Design and Analysis of Algorithm

Dynamic Programming
(Matrix Chain Multiplication)

Lecture – 58

Overview

- Dynamic Programming is a general algorithm design technique for solving problems defined by recurrences with overlapping subproblems
- Invented by American mathematician "Richard Bellman) in the year 1950s to solve optimization problems and later assimilated by Computer Science.
- "Programming" here means "planning"

Dynamic Programming

- “Method of solving complex problems by breaking them down into smaller sub-problems, solving each of those sub-problems just once, and storing their solutions.”
- The problem solving approach looks like Divide and conquer approach.(which is not true)

Dynamic Programming

Difference between Dynamic programming and Divide and Conquer approach.

Divide & Conquer	Dynamic Programming
1. Partitions a problem into independent smaller sub-problems	1. Partitions a problem into overlapping sub-problems
2. Doesn't store solutions of sub-problems. (Identical sub-problems may arise - results in the same computations are performed repeatedly.)	2. Stores solutions of sub-problems: thus avoids calculations of same quantity twice
3. Top down algorithms: which logically progresses from the initial instance down to the smallest sub-instances via intermediate sub-instances.	3. Bottom up algorithms: in which the smallest sub-problems are explicitly solved first and the results of these used to construct solutions to progressively larger sub-instances

Dynamic Programming

Is a Four-step methods

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.

Dynamic Programming

Problems:

1. 0/1 Knapsack Problem
2. Floyd-Warshall Algorithm
3. Longest Common Sub-sequence
4. Matrix Chain Multiplication

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Problem:

"Given dimensions $p_0, p_1, p_2, \ldots \ldots, p_n$ corresponding to matrix sequence $\langle A_1, A_2, \ldots \ldots, A_n \rangle$ of n matrices, where for $i = 1, 2, \ldots, n$, matrix A_i has dimension $p_{i-1} \times p_i$, determine the "multiplication sequence" that minimizes the number of scalar multiplications in computing $\langle A_1, A_2, \ldots \ldots, A_n \rangle$."

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Problem:

That is determine ho to parenthesize the multiplication.

Example:

$$A_1, A_2, A_3, A_4 = \begin{array}{l} ((A_1A_2)(A_3A_4)) \\ (A_1(A_2(A_3A_4))) \\ (A_1((A_2A_3)A_4)) \\ ((A_1A_2)(A_3A_4)) \\ (((A_1A_2)A_3)A_4) \end{array}$$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Given a $p \times q$ matrix A, a $q \times r$ matrix B and a $r \times s$ matrix C, then ABC can be computed in two ways (AB)C and A(BC):

The number of multiplications needed are:

$$mult[(AB)C] = pqr + prs,$$
$$mult[A(BC)] = qrs + pqs.$$

When p = 5, q = 4, r = 6 and s = 2, then

$$mult[(AB)C] = 180,$$
$$mult[A(BC)] = 88.$$

Which is a big difference. Hence the implication is the the multiplication "sequence" (parenthesization) is very important.

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 1: Characterize the structure of an optimal solution

- Decompose thee problem into subproblems:
  - For each pair $1 \leq i \leq j \leq n$, determine the multiplication sequence for $A_{i..j} = A_i, A_{i+1}, \ldots \ldots, A_j$ that minimize the number of multiplications.
  - Clearly, $A_{i..j}$ is a $p_{i-1} \times p_i$ matrix.
- High-Level Parenthesization for $A_{i..j}$
  - For any optimal multiplication sequence, at the last step you are multiplying two matrices $A_{i..k}$ and $A_{k+1..j}$ for some k. That is,

$$A_{i..j} = (A_i \ldots A_k)\left(A_{k+1} \ldots A_j\right) = A_{i..k}\, A_{k+1..j}$$

  - Example

$$A_{3..6} = ((A_3(A_4A_5))(A_6) = A_{3..5}A_{6..6}. \; (Here\ k = 5.)$$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 1: Characterize the structure of an optimal solution

- Thus the problem of determining the optimal sequence of multiplications is dividedinto 2 questions:
  - How do we decide where to split the chain (what is the value of k)?

    (Search all possible values of k)
  - How do we parenthesize the sub chains $A_{i..k}$ and $A_{k+1..j}$?

    (Problem has optimal substructure property that $A_{i..k}$ and $A_{k+1..j}$ must be optimal so the same procedure can be applied recursively)

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 1: Characterize the structure of an optimal solution

- What is Optimal Substructure Property?
  - If final "optimal" solution of $A_{i..j}$ involves splitting into $A_{i..k}$ and $A_{k+1..j}$ at final step then parenthesization of $A_{i..k}$ and $A_{k+1..j}$ in final optimal solution must also be optimal for the subproblems "standing alone":
  - If parenthisization of $A_{i..k}$ was not optimal we could replace it by a better parenthesization and get a cheaper final solution, leading to a contradiction.
  - Similarly, if parenthisization of $A_{k+1..j}$ was not optimal we could replace it by a better parenthesization and get a cheaper final solution, also leading to a contradiction.

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 2: Recursively define the value of optimal solution.

- For $1 \leq i \leq j \leq n$, let $m[i,j]$ denote the minimum number of multiplications needed to compute $A_{i..j}$. The optimum cost can be described by the following recursive definition.

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

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

```
MATRIX − CHAIN − ORDER(p)
n ← length[p] − 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]
                then m[i, j] ← q
                     s[i, j] ← k
return m and s
```

Lets illustrate the example with the help of an example.

