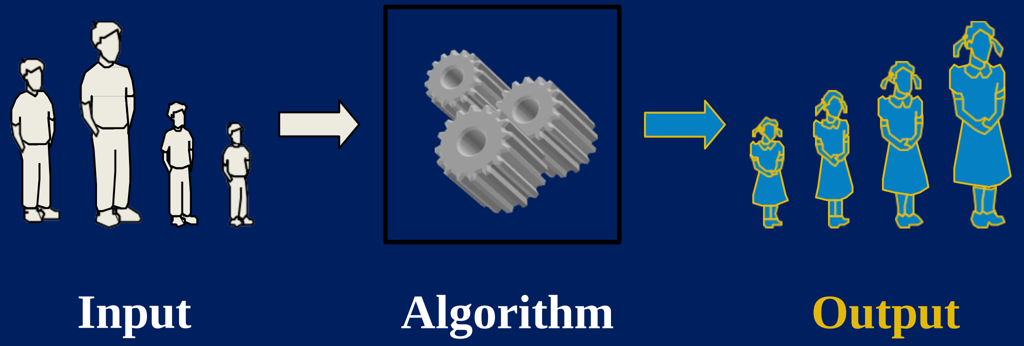
DESIGN & ANALYSIS OF ALGORITHM (BCSC0012)

Chapter 12: Dynamic Programming

Chained Matrix Multiplications



Prof. Anand Singh Jalal

Department of Computer Engineering & Applications

Chained Matrix Multiplications

Problem: given a sequence $\langle A_1, A_2, \ldots, A_n \rangle$, compute the product:

$$A_1 \cdot A_2 \cdots A_n$$

- Matrix compatibility:

$$C = A \cdot B$$
$$col_A = row_B$$
$$row_C = row_A$$
$$col_C = col_B$$

$$C = A_1 \cdot A_2 \cdots A_i \cdot A_{i+1} \cdots A_n$$
$$col_i = row_{i+1}$$
$$row_C = row_{A1}$$
$$col_C = col_{An}$$

Algorithm to Multiply 2 Matrices

Input: Matrices $A_{p \times q}$ and $B_{q \times r}$ (with dimensions $p \times q$ and $q \times r$)

Result: Matrix $C_{p \times r}$ resulting from the product $A \cdot B$

MATRIX-MULTIPLY$(A_{p \times q}, B_{q \times r})$

1. **for** $i \leftarrow 1$ **to** p
2. **for** $j \leftarrow 1$ **to** r
3. $C[i, j] \leftarrow 0$
4. **for** $k \leftarrow 1$ **to** q
5. $C[i, j] \leftarrow C[i, j] + A[i, k] \cdot B[k, j]$
6. **return** C

Scalar multiplication in line 5 dominates time to compute CNumber of scalar multiplications = *pqr*

Matrix-chain Multiplication

- Suppose we have a sequence or chain A1, A2, …, An of n matrices to be multiplied
- That is, we want to compute the product A1A2…An
- There are many possible ways (parenthesizations) to compute the product

- Example: consider the chain A_1, A_2, A_3, A_4 of 4 matrices
  - Let us compute the product $A_1A_2A_3A_4$
- There are 5 possible ways:

1. $(A_1(A_2(A_3A_4)))$
2. $(A_1((A_2A_3)A_4))$
3. $((A_1A_2)(A_3A_4))$
4. $((A_1(A_2A_3))A_4)$
5. $(((A_1A_2)A_3)A_4)$

No of possible ways: $\frac{1}{n}\,{}^{2(n-1)}C_{(n-1)}$

Matrix-chain Multiplication …

- Example: Consider three matrices $A_{10\times100}$, $B_{100\times5}$, and $C_{5\times50}$
- There are 2 ways to parenthesize
  - $((AB)C) = D_{10\times5} \cdot C_{5\times50}$
    - $AB \Rightarrow 10\cdot100\cdot5 = 5{,}000$ scalar multiplications
    - $DC \Rightarrow 10\cdot5\cdot50 = 2{,}500$ scalar multiplications

    Total: 7,500
  - $(A(BC)) = A_{10\times100} \cdot E_{100\times50}$
    - $BC \Rightarrow 100\cdot5\cdot50 = 25{,}000$ scalar multiplications
    - $AE \Rightarrow 10\cdot100\cdot50 = 50{,}000$ scalar multiplications

    Total: 75,000

Matrix-chain Multiplication …

- Matrix-chain multiplication problem
  - Given a chain $A_1, A_2, \ldots, A_n$ of n matrices, where for i=1, 2, …, n, matrix A_i has dimension $p_{i-1} \times p_i$
  - Parenthesize the product $A_1A_2\ldots A_n$ such that the total number of scalar multiplications is minimized
- Brute force method of exhaustive search takes time exponential in n

$$\begin{array}{ccccccccc} A_1 & \cdot & A_2 & \cdots & A_i & \cdot & A_{i+1} & \cdots & A_n \\ p_0 \times p_1 & & p_1 \times p_2 & & p_{i-1} \times p_i & & p_i \times p_{i+1} & & p_{n-1} \times p_n \end{array}$$

Matrix-chain Multiplication …

The Structure of an Optimal Parenthesization

- Notation:

$$A_{i\dots j} = A_i\, A_{i+1} \cdots A_j,\ i \le j$$

- Suppose that an optimal parenthesization of $A_{i\dots j}$ splits the product between A_k and A_{k+1}, where $i \le k < j$

$$\begin{aligned} A_{i\dots j} &= A_i\, A_{i+1} \cdots A_j \\ &= A_i\, A_{i+1} \cdots A_k\, A_{k+1} \cdots A_j \\ &= A_{i\dots k}\, A_{k+1\dots j} \end{aligned}$$

The parenthesization of the "prefix" $A_{i\dots k}$ must be an optimal parentesization

Matrix-chain Multiplication ...

