

Design and Analysis of Algorithms

Part II: Dynamic Programming

Lecture 12: Longest Common Subsequence

盛浩

shenghao@buaa.edu.cn

北京航空航天大学
计算机学院

北航《算法设计与分析》



动态规划篇概述

- 在算法课程第二部分“动态规划”主题中，我们将主要聚焦于如下经典问题：
 - 0-1 Knapsack (0-1背包问题)
 - Maximum Contiguous Subarray II (最大连续子数组 II)
 - Longest Common Subsequences (最长公共子序列)
 - Longest Common Substrings (最长公共子串)
 - Minimum Edit Distance (最小编辑距离)
 - Rod-Cutting (钢条切割)
 - Chain Matrix Multiplication (矩阵链乘法)



动态规划篇概述

- 在算法课程第二部分“动态规划”主题中，我们将主要聚焦于如下经典问题：
 - 0-1 Knapsack (0-1背包问题)
 - Maximum Contiguous Subarray II (最大连续子数组 II)
 - Longest Common Subsequences (最长公共子序列)
 - Longest Common Substrings (最长公共子串)
 - Minimum Edit Distance (最小编辑距离)
 - Rod-Cutting (钢条切割)
 - Chain Matrix Multiplication (矩阵链乘法)



问题背景：子序列

- 子序列
 - 将给定序列中零个或多个元素（如字符）去掉后所得结果



问题背景：子序列

- 子序列
 - 将给定序列中零个或多个元素（如字符）去掉后所得结果
- 示例
 - 给定序列 X

X	A	B	C	B	D	A	B
-----	-----	-----	-----	-----	-----	-----	-----



问题背景：子序列

- 子序列
 - 将给定序列中零个或多个元素（如字符）去掉后所得结果
- 示例
 - 给定序列 X

X	A	B	C	B	D	A	B
-----	-----	-----	-----	-----	-----	-----	-----

- X 的子序列

X_1	A	B	C	B	D	A	B
-------	-----	-----	-----	-----	-----	-----	-----



问题背景：子序列

- 子序列
 - 将给定序列中零个或多个元素（如字符）去掉后所得结果
- 示例
 - 给定序列 X

X	A	B	C	B	D	A	B
-----	-----	-----	-----	-----	-----	-----	-----

- X 的子序列

X_1	A	B	C	B	D	A	B
-------	-----	-----	-----	-----	-----	-----	-----

X_2	A	B	C	B
-------	-----	-----	-----	-----



问题背景：子序列

- 子序列
 - 将给定序列中零个或多个元素（如字符）去掉后所得结果
- 示例
 - 给定序列 X

X	A	B	C	B	D	A	B
-----	-----	-----	-----	-----	-----	-----	-----

- X 的子序列

X_1	A	B	C	B	D	A	B
-------	-----	-----	-----	-----	-----	-----	-----

X_2	A	B	C	B
-------	-----	-----	-----	-----

X_3	A	C	B	B
-------	-----	-----	-----	-----



问题背景：公共子序列

- 给定两个序列*X*和*Y*

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>
----------	----------	----------	----------	----------	----------	----------



问题背景：公共子序列

- 给定两个序列 X 和 Y

X	A	B	C	B	D	A	B
-----	-----	-----	-----	-----	-----	-----	-----

Y	B	D	C	A	B	A
-----	-----	-----	-----	-----	-----	-----

- 公共子序列示例

X_1	C	A
-------	-----	-----

Y_1	C	A
-------	-----	-----

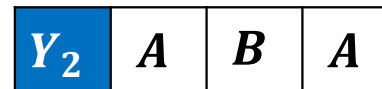
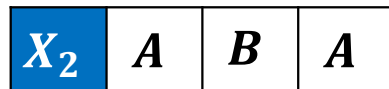
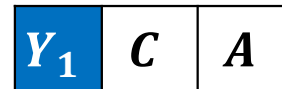
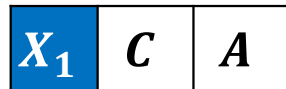


问题背景：公共子序列

- 给定两个序列*X*和*Y*



- 公共子序列示例





问题背景：公共子序列

- 给定两个序列X和Y

X	A	B	C	B	D	A	B
----------	---	---	---	---	---	---	---

Y	B	D	C	A	B	A
----------	---	---	---	---	---	---

- 公共子序列示例

X₁	C	A
----------------------	---	---

Y₁	C	A
----------------------	---	---

X₂	A	B	A
----------------------	---	---	---

Y₂	A	B	A
----------------------	---	---	---

X₃	B	C	A	B
----------------------	---	---	---	---

Y₃	B	C	A	B
----------------------	---	---	---	---



问题背景：公共子序列

- 给定两个序列X和Y

X	A	B	C	B	D	A	B
----------	---	---	---	---	---	---	---

Y	B	D	C	A	B	A
----------	---	---	---	---	---	---

- 公共子序列示例

X₁	C	A
----------------------	---	---

Y₁	C	A
----------------------	---	---

X₂	A	B	A
----------------------	---	---	---

Y₂	A	B	A
----------------------	---	---	---

X₃	B	C	A	B
----------------------	---	---	---	---

Y₃	B	C	A	B
----------------------	---	---	---	---

问题：如何求两个给定序列的最长公共子序列？



问题定义

- 形式化定义

最长公共子序列问题

Longest Common Subsequence Problem

输入

- 序列 $X = \langle x_1, x_2, \dots, x_n \rangle$ 和序列 $Y = \langle y_1, y_2, \dots, y_m \rangle$



问题定义

- 形式化定义

最长公共子序列问题

Longest Common Subsequence Problem

输入

- 序列 $X = \langle x_1, x_2, \dots, x_n \rangle$ 和序列 $Y = \langle y_1, y_2, \dots, y_m \rangle$

输出

- 求解一个公共子序列 $Z = \langle z_1, z_2, \dots, z_l \rangle$, 令

$$\max |Z|$$



问题定义

- 形式化定义

最长公共子序列问题

Longest Common Subsequence Problem

输入

- 序列 $X = \langle x_1, x_2, \dots, x_n \rangle$ 和序列 $Y = \langle y_1, y_2, \dots, y_m \rangle$

输出

- 求解一个公共子序列 $Z = \langle z_1, z_2, \dots, z_l \rangle$, 令

$$\max |Z|$$

$$\begin{aligned} s. t. \quad & \langle z_1, z_2, \dots, z_l \rangle = \langle x_{i_1}, x_{i_2}, \dots, x_{i_l} \rangle = \langle y_{j_1}, y_{j_2}, \dots, y_{j_l} \rangle \\ & (1 \leq i_1 < i_2, \dots, i_l \leq n; 1 \leq j_1 < j_2, \dots, j_l \leq m) \end{aligned}$$



问题定义

- 形式化定义

最长公共子序列问题

Longest Common Subsequence Problem

输入

- 序列 $X = \langle x_1, x_2, \dots, x_n \rangle$ 和序列 $Y = \langle y_1, y_2, \dots, y_m \rangle$

输出

- 求解一个公共子序列 $Z = \langle z_1, z_2, \dots, z_l \rangle$, 令

$\max |Z|$

优化目标

$$\begin{aligned} s. t. \quad & \langle z_1, z_2, \dots, z_l \rangle = \langle x_{i_1}, x_{i_2}, \dots, x_{i_l} \rangle = \langle y_{j_1}, y_{j_2}, \dots, y_{j_l} \rangle \\ & (1 \leq i_1 < i_2, \dots, i_l \leq n; 1 \leq j_1 < j_2, \dots, j_l \leq m) \end{aligned}$$



问题定义

- 形式化定义

最长公共子序列问题

Longest Common Subsequence Problem

输入

- 序列 $X = \langle x_1, x_2, \dots, x_n \rangle$ 和序列 $Y = \langle y_1, y_2, \dots, y_m \rangle$

输出

- 求解一个公共子序列 $Z = \langle z_1, z_2, \dots, z_l \rangle$, 令

$\max |Z|$

优化目标

$$\begin{aligned} s. t. \quad & \langle z_1, z_2, \dots, z_l \rangle = \langle x_{i_1}, x_{i_2}, \dots, x_{i_l} \rangle = \langle y_{j_1}, y_{j_2}, \dots, y_{j_l} \rangle \\ & (1 \leq i_1 < i_2, \dots, i_l \leq n; 1 \leq j_1 < j_2, \dots, j_l \leq m) \end{aligned}$$

约束条件



蛮力枚举

- 枚举所有子序列

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>
----------	----------	----------	----------	----------	----------	----------

<i>X</i>	<i>A</i>
----------	----------

枚举并检查长度为1的子序列



蛮力枚举

- 枚举所有子序列

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>
----------	----------	----------	----------	----------	----------	----------

<i>X</i>	<i>A</i>
----------	----------

<i>X</i>	<i>B</i>
----------	----------

枚举并检查长度为1的子序列

蛮力枚举



- 枚举所有子序列

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>
----------	----------	----------	----------	----------	----------	----------

<i>X</i>	<i>A</i>
----------	----------

<i>X</i>	<i>B</i>
----------	----------

<i>X</i>	<i>C</i>
----------	----------

枚举并检查长度为1的子序列



蛮力枚举

- 枚举所有子序列

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>
----------	----------	----------	----------	----------	----------	----------