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

$$\langle 30, 35, 15, 5, 10, 20, 25 \rangle$$

Solution:

Here

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

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

$$\langle 30, 35, 15, 5, 10, 20, 25 \rangle$$

Solution:

Here

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

Matrix	A_1	A_2	A_3	A_4	A_5	A_6
Dimensions	30 x 35	35 x 15	15 x 5	5 x 10	10 x 20	20 x 25

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0					
2		0				
3			0			
4				0		
5					0	
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1					
2					
3					
4					
5					

s matrix

```
n ← length[p] − 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
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	∞				
2		0				
3			0			
4				0		
5					0	
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1					
2					
3					
4					
5					

s matrix

$$
\begin{array}{l}
for\ l \leftarrow 2\ to\ n \\
\quad for\ i \leftarrow 1\ to\ n - l + 1 \\
\qquad j \leftarrow i + l - 1 \\
\qquad m[i,j] \leftarrow \infty \\
\qquad for\ k \leftarrow i\ to\ j - 1 \\
\qquad\quad q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_kp_j \\
\qquad\quad if\ q < m[i,j] \\
\qquad\qquad then\ m[i,j] \leftarrow q \\
\qquad\qquad\quad s[i,j] \leftarrow k
\end{array}
$$

$l = 2\ , i = 1, j = 2, k = 1$

$q =$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750				
2		0				
3			0			
4				0		
5					0	
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1				
2					
3					
4					
5					

s matrix

$for\ l \leftarrow 2\ to\ n$

$\quad for\ i \leftarrow 1\ to\ n - l + 1$

$\quad\quad j \leftarrow i + l - 1$

$\quad\quad m[i,j] \leftarrow \infty$

$\quad\quad for\ k \leftarrow i\ to\ j - 1$

$\quad\quad\quad q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_k p_j$

$\quad\quad\quad if\ q < m[i,j]$

$\quad\quad\quad\quad then\ m[i,j] \leftarrow q$

$\quad\quad\quad\quad\quad s[i,j] \leftarrow k$

$l = 2\ , i = 1, j = 2, k = 1$

$q = 0 + 0 + 30 * 35 * 15 = 15750$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750				
2		0	∞			
3			0			
4				0		
5					0	
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1				
2					
3					
4					
5					

s matrix

$for\ l \leftarrow 2\ to\ n$
$\quad for\ i \leftarrow 1\ to\ n - l + 1$
$\quad\quad j \leftarrow i + l - 1$
$\quad\quad m[i,j] \leftarrow \infty$
$\quad\quad for\ k \leftarrow i\ to\ j - 1$
$\quad\quad\quad q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_kp_j$
$\quad\quad\quad if\ q < m[i,j]$
$\quad\quad\quad\quad then\ m[i,j] \leftarrow q$
$\quad\quad\quad\quad\quad s[i,j] \leftarrow k$

$l = 2\ , i = 2, j = 3, k = 2$

$q =$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750				
2		0	2625			
3			0			
4				0		
5					0	
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1				
2		2			
3					
4					
5					

s matrix

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

$l = 2\ , i = 2, j = 3, k = 2$

$q = 0 + 0 + 30 * 15 * 5 = 2625$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750				
2		0	2625			
3			0	∞		
4				0		
5					0	
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1				
2		2			
3					
4					
5					

s matrix

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

$l = 2\ , i = 3, j = 4, k = 3$

$q =$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750				
2		0	2625			
3			0	750		
4				0		
5					0	
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1				
2		2			
3			3		
4					
5					

s matrix

$for\ l \leftarrow 2\ to\ n$
$\quad for\ i \leftarrow 1\ to\ n - l + 1$
$\quad\quad j \leftarrow i + l - 1$
$\quad\quad m[i,j] \leftarrow \infty$
$\quad\quad for\ k \leftarrow i\ to\ j - 1$
$\quad\quad\quad q \leftarrow m[i,k] + m[k + 1,j] + p_{i-1}p_k p_j$
$\quad\quad\quad if\ q < m[i,j]$
$\quad\quad\quad\quad then\ m[i,j] \leftarrow q$
$\quad\quad\quad\quad\quad s[i,j] \leftarrow k$

$l = 2\ , i = 3, j = 4, k = 3$

$q = 0 + 0 + 15 * 5 * 10 = 750$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750				
2		0	2625			
3			0	750		
4				0	∞	
5					0	
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1				
2		2			
3			3		
4					
5					

s matrix

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

$l = 2 \ , i = 4, j = 5, k = 4$

$q =$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750				
2		0	2625			
3			0	750		
4				0	1000	
5					0	
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1				
2		2			
3			3		
4				4	
5					

s matrix

$for\ l \leftarrow 2\ to\ n$
$\quad for\ i \leftarrow 1\ to\ n - l + 1$
$\quad\quad j \leftarrow i + l - 1$
$\quad\quad m[i,j] \leftarrow \infty$
$\quad\quad for\ k \leftarrow i\ to\ j - 1$
$\quad\quad\quad q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_kp_j$
$\quad\quad\quad if\ q < m[i,j]$
$\quad\quad\quad\quad then\ m[i,j] \leftarrow q$
$\quad\quad\quad\quad\quad s[i,j] \leftarrow k$

$l = 2\ , i = 4, j = 5, k = 4$

$q = 0 + 0 + 5 * 10 * 20 = 1000$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750				
2	A_1	0	2625			
3		A_2	0	750		
4			A_3	0	1000	
5				A_4	0	∞
6					A_5	0

A_6

m matrix

	2	3	4	5	6
1	1				
2		2			
3			3		
4				4	
5					

s matrix

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                     s[i,j] ← k
```