A Recursive Solution

- **Sub-problem:** Determine the minimum cost of parenthesizing

  $A_{i\ldots j} = A_i A_{i+1} \cdots A_j$ for $1 \le i \le j \le n$

- Let $m[i, j]$ = the minimum number of multiplications needed to compute $A_{i\ldots j}$
  - full problem ($A_{1..n}$): $m[1, n]$
  - $i = j$: $A_{i\ldots i} = A_i \Rightarrow m[i, i] = 0$, for $i = 1, 2, \ldots, n$

Matrix-chain Multiplication ...

A Recursive Solution

- Consider the subproblem of parenthesizing

$$A_{i\ldots j} = A_i\, A_{i+1} \cdots A_j \qquad \text{for } 1 \le i \le j \le n$$

$$= \underbrace{A_{i\ldots k}}_{m[i,\,k]}\ \underbrace{A_{k+1\ldots j}}_{m[k+1,j]} \quad p_{i-1}p_kp_j \qquad \text{for } i \le k < j$$

- Assume that the optimal parenthesization splits the product $A_i\, A_{i+1} \cdots A_j$ at k ($i \le k < j$)

$$m[i, j] = \underbrace{m[i, k]}_{\substack{\text{min \# of multiplications}\\ \text{to compute } A_{i\ldots k}}} + \underbrace{m[k+1, j]}_{\substack{\text{min \# of multiplications}\\ \text{to compute } A_{k+1\ldots j}}} + \underbrace{p_{i-1}p_kp_j}_{\substack{\text{\# of multiplications}\\ \text{to compute } A_{i\ldots k}A_{k\ldots j}}}$$

Matrix-chain Multiplication ...
A Recursive Solution

$$m[i, j] = m[i, k] + m[k+1, j] + p_{i-1}p_kp_j$$

- We do not know the value of k
  - There are j – i possible values for k: k = i, i+1, …, j-1
- Minimizing the cost of parenthesizing the product $A_i A_{i+1} \cdots A_j$ becomes:

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

Matrix-chain Multiplication …

Matrix-Chain-Order(p)

MATRIX-CHAIN-ORDER(p)

```
n ← length[p] − 1
for i ← 1 to n
      do m[i, i] ← 0
for l ← 2 to n                    ▷ l is the chain length.
      do for i ← 1 to n − l + 1
                do j ← i + l − 1
                   m[i, j] ← ∞
                   for k ← i to j − 1
                         do q ← m[i, k] + m[k + 1, j] + p_{i−1} p_k p_j
                            if q < m[i, j]
                               then m[i, j] ← q
                                    s[i, j] ← k
return m and s
```

There are 3 nested loops and each can iterate at most n times, so the total running time is $O(n^3)$.

Basically, we're checking different places to "split" our matrices by checking different values of k and seeing if they improve our current minimum value.

Matrix-chain Multiplication …
Extracting Optimum Sequence

- Leave a split marker indicating where the best split is (i.e. the value of k leading to minimum values of $m[i, j]$). We maintain a parallel array $s[i, j]$ in which we store the value of k providing the optimal split.

- If $s[i, j] = k$, the best way to multiply the sub-chain $A_{i\ldots j}$ is to first multiply the sub-chain $A_{i\ldots k}$ and then the sub-chain $A_{k+1\ldots j}$, and finally multiply them together. Intuitively $s[i, j]$ tells us what multiplication to perform *last*. We only need to store $s[i, j]$ if we have at least 2 matrices & $j > i$.

Matrix-chain Multiplication …

Example: DP for CMM

Matrix Chain Multiplication

$$A_1 \cdot A_2 \cdot A_3 \cdot A_4$$

$5\times4 \quad 4\times6 \quad 6\times2 \quad 2\times7$

m	1	2	3	4
1	0	120	88	
2		0	48	
3			0	84
4				0

$m[1,3] = 88$

$$\min \left\{ \boxed{\begin{array}{c} A_1 \cdot (A_2 \cdot A_3) \\ 5\times4 \quad 4\times6 \quad 6\times2 \end{array}} , \boxed{\begin{array}{c} (A_1 \cdot A_2) \cdot A_3 \\ 5\times4 \quad 4\times6 \quad 6\times2 \end{array}} \right\}$$

$m[1,1] + m[2,3] + 5\times4\times2 \qquad m[1,2] + m[3,3] + 5\times6\times2$

$0 + 48 + 40 \qquad 120 + 0 + 60$

$\underline{88} \qquad 180$

S	1	2	3	4
1		1		
2				
3				
4				

Matrix-chain Multiplication ...