<i>X</i>	<i>A</i>
----------	----------

<i>X</i>	<i>B</i>
----------	----------

<i>X</i>	<i>C</i>
----------	----------

<i>X</i>	<i>B</i>
----------	----------

<i>X</i>	<i>D</i>
----------	----------

<i>X</i>	<i>A</i>
----------	----------

<i>X</i>	<i>B</i>
----------	----------

枚举并检查长度为1的子序列



蛮力枚举

- 枚举所有子序列

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>
----------	----------	----------	----------	----------	----------	----------

<i>X</i>	<i>A</i>
----------	----------

<i>Y</i>	<i>B</i>
----------	----------

<i>X</i>	<i>B</i>
----------	----------

<i>Y</i>	<i>D</i>
----------	----------

<i>X</i>	<i>C</i>
----------	----------

<i>Y</i>	<i>C</i>
----------	----------

<i>X</i>	<i>B</i>
----------	----------

<i>Y</i>	<i>A</i>
----------	----------

<i>X</i>	<i>D</i>
----------	----------

<i>Y</i>	<i>B</i>
----------	----------

<i>X</i>	<i>A</i>
----------	----------

<i>Y</i>	<i>A</i>
----------	----------

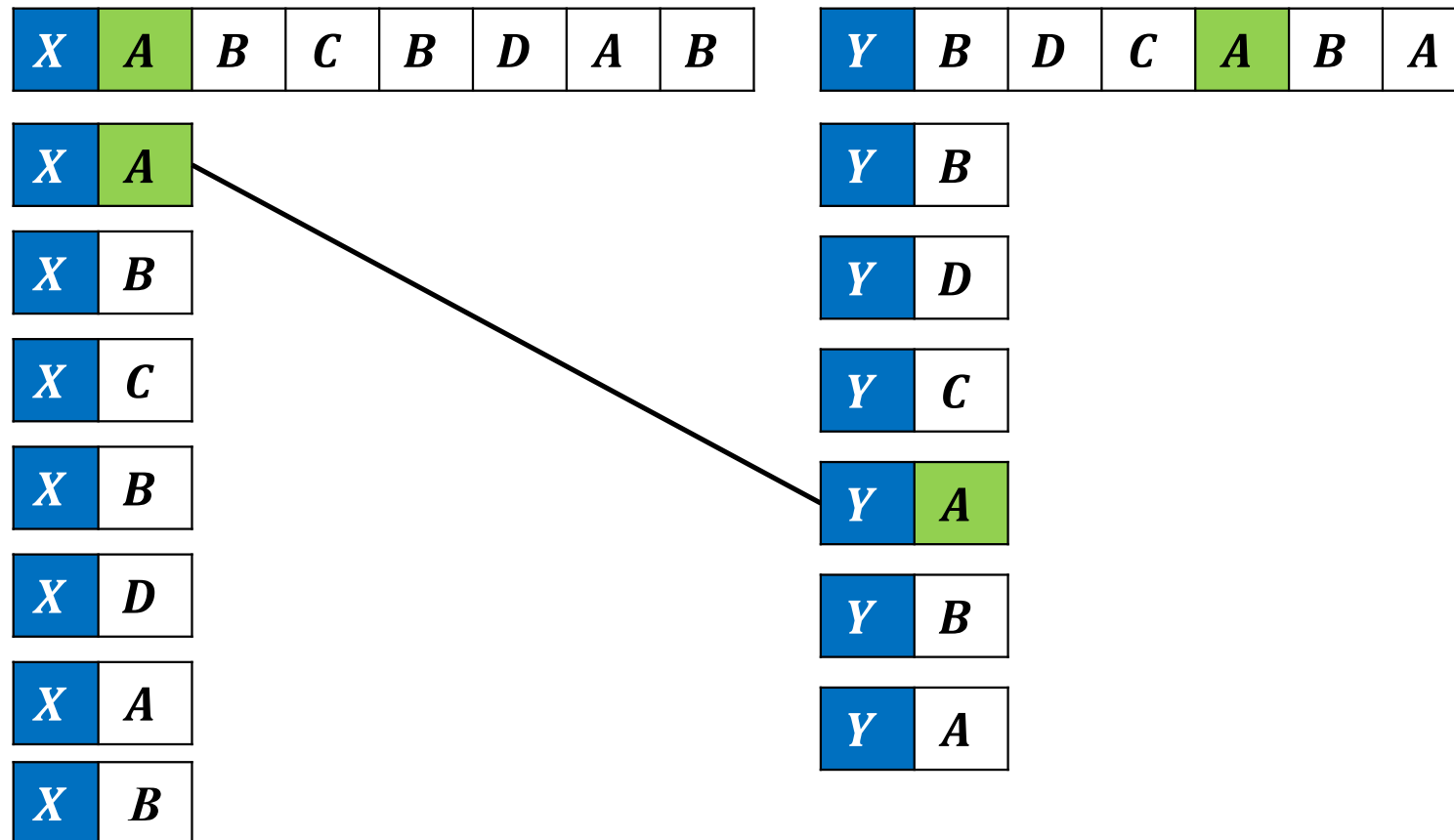
<i>X</i>	<i>B</i>
----------	----------

枚举并检查长度为1的子序列



蛮力枚举

- 枚举所有子序列



枚举并检查长度为1的子序列



蛮力枚举

- 枚举所有子序列

X	A	B	C	B	D	A	B
----------	---	---	---	---	---	---	---

X	A	B	C	B
----------	---	---	---	---

X	A	B	C	D
----------	---	---	---	---

X	A	B	C	A
----------	---	---	---	---

...

X	B	D	A	B
----------	---	---	---	---

...

X	C	B	D	B
----------	---	---	---	---

X	B	D	A	B
----------	---	---	---	---

Y	B	D	C	A	B	A
----------	---	---	---	---	---	---

Y	B	D	C	A
----------	---	---	---	---

Y	B	D	C	B
----------	---	---	---	---

...

Y	B	D	A	B
----------	---	---	---	---

...

Y	C	A	B	A
----------	---	---	---	---

枚举并检查长度为4的子序列



蛮力枚举

- 枚举所有子序列

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>
----------	----------	----------	----------	----------	----------	----------

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>
----------	----------	----------	----------	----------

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>A</i>
----------	----------	----------	----------	----------

...

...

<i>X</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

...

...

<i>X</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>Y</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>
----------	----------	----------	----------	----------

<i>X</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

枚举并检查长度为4的子序列



蛮力枚举

- 枚举所有子序列

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>
----------	----------	----------	----------	----------	----------	----------

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>
----------	----------	----------	----------	----------

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>A</i>
----------	----------	----------	----------	----------

...

...

<i>X</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

...

...

<i>X</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>Y</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>
----------	----------	----------	----------	----------

<i>X</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

枚举并检查长度为4的子序列



蛮力枚举

- 枚举所有子序列

X	A	B	C	B	D	A	B
----------	---	---	---	---	---	---	---

X	A	B	C	B	D
----------	---	---	---	---	---

X	A	B	C	B	A
----------	---	---	---	---	---

X	A	B	C	B	B
----------	---	---	---	---	---

X	A	B	B	D	A
----------	---	---	---	---	---

X	A	B	B	D	B
----------	---	---	---	---	---

...

X	C	B	D	A	B
----------	---	---	---	---	---

Y	B	D	C	A	B	A
----------	---	---	---	---	---	---

Y	B	D	C	A	B
----------	---	---	---	---	---

Y	B	D	C	A	A
----------	---	---	---	---	---

Y	B	D	C	B	A
----------	---	---	---	---	---

Y	B	D	A	B	A
----------	---	---	---	---	---

Y	B	C	A	B	A
----------	---	---	---	---	---

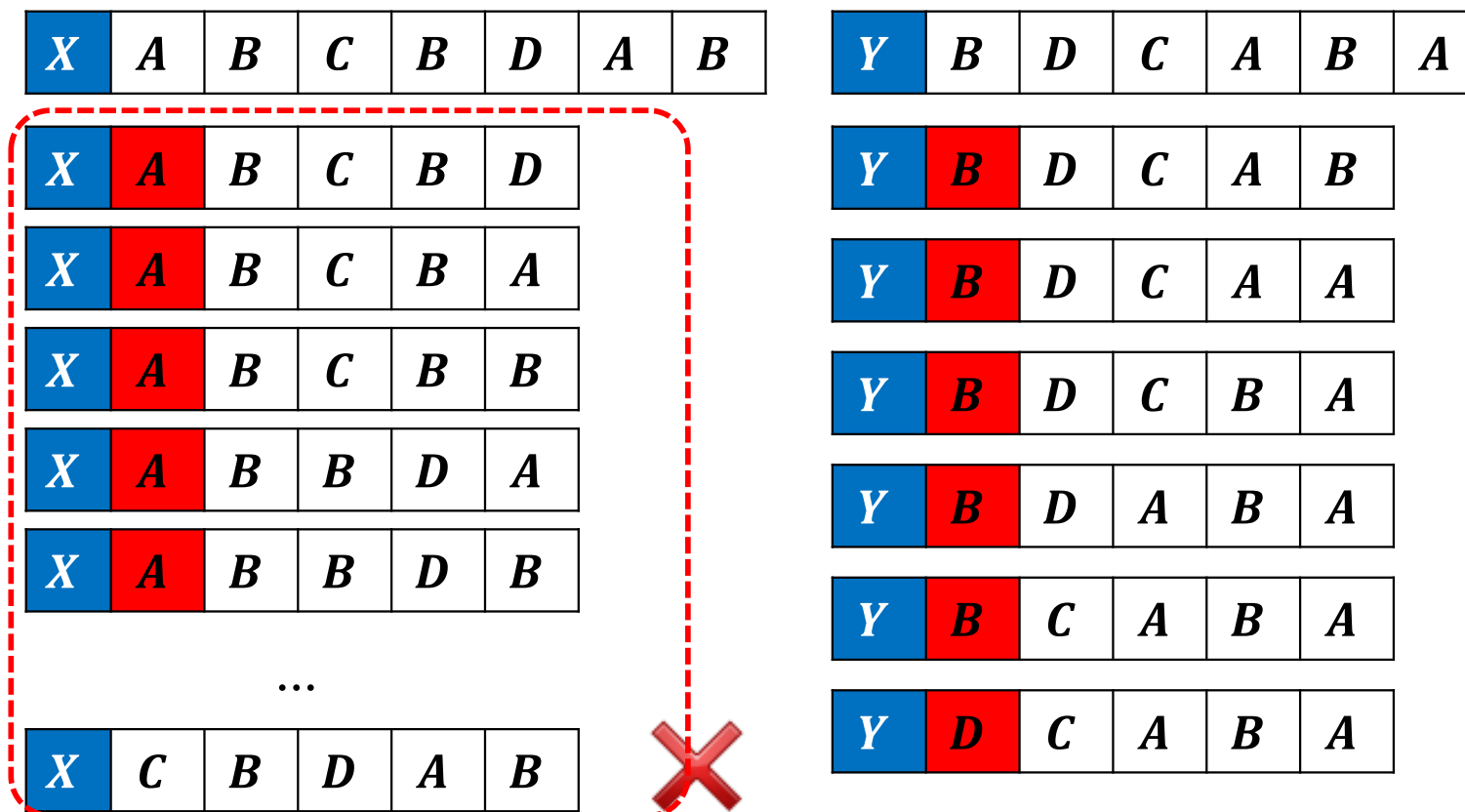
Y	D	C	A	B	A
----------	---	---	---	---	---