$l = 2\ , i = 4, j = 5, k = 4$

$q =$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750				
2		0	2625			
3			0	750		
4				0	1000	
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1				
2		2			
3			3		
4				4	
5					5

s matrix

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                     s[i,j] ← k
```

$l = 2\ , i = 4, j = 5, k = 4$

$q = 0 + 0 + 10 * 20 * 25 = 5000$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	∞			
2	A_1	0	2625			
3		A_2	0	750		
4			A_3	0	1000	
5				A_4	0	5000
6					A_5	0
						A_6

m matrix

	2	3	4	5	6
1	1				
2		2			
3			3		
4				4	
5					5

s matrix

$for\ l \leftarrow 2\ to\ n$
$\quad for\ i \leftarrow 1\ to\ n - l + 1$
$\qquad j \leftarrow i + l - 1$
$\qquad m[i,j] \leftarrow \infty$
$\qquad for\ k \leftarrow i\ to\ j - 1$
$\qquad\quad q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_kp_j$
$\qquad\quad if\ q < m[i,j]$
$\qquad\qquad then\ m[i,j] \leftarrow q$
$\qquad\qquad\quad s[i,j] \leftarrow k$

$l = 3\ \ , i = 1, j = 3, k = 1$

$q = 0 + 2625 + 30 * 35 * 5 = 7875$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	∞			
2		0	2625			
3			0	750		
4				0	1000	
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1				
2		2			
3			3		
4				4	
5					5

s matrix

$for\ l \leftarrow 2\ to\ n$
$\quad for\ i \leftarrow 1\ to\ n - l + 1$
$\qquad j \leftarrow i + l - 1$
$\qquad m[i,j] \leftarrow \infty$
$\qquad for\ k \leftarrow i\ to\ j - 1$
$\qquad\quad q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_kp_j$
$\qquad\quad if\ q < m[i,j]$
$\qquad\qquad then\ m[i,j] \leftarrow q$
$\qquad\qquad\quad s[i,j] \leftarrow k$

$l = 3\ , i = 1, j = 3, k = 1$
$q = 0 + 2625 + 30 * 35 * 5 = 7875$

$l = 3\ , i = 1, j = 3, k = 2$
$q = 15750 + 0 + 30 * 15 * 5 = 18000$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875			
2		0	2625			
3			0	750		
4				0	1000	
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1			
2		2			
3			3		
4				4	
5					5

s matrix

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

$l = 3\ , i = 1, j = 3, k = 1$

$q = 0 + 2625 + 30 * 35 * 5 = 7875$ min

$l = 3\ , i = 1, j = 3, k = 2$

$q = 15750 + 0 + 30 * 15 * 5 = 18000$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875			
2		0	2625	∞		
3			0	750		
4				0	1000	
5					0	5000
6						0

m matrix

	2	3	4	5	6
1	1	1			
2		2			
3			3		
4				4	
5					5

s matrix

$l = 3 \ , i = 2, j = 4, k = 2$

$q = 0 + 750 + 35 * 15 * 10 = 6000$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                     s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875			
2		0	2625	∞		
3			0	750		
4				0	1000	
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1			
2		2	3		
3			3		
4				4	
5					5

s matrix

$$for\ l \leftarrow 2\ to\ n$$
$$for\ i \leftarrow 1\ to\ n-l+1$$
$$j \leftarrow i+l-1$$
$$m[i,j] \leftarrow \infty$$
$$for\ k \leftarrow i\ to\ j-1$$
$$q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_kp_j$$
$$if\ q < m[i,j]$$
$$then\ m[i,j] \leftarrow q$$
$$s[i,j] \leftarrow k$$

$l = 3\ ,i = 2, j = 4, k = 2$

$q = 0 + 750 + 35 * 15 * 10 = 6000$

$l = 3\ ,i = 2, j = 4, k = 3$

$q = 2625 + 0 + 35 * 5 * 10 = 4375$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875			
2		0	2625	4375		
3			0	750		
4				0	1000	
5					0	5000
6						0

m matrix

	2	3	4	5	6
1	1	1			
2		2	3		
3			3		
4				4	
5					5

s matrix

$$
\begin{array}{l}
\textit{for } l \leftarrow 2 \textit{ to } n \\
\quad \textit{for } i \leftarrow 1 \textit{ to } n - l + 1 \\
\quad\quad j \leftarrow i + l - 1 \\
\quad\quad m[i,j] \leftarrow \infty \\
\quad\quad \textit{for } k \leftarrow i \textit{ to } j - 1 \\
\quad\quad\quad q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_k p_j \\
\quad\quad\quad \textit{if } q < m[i,j] \\
\quad\quad\quad\quad \textit{then } m[i,j] \leftarrow q \\
\quad\quad\quad\quad\quad s[i,j] \leftarrow k
\end{array}
$$

$l = 3\ , i = 2, j = 4, k = 2$

$q = 0 + 750 + 35 * 15 * 10 = 6000$

$l = 3\ , i = 2, j = 4, k = 3$

$q = 2625 + 0 + 35 * 5 * 10 = 4375$ min

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875			
2	A_1	0	2625	4375		
3		A_2	0	750	∞	
4			A_3	0	1000	
5				A_4	0	5000
6					A_5	0

A_6

m matrix

	2	3	4	5	6
1	1	1			
2		2	3		
3			3		
4				4	
5					5

s matrix

$l = 3\ , i = 3, j = 5, k = 3$

$q = 0 + 1000 + 15 * 5 * 20 = 2500$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                     s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875			
2	A_1	0	2625	4375		
3		A_2	0	750	∞	
4			A_3	0	1000	
5				A_4	0	5000
6					A_5	0
						A_6

m matrix

	2	3	4	5	6
1	1	1			
2		2	3		
3			3		
4				4	
5					5

s matrix

$for\ l \leftarrow 2\ to\ n$
$\quad for\ i \leftarrow 1\ to\ n - l + 1$
$\qquad j \leftarrow i + l - 1$
$\qquad m[i,j] \leftarrow \infty$
$\qquad for\ k \leftarrow i\ to\ j - 1$
$\qquad\quad q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_kp_j$
$\qquad\quad if\ q < m[i,j]$
$\qquad\qquad then\ m[i,j] \leftarrow q$
$\qquad\qquad\quad s[i,j] \leftarrow k$

$l = 3\ , i = 3, j = 5, k = 3$
$q = 0 + 1000 + 15 * 5 * 20 = 2500$

$l = 3\ , i = 3, j = 5, k = 4$
$q = 750 + 0 + 15 * 10 * 20 = 3750$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875			
2		0	2625	4375		
3			0	750	2500	
4				0	1000	
5					0	5000
6						0

m matrix

	2	3	4	5	6
1	1	1			
2		2	3		
3			3	3	
4				4	
5					5

s matrix

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

min

$l = 3 \;, i = 3, j = 5, k = 3$

$q = 0 + 1000 + 15 * 5 * 20 = 2500$

$l = 3 \;, i = 3, j = 5, k = 4$

$q = 750 + 0 + 15 * 10 * 20 = 3750$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875			
2		0	2625	4375		
3			0	750	2500	
4				0	1000	∞
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1			
2		2	3		
3			3		
4				4	
5					5

s matrix

$$
\begin{array}{l}
\textit{for } l \leftarrow 2 \textit{ to } n \\
\quad \textit{for } i \leftarrow 1 \textit{ to } n - l + 1 \\
\qquad j \leftarrow i + l - 1 \\
\qquad m[i,j] \leftarrow \infty \\
\qquad \textit{for } k \leftarrow i \textit{ to } j - 1 \\
\qquad\quad q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_kp_j \\
\qquad\quad \textit{if } q < m[i,j] \\
\qquad\qquad \textit{then } m[i,j] \leftarrow q \\
\qquad\qquad\quad s[i,j] \leftarrow k
\end{array}
$$

$l = 3\ , i = 4, j = 6, k = 4$

$q = 0 + 5000 + 5 * 10 * 25 = 6250$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875			
2		0	2625	4375		
3			0	750	2500	
4				0	1000	∞
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1			
2		2	3		
3			3		
4				4	
5					5

s matrix

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                     s[i,j] ← k
```

