

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.

1

Matrix Chain Multiplication (MCM)

Matrix multiplication:

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

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

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

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

\Rightarrow Minimization type of optimization problem.

\Leftarrow MCM Problem \Rightarrow We are not concerned about the product of the matrix chain.

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

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

$$= 2 \times 3 \times 4 + 0 + 2 \times 4 \times 2$$

$$= 40 \text{ scalar multiplications}$$

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

$$= 0 + 3 \times 4 \times 2 + 2 \times 3 \times 2$$

$$= 36 \text{ scalar multiplications}$$

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

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

Example: $A_1 \quad A_2 \quad A_3 \quad A_4$
 $5 \times 4 \quad 4 \times 6 \quad 6 \times 2 \quad 2 \times 7$

$$m[1,1] = m[2,2] = m[3,3] = m[4,4] = 0$$

$$m[1,2] \quad m[2,3] \quad m[3,4] = 0$$

$$\begin{array}{c} A_1 \cdot A_2 \\ 5 \times 4 \quad 4 \times 6 \\ 5 \times 4 \times 6 = 120 \\ ((A_1) \cdot (A_2)) \end{array} \quad \begin{array}{c} A_2 \cdot A_3 \\ 4 \times 6 \quad 6 \times 2 \\ 4 \times 6 \times 2 = 48 \end{array} \quad \begin{array}{c} A_3 \cdot A_4 \\ 6 \times 2 \quad 2 \times 7 \\ 6 \times 2 \times 7 = 84 \end{array}$$

$$m[1,3]$$

$$A_1 \cdot A_2 \cdot A_3$$

$$5 \times 4 \quad 4 \times 6 \quad 6 \times 2$$

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

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

$$= m[1,1] + m[2,3] = m[1,2] + m[3,3] + 5 \times 6 \times 2$$

$$+ 5 \times 4 \times 2 = 120 + 0 + 60 = 180$$

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

$\langle 5, 4, 6, 2, 7 \rangle$

\rightarrow Cost matrix
 m

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

$S \rightarrow$ Parenthesization matrix

	1	2	3	4
1	0	1	1	3
2		0	2	3
3			0	3
4				0

$$m[2,4]$$

$$A_2 \cdot A_3 \cdot A_4$$

$$\begin{array}{l}
 \swarrow \nearrow 6 \times 7 \quad \searrow \nearrow 4 \times 2 \\
 A_2 \cdot (A_3 \cdot A_4) \quad (A_2 \cdot A_3) \cdot A_4 \\
 4 \times 6 \quad 6 \times 2 \quad 2 \times 7 \quad 4 \times 6 \quad 6 \times 2 \quad 2 \times 7 \\
 m[2,2] + m[3,4] \quad m[2,3] + m[4,4] + 4 \times 2 \times 7 \\
 + 4 \times 6 \times 7 \\
 = 0 + 84 + 168 \quad = 48 + 0 + 56 \\
 = 252 \quad = \boxed{104}
 \end{array}$$

$$\begin{array}{l}
 A_1 \cdot A_2 \cdot A_3 \cdot A_4 \\
 m[1,4] = \min \left\{ \begin{array}{l}
 \overbrace{A_1 \cdot (A_2 \cdot A_3 \cdot A_4)}^{5 \times 4 \quad 4 \times 7} \\
 m[1,1] + m[2,4] + 5 \times 4 \times 7, \\
 \overbrace{m[1,2] + m[3,4] + 5 \times 6 \times 7}^{(A_1 \cdot A_2) \cdot (A_3 \cdot A_4)} \\
 \overbrace{m[1,3] + m[4,4] + 5 \times 2 \times 7}^{(A_1 \cdot A_2 \cdot A_3) \cdot A_4}
 \end{array} \right\}
 \end{array}$$

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

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

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

Optimal parenthesization:

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

MATRIX-CHAIN-ORDER(p)

```

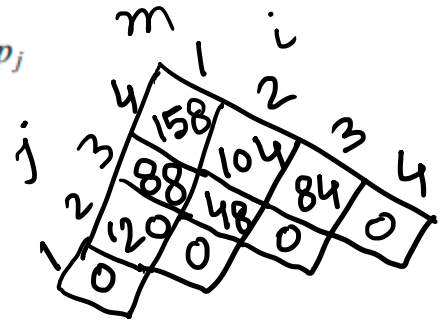
1   $n = p.length - 1$ 
2  let  $m[1..n, 1..n]$  and  $s[1..n-1, 2..n]$  be new tables
3  for  $i = 1$  to  $n$ 
4       $m[i, i] = 0$ 
5  for  $l = 2$  to  $n$  //  $l$  is the chain length
6      for  $i = 1$  to  $n - l + 1$ 
7           $j = i + l - 1$ 
8           $m[i, j] = \infty$ 
9          for  $k = i$  to  $j - 1$ 
10              $q = m[i, k] + m[k + 1, j] + p_{i-1}p_kp_j$ 
11             if  $q < m[i, j]$ 
12                  $m[i, j] = q$ 
13                  $s[i, j] = k$ 
14  return  $m$  and  $s$ 

```

→ dimensions of matrices

$P = \langle 5, 4, 6, 2, 7 \rangle$
 $p.length = 5$

$n^3 \rightarrow$ Time
 $n^2 \rightarrow$ Space

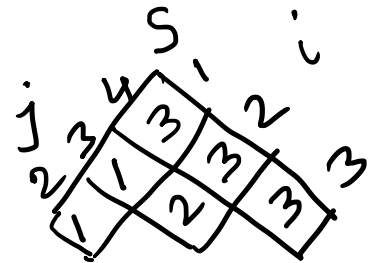


PRINT-OPTIMAL-PARENS(s, i, j)

```

1  if  $i == j$ 
2      print " $A_i$ "
3  else print "("
4      PRINT-OPTIMAL-PARENS( $s, i, s[i, j]$ )
5      PRINT-OPTIMAL-PARENS( $s, s[i, j] + 1, j$ )
6      print ")"

```



$A_1 A_2 A_3 A_4$ (

$P-O-P(s, 1, 3) \rightarrow P-O-P(s, 1, 1) \rightarrow P-O-P(s, 2, 2)$
 $\rightarrow P-O-P(s, 4, 4) \rightarrow P-O-P(s, 2, 3) \rightarrow P-O-P(s, 3, 3)$

Steps of DP w.r. to MCM problem:

1. Structure of an optimal solution:

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

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 $A_i \dots A_k$ and $A_{k+1} \dots A_j$ and then multiply them together to produce $A_i \dots A_j$. The cost of parenthesizing is the cost of computing the matrix product $A_i \dots A_k$ plus the cost of computing $A_{k+1} \dots A_j$ plus the cost of multiplying them together.

2. Recursively define the value of an optimal solution

$$m[i, j] = \begin{cases} 0 & , \text{if } i=j \\ \min_{i \leq k < j} \{m[i, k] + m[k+1, j] + p_{i-1} \cdot p_k \cdot p_j\} & , \text{if } i < j \end{cases}$$

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.