枚举并检查长度为5的子序列



蛮力枚举

- 枚举所有子序列



枚举并检查长度为5的子序列



蛮力枚举

- 枚举所有子序列

X	A	B	C	B	D	A	B
----------	---	---	---	---	---	---	---

Y	B	D	C	A	B	A
----------	---	---	---	---	---	---

X	B
----------	---

Y	B
----------	---

长度为1

X	A	B
----------	---	---

Y	A	B
----------	---	---

长度为2

X	A	B	A
----------	---	---	---

Y	A	B	A
----------	---	---	---

长度为3

X	B	D	A	B
----------	---	---	---	---

Y	B	D	A	B
----------	---	---	---	---

长度为4

X					
----------	--	--	--	--	--

Y					
----------	--	--	--	--	--

长度为5

X						
----------	--	--	--	--	--	--

Y						
----------	--	--	--	--	--	--

 长度为6



蛮力枚举

- 枚举所有子序列

X	A	B	C	B	D	A	B
----------	---	---	---	---	---	---	---

Y	B	D	C	A	B	A
----------	---	---	---	---	---	---

X	B
----------	---

Y	B
----------	---

长度为1

X	A	B
----------	---	---

Y	A	B
----------	---	---

长度为2

X	A	B	A
----------	---	---	---

Y	A	B	A
----------	---	---	---

长度为3

X	B	D	A	B
----------	---	---	---	---

Y	B	D	A	B
----------	---	---	---	---

长度为4



蛮力枚举

- 枚举所有子序列

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>
----------	----------	----------	----------	----------	----------	----------

<i>X</i>	<i>B</i>
----------	----------

<i>Y</i>	<i>B</i>
----------	----------

长度为1

<i>X</i>	<i>A</i>	<i>B</i>
----------	----------	----------

<i>Y</i>	<i>A</i>	<i>B</i>
----------	----------	----------

长度为2

<i>X</i>	<i>A</i>	<i>B</i>	<i>A</i>
----------	----------	----------	----------

<i>Y</i>	<i>A</i>	<i>B</i>	<i>A</i>
----------	----------	----------	----------

长度为3

<i>X</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

长度为4

最长公共子序列



枚举观察



<i>X</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

长度为4

枚举观察



<i>X</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>X</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>Y</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------

长度为4

长度为3

枚举观察



<i>X</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>X</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------

<i>X</i>	<i>A</i>	<i>B</i>
----------	----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>Y</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------

<i>Y</i>	<i>A</i>	<i>B</i>
----------	----------	----------

长度为4

长度为3

长度为2

枚举观察



<i>X</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>X</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------

<i>X</i>	<i>A</i>	<i>B</i>
----------	----------	----------

<i>X</i>	<i>B</i>
----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>Y</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------

<i>Y</i>	<i>A</i>	<i>B</i>
----------	----------	----------

<i>Y</i>	<i>B</i>
----------	----------

长度为4

长度为3

长度为2

长度为1



枚举观察

<i>X</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>X</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------

<i>X</i>	<i>A</i>	<i>B</i>
----------	----------	----------

<i>X</i>	<i>B</i>
----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>Y</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------

<i>Y</i>	<i>A</i>	<i>B</i>
----------	----------	----------

<i>Y</i>	<i>B</i>
----------	----------

长度为4

长度为3

长度为2

长度为1

- 可能存在最优子结构和重叠子问题



枚举观察

<i>X</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>X</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------

<i>X</i>	<i>A</i>	<i>B</i>
----------	----------	----------

<i>X</i>	<i>B</i>
----------	----------

<i>Y</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------	----------

<i>Y</i>	<i>D</i>	<i>A</i>	<i>B</i>
----------	----------	----------	----------

<i>Y</i>	<i>A</i>	<i>B</i>
----------	----------	----------

<i>Y</i>	<i>B</i>
----------	----------

长度为4

长度为3

长度为2

长度为1

- 可能存在最优子结构和重叠子问题

问题：如何利用动态规划求解？

问题结构分析

- 给出问题表示
 - $C[i, j]$: $X[1..i]$ 和 $Y[1..j]$ 的最长公共子序列长度

X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

问题结构分析



递推关系建立



自底向上计算



最优方案追踪

问题结构分析

- 给出问题表示

- $C[i, j]$: $X[1..i]$ 和 $Y[1..j]$ 的最长公共子序列长度

X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

- 明确原始问题

- $C[n, m]$: $X[1..n]$ 和 $Y[1..m]$ 的最长公共子序列长度

问题结构分析



递推关系建立



自底向上计算



最优方案追踪



递推关系建立：分析最优（子）结构

- 考察末尾字符

- 情况1： $x_7 \neq y_6$

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>	

- 情况2： $x_7 = y_6$

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
	<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	