$l = 3\ , i = 4, j = 6, k = 4$

$q = 0 + 5000 + 5 * 10 * 25 = 6250$

$l = 3\ , i = 4, j = 6, k = 5$

$q = 1000 + 0 + 5 * 20 * 25 = 3500$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875			
2		0	2625	4375		
3			0	750	2500	
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1			
2		2	3		
3			3	3	
4				4	5
5					5

s matrix

$$
\begin{array}{l}
for\ l \leftarrow 2\ to\ n \\
\quad for\ i \leftarrow 1\ to\ n - l + 1 \\
\qquad j \leftarrow i + l - 1 \\
\qquad m[i,j] \leftarrow \infty \\
\qquad for\ k \leftarrow i\ to\ j - 1 \\
\qquad\quad q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_kp_j \\
\qquad\quad if\ q < m[i,j] \\
\qquad\qquad then\ m[i,j] \leftarrow q \\
\qquad\qquad\quad s[i,j] \leftarrow k
\end{array}
$$

$l = 3\ , i = 4, j = 6, k = 4$

$q = 0 + 5000 + 5 * 10 * 25 = 6250$

$l = 3\ , i = 4, j = 6, k = 5$

$q = 1000 + 0 + 5 * 20 * 25 = 3500$ min

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	∞		
2		0	2625	4375		
3			0	750	2500	
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1			
2		2	3		
3			3	3	
4				4	5
5					5

s matrix

$l = 3\ , i = 1, j = 4$

$k = 1, q = 0 + 4375 + 30 * 35 * 10 = 14875$

$$
\begin{array}{l}
for\ l \leftarrow 2\ to\ n \\
\quad for\ i \leftarrow 1\ to\ n - l + 1 \\
\quad\quad j \leftarrow i + l - 1 \\
\quad\quad m[i,j] \leftarrow \infty \\
\quad\quad for\ k \leftarrow i\ to\ j - 1 \\
\quad\quad\quad q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_k p_j \\
\quad\quad\quad if\ q < m[i,j] \\
\quad\quad\quad\quad then\ m[i,j] \leftarrow q \\
\quad\quad\quad\quad\quad s[i,j] \leftarrow k
\end{array}
$$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	∞		
2		0	2625	4375		
3			0	750	2500	
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1			
2		2	3		
3			3	3	
4				4	5
5					5

s matrix

$l = 3\ , i = 1, j = 4$

$k = 1, q = 0 + 4375 + 30 * 35 * 10 = 14875$

$k = 2, q = 15750 + 750 + 30 * 15 * 10 = 18000$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                     s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375		
2		0	2625	4375		
3			0	750	2500	
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3		
2		2	3		
3			3	3	
4				4	5
5					5

s matrix

$l = 3\ , i = 1, j = 4$

$k = 1, q = 0 + 4375 + 30 * 35 * 10 = 14875$

$k = 2, q = 15750 + 750 + 30 * 15 * 10 = 18000$

$k = 3, q = 7875 + 0 + 30 * 5 * 10 = 9375$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375		
2		0	2625	4375		
3			0	750	2500	
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3		
2		2	3		
3			3	3	
4				4	5
5					5

s matrix

$l = 3 \ , i = 1, j = 4$

$k = 1, q = 0 + 4375 + 30 * 35 * 10 = 14875$

$k = 2, q = 15750 + 750 + 30 * 15 * 10 = 18000$ min

$k = 3, q = 7875 + 0 + 30 * 5 * 10 = 9375$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375		
2		0	2625	4375	∞	
3			0	750	2500	
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3		
2		2	3		
3			3	3	
4				4	5
5					5

s matrix

$l = 3\ , i = 2, j = 5$

$k = 2, q = 0 + 4375 + 35 * 15 * 20 = 10500$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375		
2		0	2625	4375	∞	
3			0	750	2500	
4				0	1000	3500
5					0	5000
6						0

