Dynamic Programming (DP)

Algorithm design technique mainly used in optimization problems.

⇓

can have many feasible solutions but only one solution is optimal.

↙ ↘

Max. Min. (According to the given problem)

Steps in DP:

1. Characterize the structure of an optimal solution.
2. Recursively define the value of an optimal solution.
3. Compute the value of an optimal solution, typically in a bottom-up fashion.
4. Construct an optimal solution from computed information.

Note: If we need only the value of an optimal solution, not the solution itself, then we can omit step-4. Otherwise, we can construct optimal solution from the informations obtained in step-3.

Matrix Chain Multiplication (MCM)

Matrix multiplication:

- No. of columns of 1st matrix = No. of rows of 2nd matrix
- $(A \cdot B) \cdot C = A \cdot (B \cdot C)$ ⇒ Associative property.
- $A_{2 \times \underline{3}}, B_{\underline{3} \times 4} \Rightarrow P_{2 \times 4}$ (To get $P_{2 \times 4}$, we have to perform $2 \times 3 \times 4 = 24$ no. of scalar multiplications)

$$A_{2 \times 3} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \quad B_{3 \times 4} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{bmatrix}$$

$$P_{2 \times 4} = \begin{bmatrix} a_{11} \overset{\downarrow}{\cdot} b_{11} + a_{12} \overset{\downarrow}{\cdot} b_{21} + a_{13} \overset{\downarrow}{\cdot} b_{31} & \cdots \\ \vdots & \vdots \end{bmatrix}$$

⇒ Mimization type of optimization problem.

MCM Problem

Our objective is to find out how we can fully parenthesize the product so that it minimizes the no. of scalar multiplications.

⇒ We are not concerned about the product of the matrix chain.

$$\begin{array}{ccc} A_1 & A_2 & A_3 \\ 2 \times 3 & 3 \times 4 & 4 \times 2 \end{array}$$

$$\overbrace{(A_1 \cdot A_2)}^{2 \times 4} \cdot A_3 \qquad A_1 \cdot \underbrace{(A_2 \cdot A_3)}_{3 \times 2}$$

$$\begin{array}{ccc} 2 \times 3 & 3 \times 4 & 4 \times 2 \end{array} \qquad \begin{array}{ccc} 2 \times 3 & 3 \times 4 & 4 \times 2 \end{array}$$

$$= \underline{2 \times 3 \times 4} + \underline{0} + \underline{2 \times 4 \times 2} \qquad = 0 + \underline{3 \times 4 \times 2} + \underline{2 \times 3 \times 2}$$

= 40 scalar multiplications = $\boxed{36}$ scalar multiplications

Optimal parenthesization

$((A_1) \cdot (A_2 \cdot A_3))$

Formal definition of MCM problem: Given a chain $\langle A_1, A_2, \cdots, A_n \rangle$ of n matrices, where for $i = 1, 2, \cdots, n$, matrix A_i has dimension $p_{i-1} \times p_i$, fully parenthesize the product $A_1 A_2 \cdots A_n$ in a way that minimizes the no. of scalar multiplications.

Example:

$$\begin{array}{cccc} A_1 & A_2 & A_3 & A_4 \\ 5\times4 & 4\times6 & 6\times2 & 2\times7 \end{array} \qquad \langle 5, 4, 6, 2, 7 \rangle$$

$$\overset{A_1}{m[1,1]} = \overset{A_2}{m[2,2]} = \overset{A_3}{m[3,3]} = \overset{A_4}{m[4,4]} = 0$$

$m[1,2]$

$A_1 \cdot A_2$

$5\times4 \quad 4\times6$

$5\times4\times6 = 120$

$((A_1)\cdot(A_2))$

$m[2,3]$

$A_2 \cdot A_3$

$4\times6 \quad 6\times2$

$4\times6\times2 = 48$

$m[3,4]$

$A_3 \cdot A_4$

$6\times2 \quad 2\times7$

$6\times2\times7 = 84$

$m \to$ cost matrix

$$\begin{array}{c|c|c|c|c|} m & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 120 & 88 & 158 \\ \hline 2 & & 0 & 48 & 104 \\ \hline 3 & & & 0 & 84 \\ \hline 4 & & & & 0 \\ \hline \end{array}$$

$S \to$ Parenthesization matrix

$$\begin{array}{c|c|c|c|c|} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 1 & 3 \\ \hline 2 & & 0 & 2 & 3 \\ \hline 3 & & & 0 & 3 \\ \hline 4 & & & & 0 \\ \hline \end{array}$$