问题结构分析



递推关系建立



自底向上计算

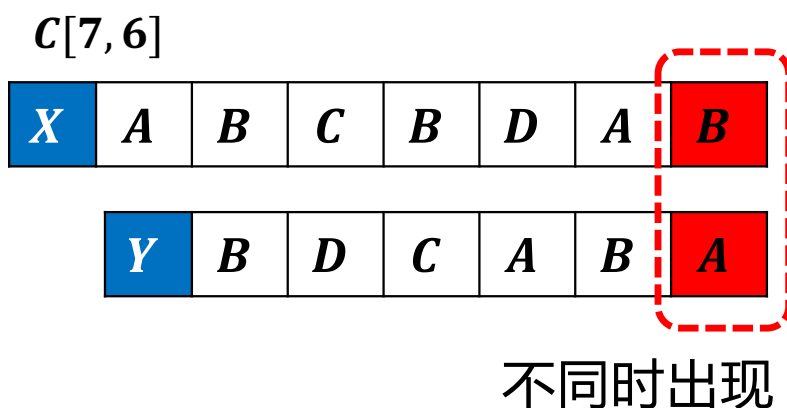


最优方案追踪



递推关系建立：分析最优（子）结构

- 考察末尾字符
 - 情况1： $x_7 \neq y_6$



问题结构分析

递推关系建立

自底向上计算

最优方案追踪

递推关系建立：分析最优（子）结构

- 考察末尾字符

- 情况1： $x_7 \neq y_6$

$C[7, 6]$

X	A	B	C	B	D	A	B
----------	---	---	---	---	---	---	----------

Y	B	D	C	A	B	A
----------	---	---	---	---	---	----------

$C[7, 6 - 1] + 0$							
X	A	B	C	B	D	A	B
Y	B	D	C	A	B	A	

问题结构分析

递推关系建立

自底向上计算

最优方案追踪



递推关系建立：分析最优（子）结构

- 考察末尾字符

- 情况1： $x_7 \neq y_6$

$C[7,6]$

X	A	B	C	B	D	A	B
Y	B	D	C	A	B	A	

$C[7,6-1] + 0$

X	A	B	C	B	D	A	B
Y	B	D	C	A	B	A	

$C[7-1,6] + 0$

X	A	B	C	B	D	A	B
Y	B	D	C	A	B	A	

问题结构分析

递推关系建立

自底向上计算

最优方案追踪



递推关系建立：分析最优（子）结构

- 考察末尾字符

- 情况1： $x_7 \neq y_6$

$C[7,6]$

X	A	B	C	B	D	A	B
Y	B	D	C	A	B		A

max

$C[7,6-1] + 0$

X	A	B	C	B	D	A	B
Y	B	D	C	A	B	A	

$C[7-1,6] + 0$

X	A	B	C	B	D	A	B
Y	B	D	C	A	B	A	

问题结构分析

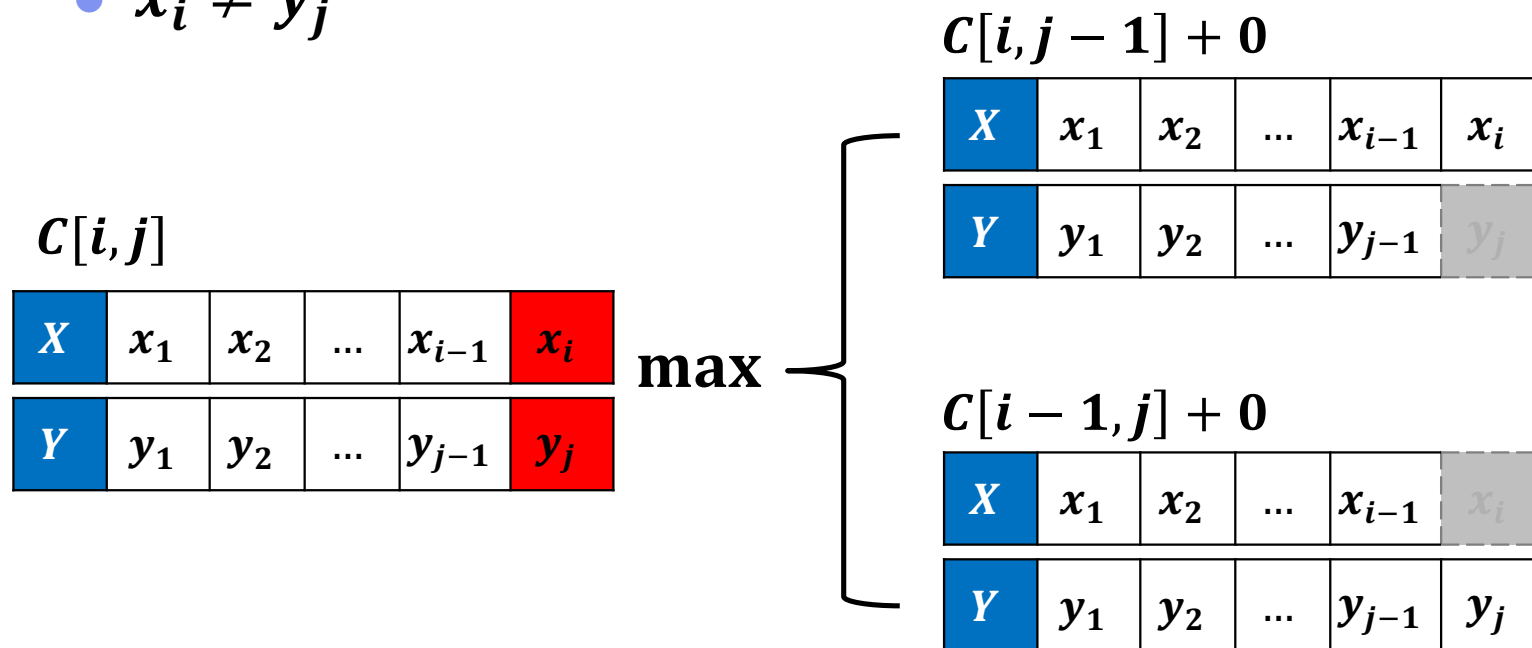
递推关系建立

自底向上计算

最优方案追踪

递推关系建立：分析最优（子）结构

- $x_i \neq y_j$



问题结构分析

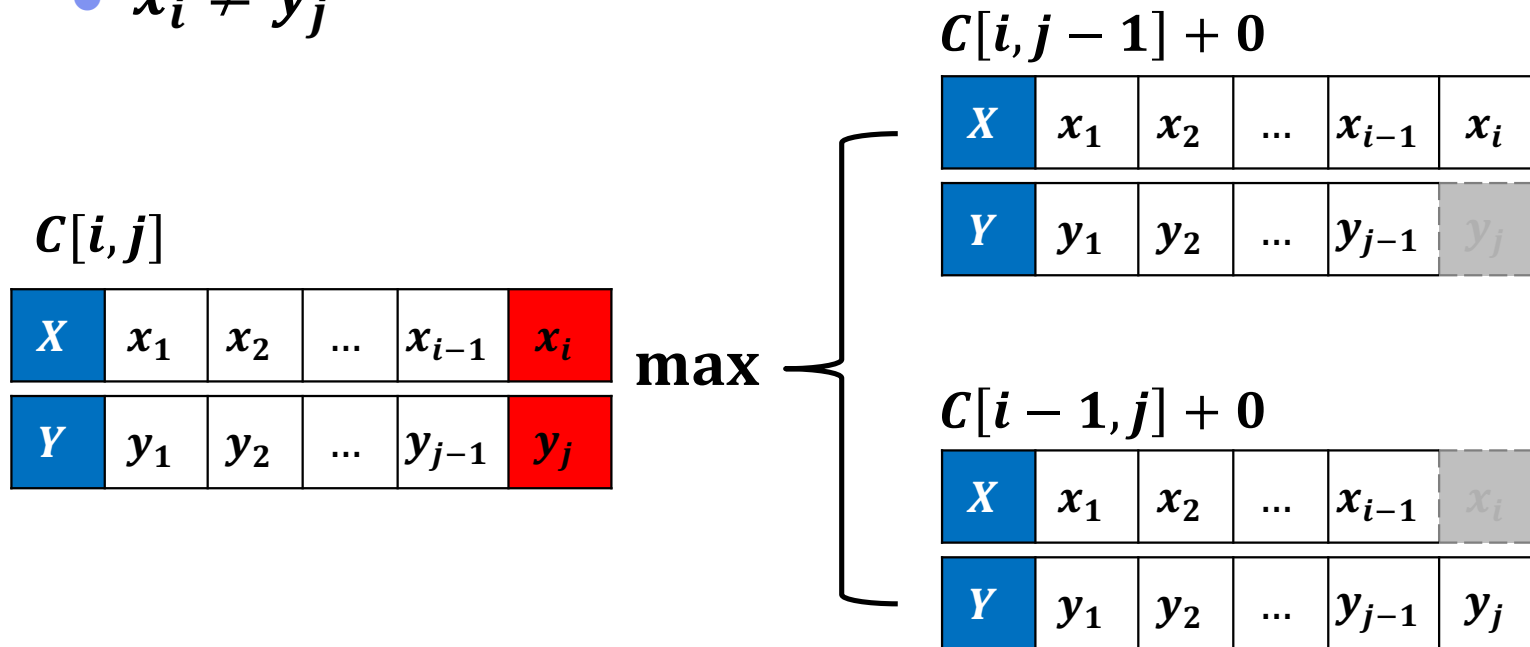
递推关系建立

自底向上计算

最优方案追踪

递推关系建立：分析最优（子）结构

- $x_i \neq y_j$



- $C[i, j] = \max\{C[i-1, j], C[i, j-1]\}$

问题结构分析

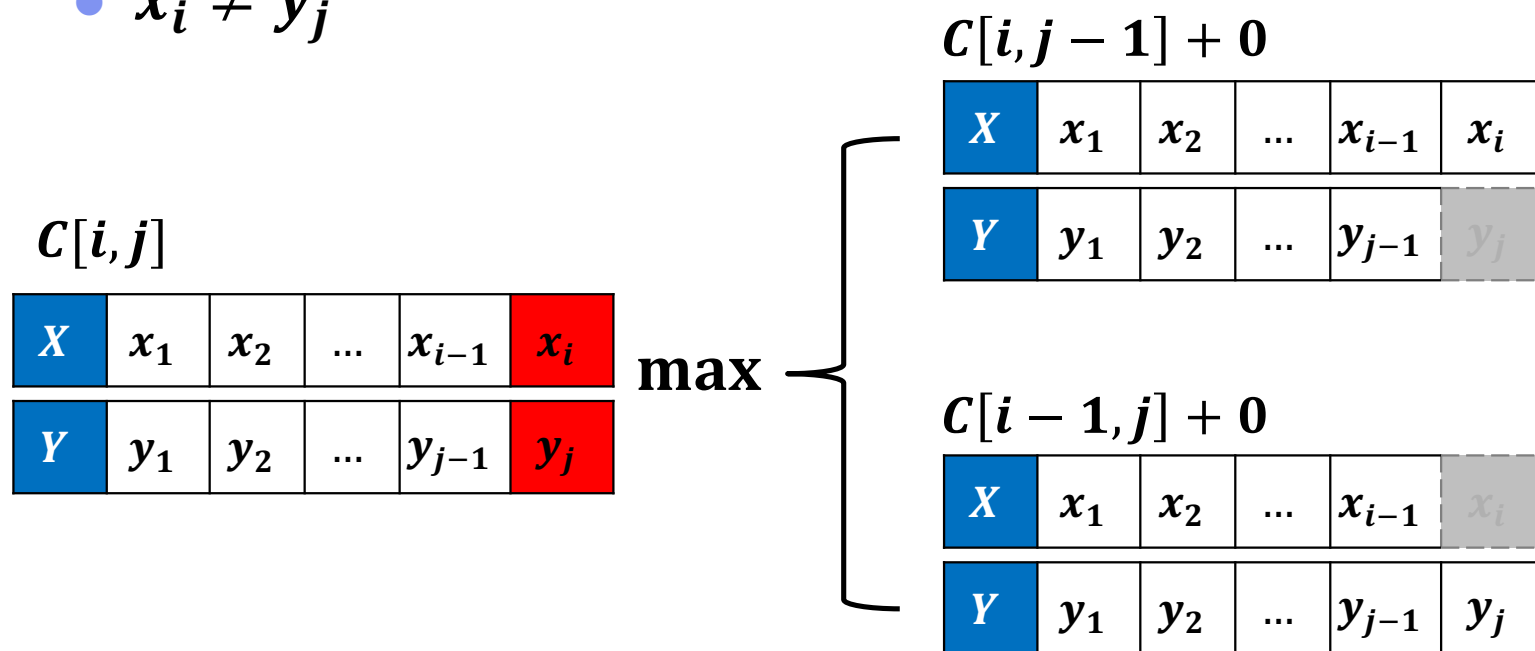
递推关系建立

自底向上计算

最优方案追踪

递推关系建立：分析最优（子）结构

- $x_i \neq y_j$



- $C[i, j] = \max\{C[i-1, j], C[i, j-1]\}$

最优子结构

问题结构分析

递推关系建立

自底向上计算

最优方案追踪



递推关系建立：分析最优（子）结构

- 考察末尾字符
 - 情况2： $x_7 = y_6$

$C[7,6]$

X	A	B	C	B	D	A	B
----------	---	---	---	---	---	---	----------

Y	B	D	C	A	B	B
----------	---	---	---	---	---	----------

问题结构分析



递推关系建立



自底向上计算



最优方案追踪



递推关系建立：分析最优（子）结构

- 考察末尾字符

- 情况2： $x_7 = y_6$

$C[7, 6]$

X	A	B	C	B	D	A	B
Y	B	D	C	A	B	B	

可同时出现

问题结构分析

递推关系建立

自底向上计算

最优方案追踪

递推关系建立：分析最优（子）结构

- 考察末尾字符

- 情况2： $x_7 = y_6$

$C[7, 6]$

X	A	B	C	B	D	A	B
----------	---	---	---	---	---	---	---

Y	B	D	C	A	B	B
----------	---	---	---	---	---	---

可同时出现

$C[7 - 1, 6 - 1] + 1$							
X	A	B	C	B	D	A	B
Y	B	D	C	A	B	B	

问题结构分析

递推关系建立

自底向上计算

最优方案追踪

递推关系建立：分析最优（子）结构

- 考察末尾字符

- 情况2： $x_7 = y_6$

$C[7, 6]$

X	A	B	C	B	D	A	B
---	---	---	---	---	---	---	---

Y	B	D	C	A	B	B
---	---	---	---	---	---	---

也可不同时出现

$$C[7-1, 6-1] + 1$$

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>B</i>	

$$C[7, 6-1] + 0$$

<i>X</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
<i>Y</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>B</i>	

$$C[7-1, 6] + 0$$

X	A	B	C	B	D	A	B
Y	B	D	C	A	B	B	

问题结构分析

递推关系建立

自底向上计算

最优方案追踪



递推关系建立：分析最优（子）结构