m matrix

	2	3	4	5	6
1	1	1	3		
2		2	3		
3			3	3	
4				4	5
5					5

s matrix

$l = 3 \; , i = 2, j = 5$

$k = 2, q = 0 + 4375 + 35 * 15 * 20 = 10500$

$k = 3, q = 2625 + 1000 + 35 * 5 * 20 = 7125$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375		
2		0	2625	4375	∞	
3			0	750	2500	
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3		
2		2	3		
3			3	3	
4				4	5
5					5

s matrix

$l = 3\ , i = 2, j = 5$

$k = 2, q = 0 + 4375 + 35 * 15 * 20 = 10500$

$k = 3, q = 2625 + 1000 + 35 * 5 * 20 = 7125$

$k = 4, q = 4375 + 0 + 35 * 10 * 20 = 11375$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375		
2		0	2625	4375	7125	
3			0	750	2500	
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3		
2		2	3	3	
3			3	3	
4				4	5
5					5

s matrix

$l = 3\ , i = 2, j = 5$

$k = 2, q = 0 + 4375 + 35 * 15 * 20 = 10500$

$k = 3, q = 2625 + 1000 + 35 * 5 * 20 = 7125$ min

$k = 4, q = 4375 + 0 + 35 * 10 * 20 = 11375$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                     s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375		
2		0	2625	4375	7125	
3			0	750	2500	∞
4				0	1000	3500
5					0	5000
6						0

m matrix

	2	3	4	5	6
1	1	1	3		
2		2	3	3	
3			3	3	
4				4	5
5					5

s matrix

$l = 3 \ , i = 3, j = 6$

$k = 3, q = 0 + 3500 + 15 * 5 * 25 = 5375$

$for\ l \leftarrow 2\ to\ n$
  $for\ i \leftarrow 1\ to\ n - l + 1$
    $j \leftarrow i + l - 1$
    $m[i,j] \leftarrow \infty$
    $for\ k \leftarrow i\ to\ j - 1$
      $q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_kp_j$
      $if\ q < m[i,j]$
        $then\ m[i,j] \leftarrow q$
          $s[i,j] \leftarrow k$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375		
2		0	2625	4375	7125	
3			0	750	2500	∞
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3		
2		2	3	3	
3			3	3	
4				4	5
5					5

s matrix

$l = 3\ , i = 3, j = 6$

$k = 3, q = 0 + 3500 + 15 * 5 * 25 = 5375$

$k = 4, q = 750 + 5000 + 15 * 10 * 25 = 9500$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                     s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375		
2	A_1	0	2625	4375	7125	
3		A_2	0	750	2500	∞
4			A_3	0	1000	3500
5				A_4	0	5000
6					A_5	0
						A_6

m matrix

	2	3	4	5	6
1	1	1	3		
2		2	3	3	
3			3	3	
4				4	5
5					5

s matrix

$l = 3\ , i = 3, j = 6$

$k = 3, q = 0 + 3500 + 15 * 5 * 25 = 5375$

$k = 4, q = 750 + 5000 + 15 * 10 * 25 = 9500$

$k = 5, q = 2500 + 0 + 15 * 20 * 25 = 10000$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375		
2		0	2625	4375	7125	
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3		
2		2	3	3	
3			3	3	3
4				4	5
5					5

s matrix

$l = 3 \ , i = 3, j = 6$

$k = 3, q = 0 + 3500 + 15 * 5 * 25 = 5375$ min

$k = 4, q = 750 + 5000 + 15 * 10 * 25 = 9500$

$k = 5, q = 2500 + 0 + 15 * 20 * 25 = 10000$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375	∞	
2		0	2625	4375	7125	
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3		
2		2	3	3	
3			3	3	3
4				4	5
5					5

s matrix

$l = 4\ , i = 1, j = 5$

$k = 1, q = 0 + 7125 + 30 * 35 * 20 = 28125$

$for\ l \leftarrow 2\ to\ n$
$\quad for\ i \leftarrow 1\ to\ n - l + 1$
$\quad\quad j \leftarrow i + l - 1$
$\quad\quad m[i,j] \leftarrow \infty$
$\quad\quad for\ k \leftarrow i\ to\ j - 1$
$\quad\quad\quad q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_k p_j$
$\quad\quad\quad if\ q < m[i,j]$
$\quad\quad\quad\quad then\ m[i,j] \leftarrow q$
$\quad\quad\quad\quad\quad s[i,j] \leftarrow k$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375	∞	
2		0	2625	4375	7125	
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3		
2		2	3	3	
3			3	3	3
4				4	5
5					5

s matrix

$l = 4\ , i = 1, j = 5$

$k = 1, q = 0 + 7125 + 30 * 35 * 20 = 28125$

$k = 2, q = 15750 + 2500 + 30 * 15 * 20 = 27250$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375	∞	
2		0	2625	4375	7125	
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3		
2		2	3	3	
3			3	3	3
4				4	5
5					5

s matrix

$l = 4\ , i = 1, j = 5$

$k = 1, q = 0 + 7125 + 30 * 35 * 20 = 28125$

$k = 2, q = 15750 + 2500 + 30 * 15 * 20 = 27250$

$k = 3, q = 7875 + 1000 + 30 * 5 * 20 = 11875$

$for\ l \leftarrow 2\ to\ n$

$\quad for\ i \leftarrow 1\ to\ n - l + 1$

$\quad\quad j \leftarrow i + l - 1$

$\quad\quad m[i,j] \leftarrow \infty$

$\quad\quad for\ k \leftarrow i\ to\ j - 1$

$\quad\quad\quad q \leftarrow m[i,k] + m[k + 1, j] + p_{i-1}p_kp_j$

$\quad\quad\quad if\ q < m[i,j]$

$\quad\quad\quad\quad then\ m[i,j] \leftarrow q$

$\quad\quad\quad\quad\quad s[i,j] \leftarrow k$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375	∞	
2		0	2625	4375	7125	
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3		
2		2	3	3	
3			3	3	3
4				4	5
5					5