$m[1,3]$

$A_1 \cdot A_2 \cdot A_3$

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

$A_1 \cdot (\overset{4\times2}{A_2 \cdot A_3})$

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

$= 0 + 48 + 40 = \boxed{88}$

$\overset{5\times6}{(A_1 \cdot A_2)} \cdot A_3$

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

$= 120 + 0 + 60 = \underline{\underline{180}}$

$m[2,4]$

$A_2 \cdot A_3 \cdot A_4$

$A_2 \cdot \overset{\nearrow 6\times7}{(A_3 \cdot A_4)}$

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

$m[2,2] + m[3,4] + 4\times6\times7$

$= 0 + 84 + 168$

$= 252$

$\overset{\nearrow 4\times2}{(A_2 \cdot A_3)} \cdot A_4$

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

$m[2,3] + m[4,4] + 4\times2\times7$

$= 48 + 0 + 56$

$= \boxed{104}$

$\overset{\nearrow 5\times4}{A_1} \cdot \overset{\nearrow 4\times7}{(A_2 \cdot A_3 \cdot A_4)}$

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

$$m[1,4] = \min \left\{ \underline{m[1,1] + m[2,4] + 5\times4\times7}, \; m[1,2] + m[3,4] + 5\times6\times7, \; m[1,3] + m[4,4] + 5\times2\times7 \right\}$$

$(A_1 \cdot A_2) \cdot (A_3 \cdot A_4)$

$(A_1 \cdot A_2 \cdot A_3) \cdot A_4$

$$= \min \{ 0 + 104 + 140, \; 120 + 84 + 210, \; 88 + 0 + 70 \}$$

$$= \min \{ 244, 414, \underline{\underline{158}} \} = 158$$

So, we need 158 scalar multiplications for this matrix chain.

Optimal parenthesization:

$$\underline{((A_1)(A_2 \; A_3))(A_4)}$$

→ dimensions of matrices

P = <5, 4, 6, 2, 7>
P.length = 5

n^3 → Time
n^2 → Space

```
MATRIX-CHAIN-ORDER(p)
n = p.length − 1
let m[1..n, 1..n] and s[1..n − 1, 2..n] be new tables
for i = 1 to n
    m[i, i] = 0
for l = 2 to n                    // l is the chain length
    for i = 1 to n − l + 1
        j = i + l − 1
        m[i, j] = ∞
        for k = i to j − 1
            q = m[i, k] + m[k + 1, j] + $p_{i-1} p_k p_j$
            if q < m[i, j]
                m[i, j] = q
                s[i, j] = k
return m and s
```

m

j \ i	1	2	3	4
4	158	104	84	0
3	88	48	0	
2	120	0		
1	0			

P-O-P

```
PRINT-OPTIMAL-PARENS(s, i, j)
if i == j
    print "A"_i
else print "("
    PRINT-OPTIMAL-PARENS(s, i, s[i, j])
    PRINT-OPTIMAL-PARENS(s, s[i, j] + 1, j)
    print ")"
```

i ↓ 1, j ↓ 4

S

j \ i	1	2	3
4	3	3	3
3	1	2	
2	1		

A_1 A_2 A_3 A_4

(

P-O-P(S,1,3) — [P-O-P(S,1,1)
P-O-P(S,2,3) — [P-O-P(S,2,2)
P-O-P(S,3,3)

→ P-O-P(S,4,4)

Steps of DP w.r. to MCM problem:

1. **Structure of an optimal solution:**

$$A_i A_{i+1} \cdots A_j \quad , i \leq j$$

To parenthesize this product, we must split it between A_k and A_{k+1} for some k in the range $i \leq k < j$. Then, we first compute the matrices $A_i \cdots A_k$ and $A_{k+1} \cdots A_j$ and then multiply them together to produce $A_i \cdots A_j$. The cost of parenthesizing is the cost of computing the matrix product $A_i \cdots A_k$ plus the cost of computing $A_{k+1} \cdots A_j$ plus the cost of multiplying them together.

2. **Recursively define the value of an optimal solution**

$$m[i,j] = \begin{cases} 0 & , \text{if } i=j \\ \min\limits_{i \leq k < j} \{ m[i,k] + m[k+1,j] + P_{i-1} \cdot P_k \cdot P_j & , \text{if } i<j \end{cases}$$

3. **Computing the value of an optimal solution in bottom up fashion:**

Compute m & s tables in bottom up fashion using Matrix_Chain_Order procedure.

4. **Constructing an optimal solution** (Optional step):

Parenthesization step using Print_Optimal_Parens procedure.