- 考察末尾字符

- 情况2： $x_7 = y_6$

$C[7,6]$

X	A	B	C	B	D	A	B
Y	B	D	C	A	B	B	

max

$C[7-1, 6-1] + 1$							
X	A	B	C	B	D	A	B
Y	B	D	C	A	B		B

$C[7, 6-1] + 0$							
X	A	B	C	B	D	A	B
Y	B	D	C	A	B		B

$C[7-1, 6] + 0$							
X	A	B	C	B	D	A	B
Y	B	D	C	A	B	B	

问题结构分析

递推关系建立

自底向上计算

最优方案追踪

递推关系建立：分析最优（子）结构

- $x_i = y_j$

$$C[i, j]$$

X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

max

$$C[i-1, j-1] + 1$$

X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

$$C[i-1, j] + 0$$

X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

$$C[i, j-1] + 0$$

X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

问题结构分析

递推关系建立

自底向上计算

最优方案追踪



递推关系建立：分析最优（子）结构

- $x_i = y_j$

$C[i, j]$					
X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

max

$$C[i-1, j-1] + 1$$

X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

$$C[i-1, j] + 0$$

X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

$$C[i, j-1] + 0$$

X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

问题：3个问题是否都需要求解？

问题结构分析

递推关系建立

自底向上计算

最优方案追踪

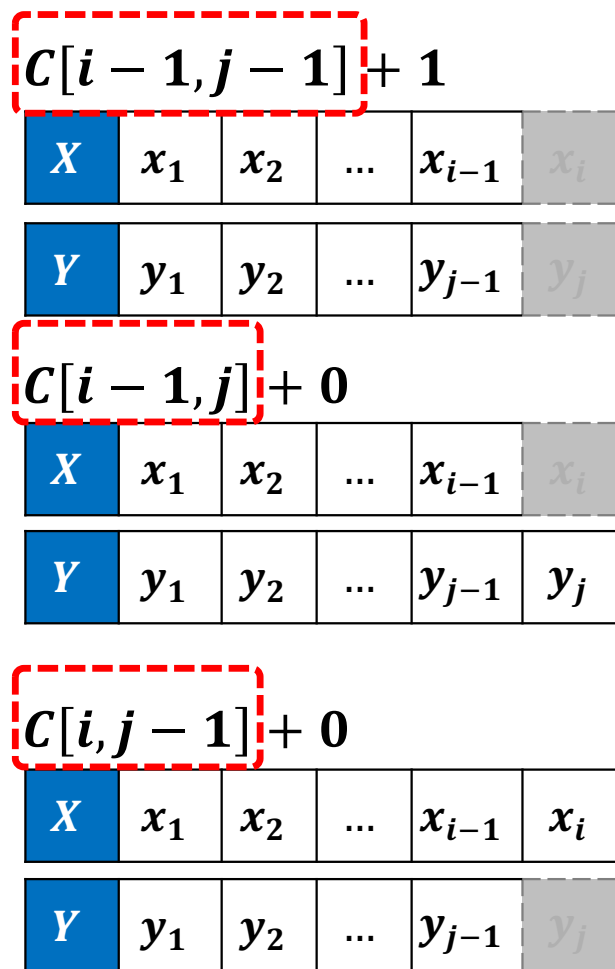
递推关系建立：分析最优（子）结构

- $x_i = y_j$
 - $C[i-1, j]$ 比 $C[i-1, j-1]$ 至多大1
 - $C[i, j-1]$ 比 $C[i-1, j-1]$ 至多大1

$$C[i, j]$$

X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

max



问题结构分析

递推关系建立

自底向上计算

最优方案追踪

递推关系建立：分析最优（子）结构

- $x_i = y_j$
 - $C[i-1, j]$ 比 $C[i-1, j-1]$ 至多大1
 - $C[i, j-1]$ 比 $C[i-1, j-1]$ 至多大1
 - $C[i-1, j-1] + 1$, 另外两个+0

$$C[i, j]$$

X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

max

$C[i-1, j-1] + 1$					
X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

$C[i-1, j] + 0$					
X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

$C[i, j-1] + 0$					
X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

问题结构分析

递推关系建立

自底向上计算

最优方案追踪

递推关系建立：分析最优（子）结构

- $x_i = y_j$
 - $C[i-1, j]$ 比 $C[i-1, j-1]$ 至多大1
 - $C[i, j-1]$ 比 $C[i-1, j-1]$ 至多大1
 - $C[i-1, j-1] + 1$, 另外两个+0

$C[i, j]$					
X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

max

$$C[i-1, j-1] + 1 \geq \max\{C[i, j-1], C[i-1, j]\}$$

$C[i-1, j-1] + 1$					
X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

$C[i-1, j] + 0$					
X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

$C[i, j-1] + 0$					
X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

问题结构分析

递推关系建立

自底向上计算

最优方案追踪



递推关系建立：分析最优（子）结构

- $x_i = y_j$
 - $C[i-1, j]$ 比 $C[i-1, j-1]$ 至多大1
 - $C[i, j-1]$ 比 $C[i-1, j-1]$ 至多大1
 - $C[i-1, j-1] + 1$, 另外两个+0