s matrix

$l = 4\ , i = 1, j = 5$

$k = 1, q = 0 + 7125 + 30 * 35 * 20 = 28125$

$k = 2, q = 15750 + 2500 + 30 * 15 * 20 = 27250$

$k = 3, q = 7875 + 1000 + 30 * 5 * 20 = 11875$

$k = 4, q = 9375 + 0 + 30 * 10 * 20 = 15375$

$for\ l \leftarrow 2\ to\ n$

$\quad for\ i \leftarrow 1\ to\ n - l + 1$

$\quad\quad j \leftarrow i + l - 1$

$\quad\quad m[i,j] \leftarrow \infty$

$\quad\quad for\ k \leftarrow i\ to\ j - 1$

$\quad\quad\quad q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_kp_j$

$\quad\quad\quad if\ q < m[i,j]$

$\quad\quad\quad\quad then\ m[i,j] \leftarrow q$

$\quad\quad\quad\quad\quad s[i,j] \leftarrow k$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375	11875	
2		0	2625	4375	7125	
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3	3	
2		2	3	3	
3			3	3	3
4				4	5
5					5

s matrix

$l = 4\ , i = 1, j = 5$

$k = 1, q = 0 + 7125 + 30 * 35 * 20 = 28125$

$k = 2, q = 15750 + 2500 + 30 * 15 * 20 = 27250$

$k = 3, q = 7875 + 1000 + 30 * 5 * 20 = 11875$ min

$k = 4, q = 9375 + 0 + 30 * 10 * 20 = 15375$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                     s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375	11875	
2		0	2625	4375	7125	∞
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3	3	
2		2	3	3	
3			3	3	3
4				4	5
5					5

s matrix

$l = 4\ , i = 2, j = 6$

$k = 2, q = 0 + 5375 + 35 * 15 * 25 = 15375$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375	11875	
2		0	2625	4375	7125	∞
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3	3	
2		2	3	3	
3			3	3	3
4				4	5
5					5

s matrix

$l = 4 \ , i = 2, j = 6$

$k = 2, q = 0 + 5375 + 35 * 15 * 25 = 15375$

$k = 3, q = 2625 + 3500 + 35 * 5 * 25 = 10500$

$$
\begin{array}{l}
for\ l \leftarrow 2\ to\ n \\
\quad for\ i \leftarrow 1\ to\ n - l + 1 \\
\qquad j \leftarrow i + l - 1 \\
\qquad m[i,j] \leftarrow \infty \\
\qquad for\ k \leftarrow i\ to\ j - 1 \\
\qquad\quad q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_kp_j \\
\qquad\quad if\ q < m[i,j] \\
\qquad\qquad then\ m[i,j] \leftarrow q \\
\qquad\qquad\quad s[i,j] \leftarrow k
\end{array}
$$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375	11875	
2		0	2625	4375	7125	∞
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3	3	
2		2	3	3	
3			3	3	3
4				4	5
5					5

s matrix

$l = 4\ , i = 2, j = 6$

$k = 2, q = 0 + 5375 + 35 * 15 * 25 = 15375$

$k = 3, q = 2625 + 3500 + 35 * 5 * 25 = 10500$

$k = 4, q = 4375 + 5000 + 35 * 10 * 25 = 18125$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375	11875	
2		0	2625	4375	7125	∞
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3	3	
2		2	3	3	
3			3	3	3
4				4	5
5					5

s matrix

$l = 4\ , i = 2, j = 6$

$k = 2, q = 0 + 5375 + 35 * 15 * 25 = 15375$

$k = 3, q = 2625 + 3500 + 35 * 5 * 25 = 10500$

$k = 4, q = 4375 + 5000 + 35 * 10 * 25 = 18125$

$k = 5, q = 7125 + 0 + 35 * 20 * 25 = 24625$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                     s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

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

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3	3	
2		2	3	3	3
3			3	3	3
4				4	5
5					5

s matrix

$l = 4\ , i = 2, j = 6$

$k = 2, q = 0 + 5375 + 35 * 15 * 25 = 15375$

$k = 3, q = 2625 + 3500 + 35 * 5 * 25 = 10500$ min

$k = 4, q = 4375 + 5000 + 35 * 10 * 25 = 18125$

$k = 5, q = 7125 + 0 + 35 * 20 * 25 = 24625$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375	11875	∞
2		0	2625	4375	7125	10500
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3	3	
2		2	3	3	3
3			3	3	3
4				4	5
5					5

s matrix

$l = 4\ , i = 1, j = 6$

$k = 1, q = 0 + 10500 + 30 * 35 * 25 = 36750$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375	11875	∞
2		0	2625	4375	7125	10500
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3	3	
2		2	3	3	3
3			3	3	3
4				4	5
5					5

s matrix

$l = 4 \ , i = 1, j = 6$

$k = 1, q = 0 + 10500 + 30 * 35 * 25 = 36750$

$k = 2, q = 15750 + 5375 + 30 * 15 * 25 = 32375$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375	11875	∞
2		0	2625	4375	7125	10500
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3	3	
2		2	3	3	3
3			3	3	3
4				4	5
5					5

s matrix

$l = 4\ ,i = 1, j = 6$

$k = 1, q = 0 + 10500 + 30 * 35 * 25 = 36750$

$k = 2, q = 15750 + 5375 + 30 * 15 * 25 = 32375$

$k = 3, q = 7875 + 3500 + 30 * 5 * 25 = 15125$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                    s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375	11875	∞
2		0	2625	4375	7125	10500
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3	3	
2		2	3	3	3
3			3	3	3
4				4	5
5					5

