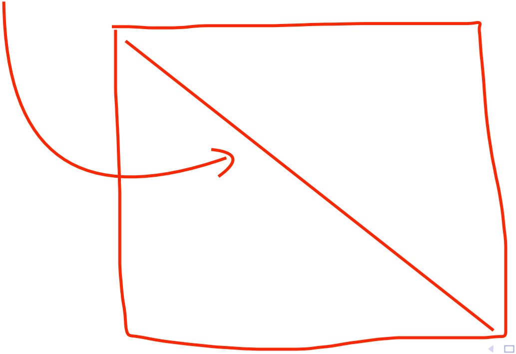


$i \leq j$

Recurrence and base case

$$T[i,j] = \min_{i \leq \ell < j} \{ T[i,\ell] + T[\ell+1,j] + m_{i-1} m_\ell m_j \}$$

Base cases: $T[i,i] = 0$.



Pseudocode

CHAIN MATRIX MULTIPLICATION

1. $C(i,i) = 0$ for all i
2. For $s = 1$ to $n-1$
   (a) For $i = 1$ to $n-s$
      i. $j = i + s$
      ii. $C(i,j) = \min_{i \le k < j}\{C(i,k) + C(k+1,j) + m_{i-1} \cdot m_k \cdot m_j\}$
3. Return $C(1,n)$.

Running time: $O(n^3)$.