$C[i, j]$					
X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

max

$$C[i-1, j-1] + 1 \geq \max\{C[i, j-1], C[i-1, j]\}$$

$C[i-1, j-1] + 1$ 已充分

X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

$C[i-1, j] + 0$ 非必要

X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

$C[i, j-1] + 0$ 非必要

X	x_1	x_2	...	x_{i-1}	x_i
Y	y_1	y_2	...	y_{j-1}	y_j

问题结构分析

递推关系建立

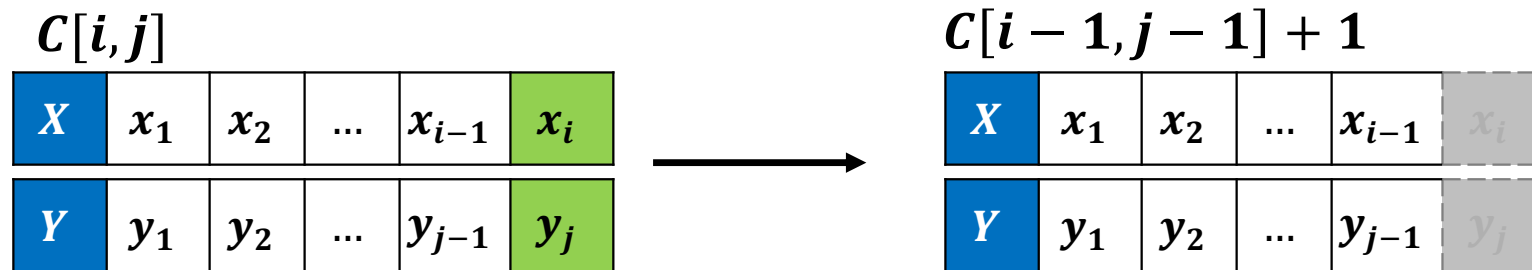
自底向上计算

最优方案追踪



递推关系建立：分析最优（子）结构

- $x_i = y_j$



问题结构分析



递推关系建立



自底向上计算

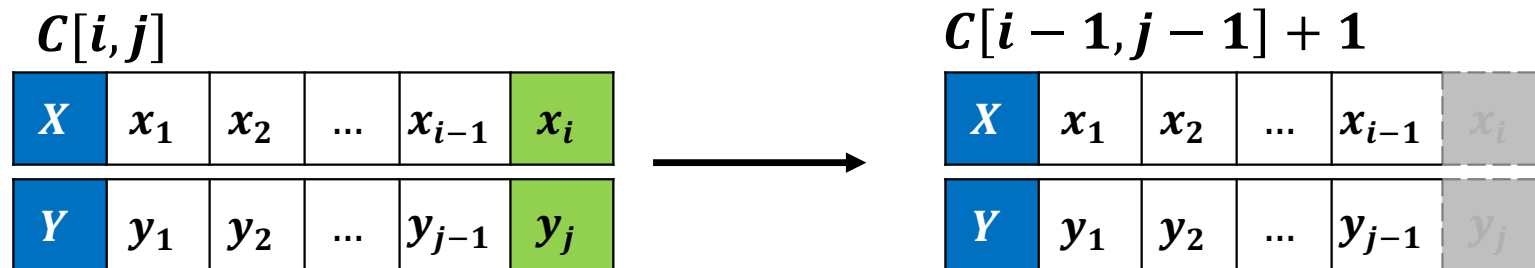


最优方案追踪



递推关系建立：分析最优（子）结构

- $x_i = y_j$



- $C[i, j] = C[i-1, j-1] + 1$

问题结构分析

递推关系建立

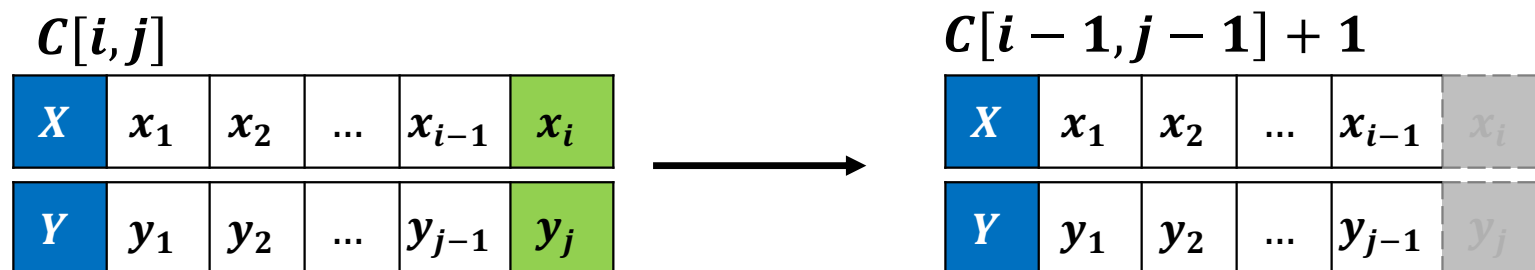
自底向上计算

最优方案追踪



递推关系建立：分析最优（子）结构

- $x_i = y_j$



- $C[i, j] = C[i-1, j-1] + 1$

最优子结构

问题结构分析

递推关系建立

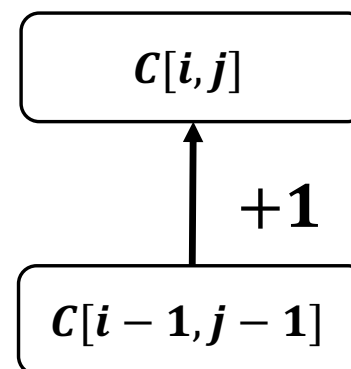
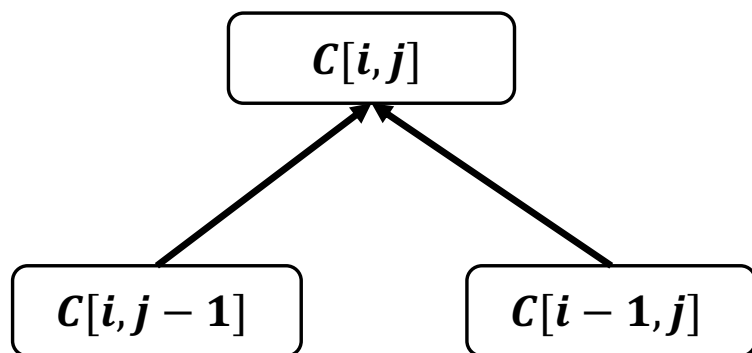
自底向上计算

最优方案追踪



递推关系建立：构造递推公式

- $$C[i, j] = \begin{cases} \max\{C[i-1, j], C[i, j-1]\}, & x_i \neq y_j \\ C[i-1, j-1] + 1, & x_i = y_j \end{cases}$$



问题结构分析



递推关系建立



自底向上计算



最优方案追踪



自底向上计算：确定计算顺序

- 初始化

- $C[i, 0] = C[0, j] = 0$
 - 某序列长度为0时，最长公共子序列长度为0

$C[i, j]$	$j = 0$	$j = 1$	$j = 2$...	$j = m$
$i = 0$					
$i = 1$					
$i = 2$					
...					
$i = n$					

问题结构分析



递推关系建立



自底向上计算



最优方案追踪



自底向上计算：确定计算顺序

- 初始化

- $C[i, 0] = C[0, j] = 0$
 - 某序列长度为0时，最长公共子序列长度为0

$C[i, j]$	$j = 0$	$j = 1$	$j = 2$...	$j = m$
$i = 0$	0	0	0	0	0
$i = 1$	0				
$i = 2$	0				
...	0				
$i = n$	0				

初始化

问题结构分析



递推关系建立



自底向上计算



最优方案追踪

自底向上计算：确定计算顺序

- 初始化

- $C[i, 0] = C[0, j] = 0$
 - 某序列长度为0时，最长公共子序列长度为0

- 递推公式

- $$C[i, j] = \begin{cases} \max\{C[i-1, j], C[i, j-1]\}, & x_i \neq y_j \\ C[i-1, j-1] + 1, & x_i = y_j \end{cases}$$

$C[i, j]$	$j = 0$	$j = 1$	$j = 2$...	$j = m$
$i = 0$	0	0	0	0	0
$i = 1$	0				
$i = 2$	0				
...	0				
$i = n$	0				

问题结构分析



递推关系建立



自底向上计算



最优方案追踪



自底向上计算：确定计算顺序

- 初始化

- $C[i, 0] = C[0, j] = 0$
 - 某序列长度为0时，最长公共子序列长度为0

- 递推公式

- $$C[i, j] = \begin{cases} \max\{C[i-1, j], C[i, j-1]\}, & x_i \neq y_j \\ C[i-1, j-1] + 1, & x_i = y_j \end{cases}$$

$C[i, j]$	$j = 0$	$j = 1$	$j = 2$	\dots	$j = m$
$i = 0$	0	0	0	0	0
$i = 1$	0				
$i = 2$	0				
\dots	0				
$i = n$	0				

Diagram illustrating the dynamic programming table structure. A black arrow points from the cell $C[i, j]$ to the cell $C[i-1, j]$ (above it). A red arrow points from the cell $C[i, j]$ to the cell $C[i, j-1]$ (to its left).

问题结构分析



递推关系建立



自底向上计算



最优方案追踪



自底向上计算：确定计算顺序

- 初始化

- $C[i, 0] = C[0, j] = 0$
 - 某序列长度为0时，最长公共子序列长度为0

- 递推公式

- $$C[i, j] = \begin{cases} \max\{C[i-1, j], C[i, j-1]\}, & x_i \neq y_j \\ C[i-1, j-1] + 1 & , x_i = y_j \end{cases}$$

$C[i, j]$	$j = 0$	$j = 1$	$j = 2$...	$j = m$
$i = 0$	0	0	0	0	0
$i = 1$	0				
$i = 2$	0				
...	0				
$i = n$	0				

Diagram illustrating the dynamic programming table structure. A red arrow points to the cell $C[i, j]$ in the row i and column j . A black arrow points to the cell $C[i-1, j]$ in the row $i-1$ and column j . A black arrow points to the cell $C[i, j-1]$ in the row i and column $j-1$.