s matrix

$l = 4\ , i = 1, j = 6$

$k = 1, q = 0 + 10500 + 30 * 35 * 25 = 36750$

$k = 2, q = 15750 + 5375 + 30 * 15 * 25 = 32375$

$k = 3, q = 7875 + 3500 + 30 * 5 * 25 = 15125$

$k = 4, q = 9375 + 5000 + 30 * 10 * 25 = 21875$

$for\ l \leftarrow 2\ to\ n$

$\quad for\ i \leftarrow 1\ to\ n - l + 1$

$\quad\quad j \leftarrow i + l - 1$

$\quad\quad m[i,j] \leftarrow \infty$

$\quad\quad for\ k \leftarrow i\ to\ j - 1$

$\quad\quad\quad q \leftarrow m[i,k] + m[k+1,j] + p_{i-1}p_kp_j$

$\quad\quad\quad if\ q < m[i,j]$

$\quad\quad\quad\quad then\ m[i,j] \leftarrow q$

$\quad\quad\quad\quad\quad s[i,j] \leftarrow k$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	1	2	3	4	5	6
1	0	15750	7875	9375	11875	∞
2		0	2625	4375	7125	10500
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	2	3	4	5	6
1	1	1	3	3	
2		2	3	3	3
3			3	3	3
4				4	5
5					5

s matrix

$$l = 4 \;, i = 1, j = 6$$

$$k = 1, q = 0 + 10500 + 30 * 35 * 25 = 36750$$
$$k = 2, q = 15750 + 5375 + 30 * 15 * 25 = 32375$$
$$k = 3, q = 7875 + 3500 + 30 * 5 * 25 = 15125$$
$$k = 4, q = 9375 + 5000 + 30 * 10 * 25 = 21875$$
$$k = 5, q = 11875 + 0 + 30 * 20 * 25 = 26875$$

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                     s[i,j] ← k
```

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

Solution:

	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

A_1 A_2 A_3 A_4 A_5 A_6

m matrix

	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

s matrix

```
for l ← 2 to n
    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]
                then m[i,j] ← q
                     s[i,j] ← k
```

$l = 4\ , i = 1, j = 6$

$k = 1, q = 0 + 10500 + 30 * 35 * 25 = 36750$

$k = 2, q = 15750 + 5375 + 30 * 15 * 25 = 32375$

$k = 3, q = 7875 + 3500 + 30 * 5 * 25 = 15125$ min

$k = 4, q = 9375 + 5000 + 30 * 10 * 25 = 21875$

$k = 5, q = 11875 + 0 + 30 * 20 * 25 = 26875$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 3: Compute optimal cost.

$MATRIX-CHAIN-ORDER(p)$

```
n ← length[p] − 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]
                then m[i, j] ← q
                     s[i, j] ← k
return m and s
```

A simple inspection of the nested loop structure of MATRIX-CHAIN-ORDER yields a running time of $O(n^3)$ for the algorithm. The loops are nested three deep, and each loop index $(l, i, and\ k)$ takes on at most $n - 1$ values.

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 4: Construct / print the optimal solution.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

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

$if\ i\ =\ j$

$\quad then\quad print\ "A_i"$

$\quad else\quad print\ "("$

$\qquad\qquad PRINT-OPTIMAL-PARENS(s,i,s[i,j])$

$\qquad\qquad PRINT-OPTIMAL-PARENS(s,s[i,j]\ +\ 1,j)$

$\qquad\qquad print\ ")"$

Lets see, how in the discussed example the call $PRINT-OPTIMAL-PARENS(s,1,6)$ prints the parenthesization $((A1\ (A2\ A3))\ ((A4\ A5)A6))$.

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 4: Construct / print the optimal solution.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

$if\ i = j$

$\quad then \quad print\ "A_i"$

$\quad else \quad print\ "("$

$\qquad\qquad PRINT - OPTIMAL - PARENS(s, i, s[i, j])$

$\qquad\qquad PRINT - OPTIMAL - PARENS(s, s[i, j] + 1, j)$

$\qquad\qquad print\ ")"$

	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

s matrix

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 4: Construct / print the optimal solution.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

