①

Dynamic Programming Approach L1/L2

→ In this approach, solve the subproblem just once and save the solution in a Table.

→ The solution will be retrieve when the same problem is encountered later on or when we need them later to solve longer subproblems.

Steps in Developing a dynamic Programming problems:-

Followings are the steps required to developing a dynamic Programming problems

→ characterize the structure of an optimal solution

→ Recursively define the value of an optimal solution

→ compute the value of an optimal solution in a bottom-up fashion.

→ construct an optimal solution from computed information

Examples of Problem that uses dynamic programming approach to find optimal solution's:-

→ MultiStage Graph Problem

→ 0/1 knapsack Problem

→ Travelling Sales Person Problem

→ matrix chain Multiplication Problem.

→ All pairs Shortest path problem

→

matrix Chain Multiplication Problem:

Problem: We are given sequence or chain of n matrices $A_1 A_2 - A_n$ to be multiplied and our objective is to compute Product $A_1 A_2 \cdots A_n$ with minimum cost i.e Minimum no. of Computation required.

We know that matrix Multiplication is Associative so all ways of Parenthesization will yield the same answer ($n>=3$) but cost of evaluation varies drastically.

Let us consider example to compute the product of matrices $A_1 A_2 A_3 A_4$. There are five different ways of parenthesization to compute the product of $A_1 A_2 A_3 A_4$ and the answer by each one will be same. But cost of evaluation varies drastically. The ways of parenthesization are

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

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

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

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

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

Example: Suppose we compute the product of three matrices $A_1 A_2 A_3$

$A_1 : 10 \times 100$

$A_2 : 100 \times 5$

$A_3 : 5 \times 50$

there are two ways of parenthesization of $A_1 A_2 A_3$.

$((A_1 A_2) A_3)$: total cost $= 10 \times 100 \times 5 + 10 \times 5 \times 50 = 7500$

$(A_1 (A_2 A_3))$: total cost $= 100 \times 5 \times 50 + 10 \times 100 \times 50 = 75000$

⇒ out of two possible paranthesization, second paranthesization gives optimal result of cost of computation.

Explanation of matrix chain multiplication Problem:

The matrix chain multiplication Problem can be stated as follows: Given a chain of n matrices $\langle A_1, A_2 \cdots A_n \rangle$, 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 minimize the number of Scalar Multiplication.

Let us use dynamic Programming approach and problem is divided as:

for $i < j$

$A_{i \cdots j}$ denotes $A_i \cdot A_{i+1} A_{i+2} \cdots A_j$

To evaluate $A_{i \cdots j}$ we must find an Index k such that

$$A_{i \cdots j} = A_{i \cdots k} \cdot A_{k+1 \cdots j} \quad \text{for Some } k \text{ in between } i \,\&\, j$$

$\Rightarrow$ total cost to get $A_{i \cdots j}$ = total cost to get $A_{i \cdots k}$ + total cost to get $A_{k+1 \cdots j}$ + cost of combining.

$\Rightarrow$ optimal cost of $A_{i \cdots j}$ can be obtained by getting each of $A_{i \cdots k}$ and $A_{k+1 \cdots j}$ optimally.

$\Rightarrow$ The optimal solution of orginal problem can be obtaned from the optimal Solutions of Sub Problems.

Recursive Solution is given as:

Let $m[i, j]$ = minimal no. of computation needed to compute matrix $A_i A_{i+1} \cdots A_j$ or $A_{i \cdots j}$

Now $m[i, j]$ can be defined as follows:

$$m[i, i] = 0 \quad \text{for } i = 1, \cdots n$$

Let k be the index for optimal solution then $m[i, j]$ can be defined recursively as

$$m[i, j] = m[i, k] + m[k+1 \cdots j] + p_{i-1} \times p_k \times p_j$$

$\Rightarrow m[i,j] = m[i,k] + m[k+1,j] + p_{i-1} p_k p_j$

$\Downarrow$ minimum cost for computing subproblem $A_{i \cdots k}$

$\Downarrow$ minimum cost for computing subproblem $A_{k+1 \cdots j}$

$\Downarrow$ cost of combining the two matrices together.

$\Rightarrow$ the recursive definition for the minimum cost of parenthesizing the product $A_i A_{i+1} \cdots A_j$ becomes.

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

to keep track of successive decisions, let use $S[i,j]$ to denote the value of index k from which sequence $A_i \cdots A_j$ to be partitioned.

$\Rightarrow$ optimal costs are calculated by using a tabular bottom-up approach. Here we use auxillary tables.

$m[1 \cdots n, 1 \cdots n]$ for storing $m[i,j]$ cost and

$S[1 \cdots n-1, 2 \cdots n]$ for recording indices (k) for optimal cost.