问题结构分析



递推关系建立



自底向上计算



最优方案追踪

自底向上计算：确定计算顺序

- 初始化

- $C[i, 0] = C[0, j] = 0$
 - 某序列长度为0时，最长公共子序列长度为0

- 递推公式

- $$C[i, j] = \begin{cases} \max\{C[i-1, j], C[i, j-1]\}, & x_i \neq y_j \\ C[i-1, j-1] + 1, & x_i = y_j \end{cases}$$

$C[i, j]$	$j = 0$	$j = 1$	$j = 2$...	$j = m$
$i = 0$	0	0	0	0	0
$i = 1$	0				
$i = 2$	0				
...	0				
$i = n$	0				

问题结构分析



递推关系建立



自底向上计算



最优方案追踪

自底向上计算：依次求解问题

- 初始化

- $C[i, 0] = C[0, j] = 0$
 - 某序列长度为0时，最长公共子序列长度为0

- 递推公式

- $$C[i, j] = \begin{cases} \max\{C[i-1, j], C[i, j-1]\}, & x_i \neq y_j \\ C[i-1, j-1] + 1, & x_i = y_j \end{cases}$$

$C[i, j]$	$j = 0$	$j = 1$	$j = 2$...	$j = m$
$i = 0$	0	0	0	0	0
$i = 1$	0	→			
$i = 2$	0	←	→	→	→
...	0	←	→	→	→
$i = n$	0	←	→	→	→ ★

自底向上计算

问题结构分析



递推关系建立



自底向上计算



最优方案追踪

最优方案追踪：记录决策过程

- 构造追踪数组 $rec[1..n]$ ，记录子问题来源

$$rec[i,j] = \begin{cases} LU, & \text{if } C[i,j] = C[i-1,j-1] + 1 \\ U, & \text{if } C[i,j] = C[i-1,j] \\ L, & \text{if } C[i,j] = C[i,j-1] \end{cases}$$

$C[i,j]$	$j = 0$	$j = 1$	$j = 2$...	$j = m$
$i = 0$					
$i = 1$					
$i = 2$					
...					
$i = n$					

问题结构分析



递推关系建立



自底向上计算



最优方案追踪



最优方案追踪：记录决策过程

- 构造追踪数组 $rec[1..n]$ ，记录子问题来源

$$rec[i, j] = \begin{cases} LU, & \text{if } C[i, j] = C[i-1, j-1] + 1 \\ U, & \text{if } C[i, j] = C[i-1, j] \\ L, & \text{if } C[i, j] = C[i, j-1] \end{cases}$$

$C[i, j]$	$j = 0$	$j = 1$	$j = 2$	$...$	$j = m$
$i = 0$					
$i = 1$					
$i = 2$					
$...$					
$i = n$					

Diagram illustrating the DP table structure and the current state of calculation. A red arrow points to the cell $C[i, j]$ from the cell $C[i-1, j-1]$. A black arrow points to the cell $C[i, j]$ from the cell $C[i, j-1]$.

- 最长公共子序列末尾为 $X[i] = Y[j]$

问题结构分析



递推关系建立



自底向上计算



最优方案追踪



最优方案追踪：记录决策过程

- 构造追踪数组 $rec[1..n]$ ，记录子问题来源

$$rec[i,j] = \begin{cases} LU, & \text{if } C[i,j] = C[i-1,j-1] + 1 \\ U, & \text{if } C[i,j] = C[i-1,j] \\ L, & \text{if } C[i,j] = C[i,j-1] \end{cases}$$

$C[i,j]$	$j = 0$	$j = 1$	$j = 2$	$...$	$j = m$
$i = 0$					
$i = 1$					
$i = 2$					
$...$					
$i = n$					

Diagram illustrating the DP table structure. A black arrow points from the cell $C[i,j]$ to the cell $C[i-1,j-1]$, and a red arrow points from the cell $C[i,j]$ to the cell $C[i,j-1]$.

- 最长公共子序列在 $X[1..i-1]$ 和 $Y[1..j]$ 中

问题结构分析



递推关系建立



自底向上计算



最优方案追踪

最优方案追踪：记录决策过程

- 构造追踪数组 $rec[1..n]$ ，记录子问题来源

$$rec[i,j] = \begin{cases} LU, & \text{if } C[i,j] = C[i-1,j-1] + 1 \\ U, & \text{if } C[i,j] = C[i-1,j] \\ L, & \text{if } C[i,j] = C[i,j-1] \end{cases}$$

$C[i,j]$	$j = 0$	$j = 1$	$j = 2$...	$j = m$
$i = 0$					
$i = 1$					
$i = 2$					
...					
$i = n$					

- 最长公共子序列在 $X[1..i]$ 和 $Y[1..j-1]$ 中

问题结构分析



递推关系建立



自底向上计算



最优方案追踪



最优方案追踪：输出最优方案

- 输出最长公共子序列

- $$rec[i, j] = \begin{cases} LU, & \text{if } C[i, j] = C[i - 1, j - 1] + 1 \\ U, & \text{if } C[i, j] = C[i - 1, j] \\ \textcolor{red}{L}, & \text{if } C[i, j] = C[i, j - 1] \end{cases}$$

$C[i, j]$	$j = 0$	$j = 1$	$j = 2$...	$j = m$
$i = 0$					
$i = 1$					
$i = 2$					
...					
$i = n$					

$rec[] = L$

- 最长公共子序列在 $X[1..i]$ 和 $Y[1..j - 1]$ 中

问题结构分析



递推关系建立



自底向上计算



最优方案追踪



最优方案追踪：输出最优方案

- 输出最长公共子序列

- $$rec[i, j] = \begin{cases} LU, & \text{if } C[i, j] = C[i - 1, j - 1] + 1 \\ U, & \text{if } C[i, j] = C[i - 1, j] \\ L, & \text{if } C[i, j] = C[i, j - 1] \end{cases}$$

$C[i, j]$	$j = 0$	$j = 1$	$j = 2$...	$j = m$
$i = 0$					
$i = 1$					
$i = 2$					
...					
$i = n$			$rec[] = U$	\uparrow	\leftarrow

$rec[] = L$

- 最长公共子序列在 $X[1..i - 1]$ 和 $Y[1..j]$ 中

问题结构分析



递推关系建立



自底向上计算



最优方案追踪



最优方案追踪：输出最优方案

- 输出最长公共子序列

- $$rec[i, j] = \begin{cases} LU, & \text{if } C[i, j] = C[i - 1, j - 1] + 1 \\ U, & \text{if } C[i, j] = C[i - 1, j] \\ L, & \text{if } C[i, j] = C[i, j - 1] \end{cases}$$

$C[i, j]$	$j = 0$	$j = 1$	$j = 2$...	$j = m$
$i = 0$					
$i = 1$					
$i = 2$					
...					
$i = n$					

Diagram illustrating the backtracking process for the Longest Common Subsequence (LCS) problem. The table shows the DP table $C[i, j]$ and the corresponding $rec[i, j]$ values. Red dashed boxes highlight the path from the bottom-right cell (n, m) back to the start, indicating the sequence of operations (L, U, L) that led to the optimal solution. Arrows point to the cells $rec[i, j] = U$ and $rec[i, j] = L$.

- 最长公共子序列末尾为 $X[i] = Y[j]$

问题结构分析



递推关系建立



自底向上计算



最优方案追踪



最优方案追踪：输出最优方案

- 输出最长公共子序列

- $$rec[i, j] = \begin{cases} LU, & \text{if } C[i, j] = C[i - 1, j - 1] + 1 \\ U, & \text{if } C[i, j] = C[i - 1, j] \\ L, & \text{if } C[i, j] = C[i, j - 1] \end{cases}$$

$C[i, j]$	$j = 0$	$j = 1$	$j = 2$...	$j = m$
$i = 0$					
$i = 1$					
$i = 2$					
...					
$i = n$					

Diagram illustrating the backtracking process for the Longest Common Subsequence (LCS) problem. Red dashed boxes highlight the path from the bottom-right cell (i=n, j=m) back to the top-left cell (i=0, j=0). Arrows indicate the direction of backtracking. Labels indicate the state of the rec array at each step: $rec[] = LU$ (top-right), $rec[] = U$ (middle-left), and $rec[] = L$ (bottom-right).

问题结构分析

递推关系建立

自底向上计算

最优方案追踪

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0							
1							
2							
3							
4							
5							
6							
7							

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>
Y_j	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0						
2	0						
3	0						
4	0						
5	0						
6	0						
7	0						

初始化

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$X_i \neq Y_j$

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0						
2	0						
3	0						
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$X_i \neq Y_j$

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0						
2	0						
3	0						
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$X_i \neq Y_j$

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0						
2	0						
3	0						
4	0						
5	0						
6	0						
7	0						

$C[1, 1] = \max\{C[1, 0], C[0, 1]\}$

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U					
2						
3						
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0				
2	0						
3	0						
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U				
2						
3						
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0			
2	0						
3	0						
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U			
2						
3						
4						
5						
6						
7						

算法实例



		1	2	3	4	5	6	7
X_i		A	B	C	B	D	A	B
Y_j		B	D	C	A	B	A	

$C[[]]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1		
2	0						
3	0						
4	0						
5	0						
6	0						
7	0						

$rec[[]]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU		
2						
3						
4						
5						
6						
7						

$X_i = Y_j$

$C[1, 4] = C[0, 3] + 1$

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1		
2	0						
3	0						
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	
2						
3						
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0						
3	0						
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2						
3						
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1					
3	0						
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU					
3						
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1				
3	0						
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L				
3						
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1			
3	0						
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L			
3						
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1		
3	0						
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U		
3						
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	
3	0						
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	
3						
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0						
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3						
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1					
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U					
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1				
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U				
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2			
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU			
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2		
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L		
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0						
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4						
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1					
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU					
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1				
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U				
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2			
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U			
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2		
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U		
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0						
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5						
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1					
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U					
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2				
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU				
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2			
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U			
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2		
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U		
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0						
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6						
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1					
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U					
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2				
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U				
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2			
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U			
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3		
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU		
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0						