Example: DP for CMM

Matrix Chain Multiplication

$$A_1 \cdot A_2 \cdot A_3 \cdot A_4$$

$$\underset{d_0}{5} \times \underset{d_1}{4} \quad 4 \times \underset{d_2}{6} \quad 6 \times \underset{d_3}{2} \quad 2 \times \underset{d_4}{7}$$

$$m[1,4] = \min \begin{cases} \underset{0}{m[1,1]} + \underset{104}{m[2,4]} + \underset{+140}{5\times4\times7}, \quad \underset{120}{m[1,2]} + \underset{84}{m[3,4]} + \underset{210}{5\times6\times7} \\ \underset{88}{m[1,3]} + \underset{0}{m[4,4]} + \underset{70}{5\times2\times7} \end{cases}$$

$$m[i,j] = \min \left\{ m[i,k] + m[k+1,j] + d_{i-1} * d_k * d_j \right\}$$

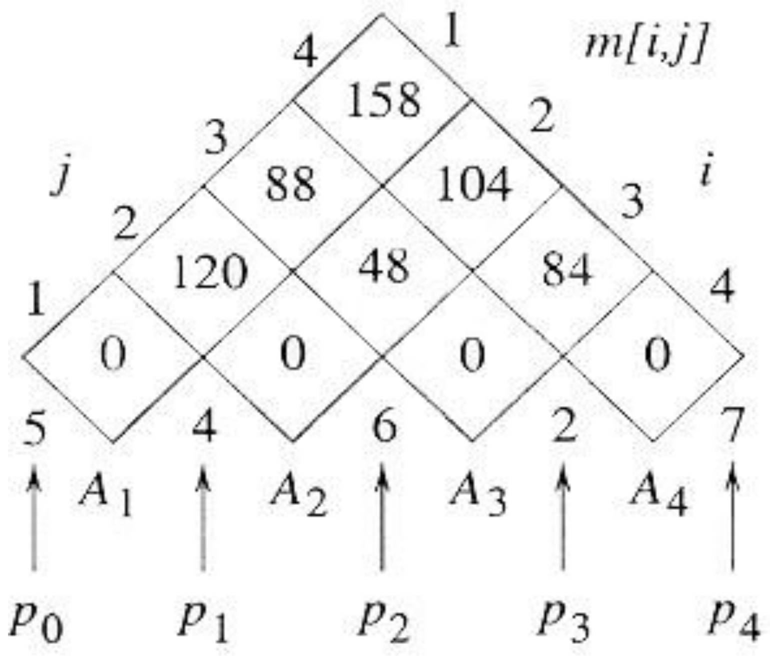
m	1	2	3	4
1	0	120	88	158
2		0	48	104
3			0	84
4				0

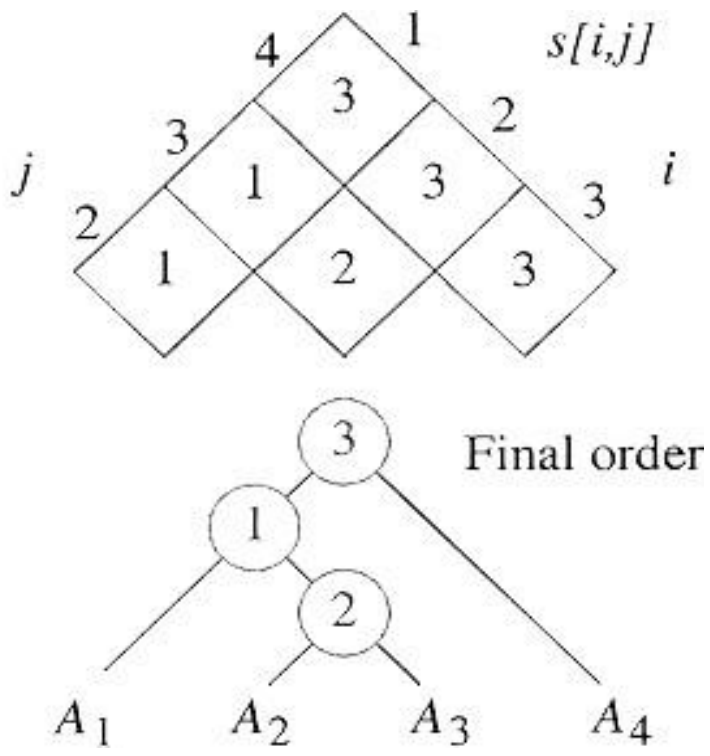
s	1	2	3	4
1		1	1	3
2			2	3
3				3
4				

Matrix-chain Multiplication …

Example: DP for CMM

- The initial set of dimensions are <5, 4, 6, 2, 7>: we are multiplying A_1 (5x4) times A_2 (4x6) times A_3 (6x2) times A_4 (2x7). Optimal sequence is $(A_1 (A_2 A_3)) A_4$.





Any Questions ?

Dr. Anand Singh Jalal
Professor
Email: asjalal@gla.ac.in