Algorithm for matrix-chain-order(P):

```
matrix-chain-order(P)
{
1. n ← length[p] - 1 ;
2. for i = 1 to n do
3. m[i,i] ← 0 ;
4. for l = 2 to n do        // l is the chain length
   {
5.   for i = 1 to n - l + 1 do
     {
6.     j = i + l - 1
7.     m[i,j] = ∞
8.     for k = i to j - 1 do
9.     {
         q ← m[i,k] + m[k+1,j] + p_{i-1} × p_k × p_j
10.      if (q < m[i,j]) then
         {
11.        m[i,j] = q ;
12.        s[i,j] = k ;
         }
       }
     }
   }
13. Return m & s .
```

Example: Multiply the following matrices in chain and find the optimal solution.

matrix	dimension	
A_1	30×35	$p_0 \times p_1$
A_2	35×15	$p_1 \times p_2$
A_3	15×5	$p_2 \times p_3$
A_4	5×10	$p_3 \times p_4$
A_5	10×20	$p_4 \times p_5$
A_6	20×25	$p_5 \times p_6$

⑥

$\Rightarrow$ p_0 p_1 p_2 p_3 p_4 p_5 p_6
30 35 15 5 10 20 25

length[p] = 7
$\Rightarrow n = \text{length}[p] - 1 = 6$

we construct two table

table for m

i \ j →	1	2	3	4	5	6
1	0	15750	7875	9375	11875	15125
2		0	2625	4375	7125	10500
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

table for s

i \ j →	2	3	4	5	6
1	1	1	3	3	3
2		2	3	3	3
3			3	3	3
4				4	5
5					5

From the algorithm.

$m[i,i] = 0$ for $i = 1, \cdots n$

$\Rightarrow m[1,1] = m[2,2] = m[3,3] = m[4,4] = m[5,5] = m[6,6] = 0$

then we compute $m[1,2] = \min\{m[1,1] + m[2,2] + p_0 p_1 p_2\}$

$= 0 + 0 + 30 \times 35 \times 15$ since $k = i$ to $j-1 \Rightarrow k = 1$

$= 15750$

then we compute

$m[2,3] = \min\{\underbrace{m[2,2]}_{=0} + \underbrace{m[3,3]}_{=0} + p_1 \times p_2 \times p_3\}$ $k=2$

$= p_1 \times p_2 \times p_3 = 35 \times 15 \times 5 = 2625$ $k=3$

Similarly,

$m[3,4] = p_2 \times p_3 \times p_4 = 15 \times 5 \times 10 = 750$ $k=4$

$m[4,5] = p_3 \times p_4 \times p_5 = 5 \times 10 \times 20 = 1000$ $k=5$

$m[5,6] = p_4 \times p_5 \times p_6 = 10 \times 20 \times 25 = 5000$

then we compute

$$m[1,3] = \min_{i \le k < j} \begin{cases} m[1,1] + m[2,3] + p_0 \times p_1 \times p_3 \\ m[1,2] + m[3,3] + p_0 \times p_2 \times p_3 \end{cases}$$

from the table, the values of $m[1,1]$, $m[1,2]$, $m[2,3]$ & $m[3,3]$ are saved. hence take these value from table and find $m[1,3]$

$$\Rightarrow m[1,3] = \min \begin{cases} \overset{0}{m[1,1]} + \overset{2625}{m[2,3]} + \overset{30}{p_0} \times \overset{35}{p_1} \times \overset{5}{p_3} \\ \underset{15750}{m[1,2]} + \underset{0}{m[3,3]} + \overset{30}{p_0} \times \overset{15}{p_2} \times \overset{5}{p_3} \end{cases}$$

$$= \min \begin{cases} 2625 + 5250 = 7875 \\ 15750 + 2250 = 18000 \end{cases}$$

$= 7875$ $\Rightarrow k = 1$ for optimized cost

find $m[2,4] = \min \begin{cases} \overset{0}{m[2,2]} + m[3,4] + p_1 \times p_2 \times p_4 = 750 + 5250 = 6000 \\ m[2,3] + \underset{0}{m[4,4]} + p_1 \times p_3 \times p_4 = 2625 + 1750 = 4375 \end{cases}$

$= 4375$ & $k = 3$ in which cost optimized.

find $m[3,5] = \min \begin{cases} \overset{0}{m[3,3]} + m[4,5] + p_2 \times p_3 \times p_5 \\ m[3,4] + \underset{0}{m[5,5]} + p_2 \times p_4 \times p_5 \end{cases}$

$$= \min \begin{cases} 1000 + 1500 = 2500 \\ 750 + 3000 = 3750 \end{cases}$$

$= 2500$ & $k = 3$ in which cost optimized

find $m[4,6] = \min \begin{cases} \overset{0}{m[4,4]} + m[5,6] + p_3 \times p_4 \times p_6 = 5000 + 1250 = 6250 \\ m[4,5] + \underset{0}{m[6,6]} + p_3 \times p_5 \times p_6 = 1000 + 2500 = 3500 \end{cases}$