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
7						

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0	1					

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
7	LU					

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0	1	2				

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
7	LU	U				

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0	1	2	2			

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
7	LU	U	U			

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0	1	2	2	3		

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
7	LU	U	U	U		

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0	1	2	2	3	4	

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
7	LU	U	U	U	LU	

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0	1	2	2	3	4	4

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
7	LU	U	U	U	LU	U

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0	1	2	2	3	4	4

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
			U	U	LU	U

最长公共子序列的长度

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

--	--	--	--

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0	1	2	2	3	4	4

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
7	LU	U	U	U	LU	U

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

			A
--	--	--	---

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0	1	2	2	3	4	4

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
7	LU	U	U	U	LU	U

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

			A
--	--	--	---

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0	1	2	2	3	4	4

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
7	LU	U	U	U	LU	U

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

		B	A
--	--	---	---

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0	1	2	2	3	4	4

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
7	LU	U	U	U	LU	U

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

		B	A
--	--	---	---

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0	1	2	2	3	4	4

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
7	LU	U	U	U	LU	U

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

	C	B	A
--	---	---	---

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0	1	2	2	3	4	4

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
7	LU	U	U	U	LU	U

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

	C	B	A
--	---	---	---

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0	1	2	2	3	4	4

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
7	LU	U	U	U	LU	U

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

B	C	B	A
---	---	---	---

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0	1	2	2	3	4	4

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
7	LU	U	U	U	LU	U

算法实例



	1	2	3	4	5	6	7
X_i	A	B	C	B	D	A	B
Y_j	B	D	C	A	B	A	

B	C	B	A
---	---	---	---

$C[]$

$j \backslash i$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1
2	0	1	1	1	1	2	2
3	0	1	1	2	2	2	2
4	0	1	1	2	2	3	3
5	0	1	2	2	2	3	3
6	0	1	2	2	3	3	4
7	0	1	2	2	3	4	4

$rec[]$

$j \backslash i$	1	2	3	4	5	6
1	U	U	U	LU	L	LU
2	LU	L	L	U	LU	L
3	U	U	LU	L	U	U
4	LU	U	U	U	LU	L
5	U	LU	U	U	U	U
6	U	U	U	LU	U	LU
7	LU	U	U	U	LU	U

最长公共子序列



伪代码

- Longest-Common-Subsequence(X, Y)

输入: 两个序列 X, Y

输出: X 和 Y 的最长公共子序列

$n \leftarrow \text{length}(X)$

$m \leftarrow \text{length}(Y)$

//初始化

新建二维数组 $C[0..n, 0..m]$ 和 $rec[0..n, 0..m]$

for $i \leftarrow 0$ to n do

$C[i, 0] \leftarrow 0$

end

for $j \leftarrow 0$ to m do

$C[0, j] \leftarrow 0$

end

序列长度

伪代码

- Longest-Common-Subsequence(X, Y)

输入: 两个序列 X, Y

输出: X 和 Y 的最长公共子序列

$n \leftarrow \text{length}(X)$

$m \leftarrow \text{length}(Y)$

//初始化

新建二维数组 $C[0..n, 0..m]$ 和 $rec[0..n, 0..m]$

for $i \leftarrow 0$ to n do

 | $C[i, 0] \leftarrow 0$

end

for $j \leftarrow 0$ to m do

 | $C[0, j] \leftarrow 0$

end

初始化

伪代码



• Longest-Common-Subsequence(X, Y)

//动态规划

```
for  $i \leftarrow 1$  to  $n$  do
  for  $j \leftarrow 1$  to  $m$  do
    if  $X_i = Y_j$  then
       $C[i, j] \leftarrow C[i - 1, j - 1] + 1$ 
       $rec[i, j] \leftarrow "LU"$ 
    end
    else if  $C[i - 1, j] \geq C[i, j - 1]$  then
       $C[i, j] \leftarrow C[i - 1, j]$ 
       $rec[i, j] \leftarrow "U"$ 
    end
    else
       $C[i, j] \leftarrow C[i, j - 1]$ 
       $rec[i, j] \leftarrow "L"$ 
    end
  end
end
return  $C, rec$ 
```

依次计算子问题

伪代码



• Longest-Common-Subsequence(X, Y)

//动态规划

for $i \leftarrow 1$ to n do

for $j \leftarrow 1$ to m do

if $X_i = Y_j$ then

$C[i, j] \leftarrow C[i - 1, j - 1] + 1$

$rec[i, j] \leftarrow "LU"$

end

else if $C[i - 1, j] \geq C[i, j - 1]$ then

$C[i, j] \leftarrow C[i - 1, j]$

$rec[i, j] \leftarrow "U"$

end

else

$C[i, j] \leftarrow C[i, j - 1]$

$rec[i, j] \leftarrow "L"$

end

end

end

return C, rec

末尾相等

伪代码



• Longest-Common-Subsequence(X, Y)

//动态规划

```
for  $i \leftarrow 1$  to  $n$  do
  for  $j \leftarrow 1$  to  $m$  do
    if  $X_i = Y_j$  then
       $C[i, j] \leftarrow C[i - 1, j - 1] + 1$ 
       $rec[i, j] \leftarrow "LU"$ 
    end
    else if  $C[i - 1, j] \geq C[i, j - 1]$  then
       $C[i, j] \leftarrow C[i - 1, j]$ 
       $rec[i, j] \leftarrow "U"$ 
    end
    else
       $C[i, j] \leftarrow C[i, j - 1]$ 
       $rec[i, j] \leftarrow "L"$ 
    end
  end
end
return  $C, rec$ 
```

记录长度和决策

伪代码



• Longest-Common-Subsequence(X, Y)

//动态规划

```
for  $i \leftarrow 1$  to  $n$  do
  for  $j \leftarrow 1$  to  $m$  do
    if  $X_i = Y_j$  then
       $C[i, j] \leftarrow C[i - 1, j - 1] + 1$ 
       $rec[i, j] \leftarrow "LU"$ 
    end
    else if  $C[i - 1, j] \geq C[i, j - 1]$  then
       $C[i, j] \leftarrow C[i - 1, j]$ 
       $rec[i, j] \leftarrow "U"$ 
    end
    else
       $C[i, j] \leftarrow C[i, j - 1]$ 
       $rec[i, j] \leftarrow "L"$ 
    end
  end
end
return  $C, rec$ 
```

末尾不等



伪代码

- **Print-LCS(*rec*, *X*, *i*, *j*)**

输入: 追踪数组 rec , 序列 X , 当前位置 i 和 j

输出: $X[1..i]$ 和 $Y[1..j]$ 的最长公共子序列

if $i = 0$ or $j = 0$ then

 return NULL

end

if $rec[i, j] = \text{"LU"}$ then

 Print-LCS($rec, X, i - 1, j - 1$)

 print x_i

end

else if $rec[i, j] = \text{"U"}$ then

 Print-LCS($rec, X, i - 1, j$)

end

else

 Print-LCS($rec, X, i, j - 1$)

end

倒序追踪方案



伪代码

- **Print-LCS(rec, X, i, j)**

输入: 追踪数组 rec , 序列 X , 当前位置 i 和 j

输出: $X[1..i]$ 和 $Y[1..j]$ 的最长公共子序列

```
if  $i = 0$  or  $j = 0$  then  
| return NULL
```

递归终止：序列长度为0

```
end
```

```
if  $rec[i, j] = \text{"LU"}$  then
```

```
| Print-LCS( $rec, X, i - 1, j - 1$ )  
| print  $x_i$ 
```

```
end
```

```
else if  $rec[i, j] = \text{"U"}$  then
```

```
| Print-LCS( $rec, X, i - 1, j$ )
```

```
end
```

```
else
```

```
| Print-LCS( $rec, X, i, j - 1$ )
```

```
end
```



伪代码

- **Print-LCS(*rec*, *X*, *i*, *j*)**

输入: 追踪数组 rec , 序列 X , 当前位置 i 和 j

输出: $X[1..i]$ 和 $Y[1..j]$ 的最长公共子序列

if $i = 0$ or $j = 0$ then

 return NULL

end

if $rec[i, j] = \text{"LU"}$ then

 Print-LCS($rec, X, i - 1, j - 1$)

 print x_i

end

else if $rec[i, j] = \text{"U"}$ then

 Print-LCS($rec, X, i - 1, j$)

end

else

 Print-LCS($rec, X, i, j - 1$)

end

追踪方案：末尾相等



伪代码

- **Print-LCS(*rec*, *X*, *i*, *j*)**

输入: 追踪数组*rec*, 序列*X*, 当前位置*i*和*j*

输出: *X*[1..*i*]和*Y*[1..*j*]的最长公共子序列

if *i* = 0 or *j* = 0 then

 | return NULL

end

if *rec*[*i*, *j*] = "LU" then

 | Print-LCS(*rec*, *X*, *i* - 1, *j* - 1)

 | print *x_i*

end

else if *rec*[*i*, *j*] = "U" then

 | Print-LCS(*rec*, *X*, *i* - 1, *j*)

end

else

 | Print-LCS(*rec*, *X*, *i*, *j* - 1)

end

追踪方案：末尾不等



时间复杂度分析

● Longest-Common-Subsequence(X, Y)

//动态规划

```
for  $i \leftarrow 1$  to  $n$  do
  for  $j \leftarrow 1$  to  $m$  do
    if  $X_i = Y_j$  then
       $C[i, j] \leftarrow C[i - 1, j - 1] + 1$ 
       $rec[i, j] \leftarrow "LU"$ 
    end
    else if  $C[i - 1, j] \geq C[i, j - 1]$  then
       $C[i, j] \leftarrow C[i - 1, j]$ 
       $rec[i, j] \leftarrow "U"$ 
    end
    else
       $C[i, j] \leftarrow C[i, j - 1]$ 
       $rec[i, j] \leftarrow "L"$ 
    end
  end
end
return  $C, rec$ 
```

时间复杂度： $O(n \cdot m)$



谢谢