```
PRINT − OPTIMAL − PARENS(s, i, j)
if i = j
   then print "A_i"
   else print "("
        PRINT − OPTIMAL − PARENS(s, i, s[i, j])
        PRINT − OPTIMAL − PARENS(s, s[i, j] + 1, j)
        print ")"
```

POP(S,1,6)

	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

s matrix

Dynamic Programming

Problem 4: Matrix Chain Multiplication

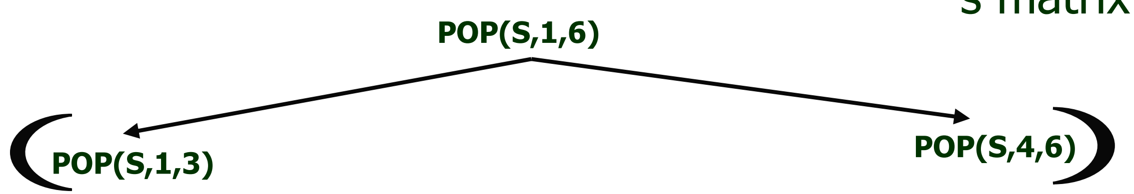
Step 4: Construct / print the optimal solution.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

	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

s matrix



Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 4: Construct / print the optimal solution.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

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

$if\ i\ =\ j$

$\quad then\quad print\ "A_i"$

$\quad else\quad print\ "("$

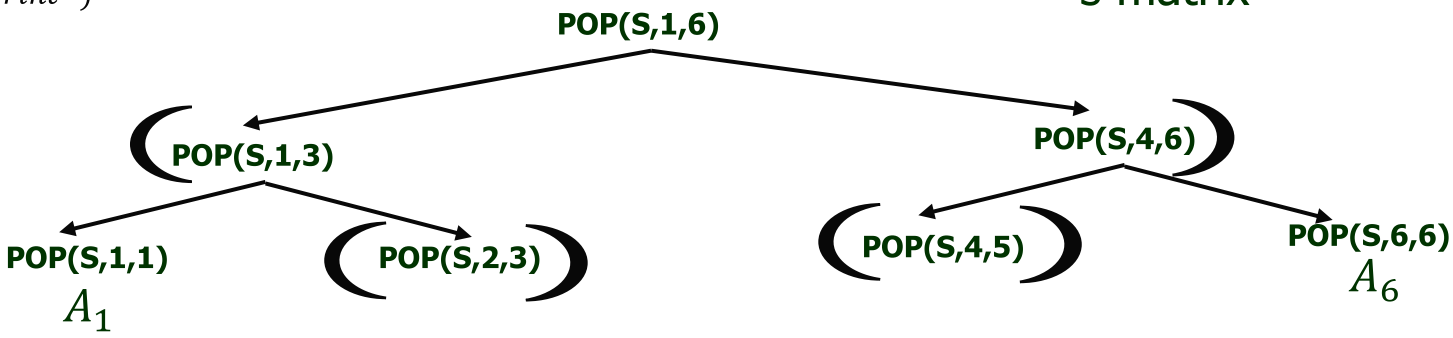
$\qquad\qquad PRINT-OPTIMAL-PARENS(s,i,s[i,j])$

$\qquad\qquad PRINT-OPTIMAL-PARENS(s,s[i,j]\ +\ 1,j)$

$\qquad\qquad print\ ")"$

	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

s matrix



Dynamic Programming

Problem 4: Matrix Chain Multiplication

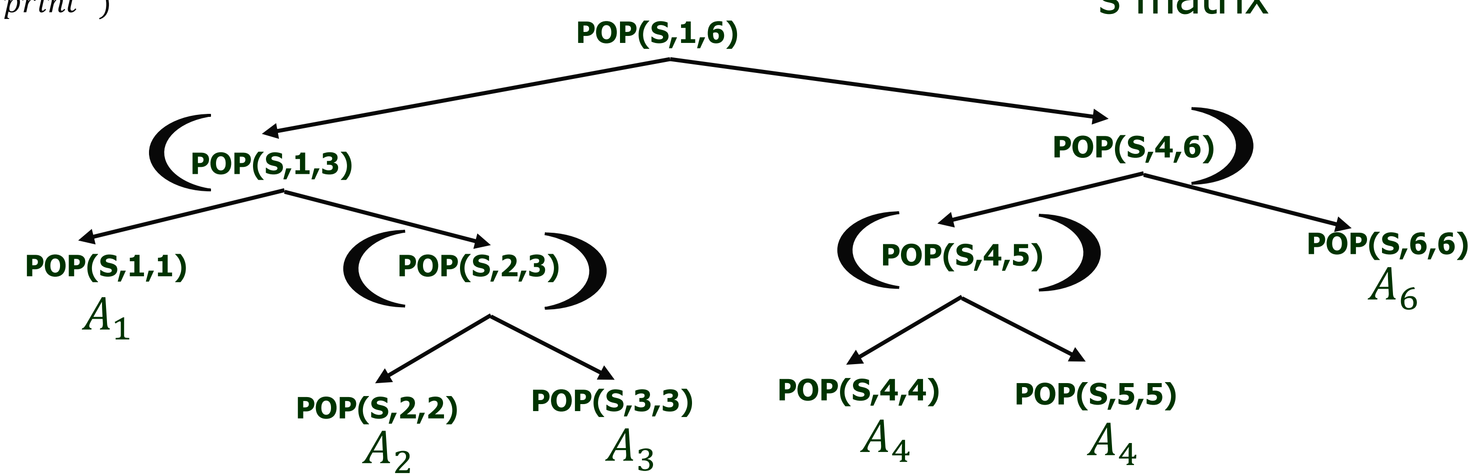
Step 4: Construct / print the optimal solution.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

```
PRINT − OPTIMAL − PARENS(s, i, j)
if i = j
   then  print "A_i"
   else  print "("
         PRINT − OPTIMAL − PARENS(s, i, s[i, j])
         PRINT − OPTIMAL − PARENS(s, s[i, j] + 1, j)
         print ")"
```

	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

s matrix



$$((A1\ (A2\ A3))\ ((A4\ A5)\ A6))$$

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Step 4: Construct / print the optimal solution.

Example 1: Find an optimal parenthesization of a matrix chain product whose sequence of dimension is :

$$
\begin{array}{ll}
\multicolumn{2}{l}{PRINT-OPTIMAL-PARENS(s,i,j)} \\
\multicolumn{2}{l}{if\ i = j} \\
then & print\ "A_i" \\
else & print\ "(" \\
 & PRINT-OPTIMAL-PARENS(s,i,s[i,j]) \\
 & PRINT-OPTIMAL-PARENS(s,s[i,j]+1,j) \\
 & print\ ")"
\end{array}
$$

	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

s matrix

POP(s, 1, 6), s[1, 6] = 3, (A1A2A3)(A4A5A6)

POP(s, 1, 3), s[1, 3] = 1, ((A1)(A2A3))(A4A5A6)

POP(s, 4, 6), s[4, 6] = 5, ((A1)(A2A3))((A4A5)(A6))

POP(s, 2, 3), s[2, 3] = 2, ((A1)((A2)(A3)))((A4A5)(A6))

POP(s, 4, 5), s[4, 5] = 4, ((A1)((A2)(A3)))(((A4)(A5))(A6))

Hence the product is computed as follows

(A1(A2A3))((A4A5)A6).

Dynamic Programming

Problem 4: Matrix Chain Multiplication

Example 2: Find an optimal parenthesization of a matrix-chain product whose sequence of dimensions is $\langle 5, 10, 3, 12, 5, 50, 6 \rangle$

Example 3: Find an optimal parenthesization of a matrix-chain product whose sequence of dimensions is $\langle 5, 4, 6, 2, 7 \rangle$

.

Self practice

Thank u