$= 3500$ & $k = 5$ in which cost optimum.

find $m[1,4] = \min \begin{cases} \overset{0}{m[1,1]} + m[2,4] + p_0 \times p_1 \times p_4 = 4375 + 10500 = 14875 \\ m[1,2] + m[3,4] + p_0 \times p_2 \times p_4 = 15750 + 750 + 4500 = 21000 \\ m[1,3] + \underset{0}{m[4,4]} + p_0 \times p_3 \times p_4 = 7875 + 1500 = 9375 \end{cases}$

$= 9375$ & $k = 3$ in which cost optimum.

⑧ $m[2,5] = \min \begin{cases} m[2,2] + m[3,5] + p_1 \times p_2 \times p_5 = 2500 + 10500 = 13000 \\ m[2,3] + m[4,5] + p_1 \times p_3 \times p_5 = 2625 + 1000 + 3500 = 7125 \\ m[2,4] + m[5,5] + p_1 \times p_4 \times p_5 = 4375 + 7000 = 11375 \end{cases}$

$= 7125$

& $k = 3$ for which cost is optimum.

$m[3,6] = \min \begin{cases} m[3,3] + m[4,6] + p_2 \times p_3 \times p_6 = 3500 + 1875 = 5375 \\ m[3,4] + m[5,6] + p_2 \times p_4 \times p_6 = 750 + 5000 + 3750 = 9500 \\ m[3,5] + m[6,6] + p_2 \times p_5 \times p_6 = 2500 + 7500 = 10000 \end{cases}$

$= 5375$

& $k = 3$ in which cost is optimum.

$m[1,5] = \min \begin{cases} m[1,1] + m[2,5] + p_0 \times p_1 \times p_5 = 7125 + 21000 = 28125 \\ m[1,2] + m[3,5] + p_0 \times p_2 \times p_5 = 15750 + 2500 + 9000 = 27250 \\ m[1,3] + m[4,5] + p_0 \times p_3 \times p_5 = 7875 + 1000 + 3000 = 11875 \\ m[1,4] + m[5,5] + p_0 \times p_4 \times p_5 = 9375 + 6000 = 15375 \end{cases}$

$= 11875$

& $k = 3$ for which cost is optimum.

$m[2,6] = \min \begin{cases} m[2,2] + m[3,6] + p_1 \times p_2 \times p_6 = 5375 + 13125 = 18500 \\ m[2,3] + m[4,6] + p_1 \times p_3 \times p_6 = 2625 + 3000 + 4375 = 10500 \\ m[2,4] + m[5,6] + p_1 \times p_4 \times p_6 = 4375 + 5000 + 8750 = 18125 \\ m[2,5] + m[6,6] + p_1 \times p_5 \times p_6 = 7125 + 17500 = 24625 \end{cases}$

$= 10500$

& $k = 3$ for which cost is optimum.

$m[1,6] = \min \begin{cases} m[1,1] + m[2,6] + p_0 \times p_1 \times p_6 = 10500 + 26250 = 36750 \\ m[1,2] + m[3,6] + p_0 \times p_2 \times p_6 = 15750 + 5375 + 11250 = 32375 \\ m[1,3] + m[4,6] + p_0 \times p_3 \times p_6 = 7875 + 3500 + 3750 = 15125 \\ m[1,4] + m[5,6] + p_0 \times p_4 \times p_6 = 9375 + 5000 + 7500 = 21875 \\ m[1,5] + m[6,6] + p_0 \times p_5 \times p_6 = 11875 + 15000 = 26875 \end{cases}$

$= 15125$ & $k = 3$ for which $m[1,6]$ is optimum.

(a)

$\Rightarrow$ total number of computation required to compute
$A_1 \cdot A_2 \cdot A_3 A_4 \cdot A_5 \cdot A_6 = A_{1 \cdots 6} = 15125$
which gives optimal solution

$\Rightarrow$ optimal way to multiplication of $A_{1 \cdots 6}$.

$= A'_{1 \cdots 3} \times A_{4 \cdots 6}$.

$= (A_1 \cdot A_{2 \cdot 3}) \times (A_{4 \cdot 5} \cdot A_6)$

$= ((A_1 \cdot (A_2 \cdot A_3))((A_4 \cdot A_5) \cdot A_6))$

[see the table S for the value of k]

$= 35 \times 15 \times 5 + 30 \times 35 \times 5 + 5 \times 10 \times 20 + 5 \times 20 \times 25 + 30 \times 5 \times 25$

$= 2625 + 5250 + 1000 + 2500 + 3750$

$= 15125$