

# Design and Analysis of Algorithms

## Part II: Dynamic Programming

### Lecture 14: Minimum Edit Distance

---

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

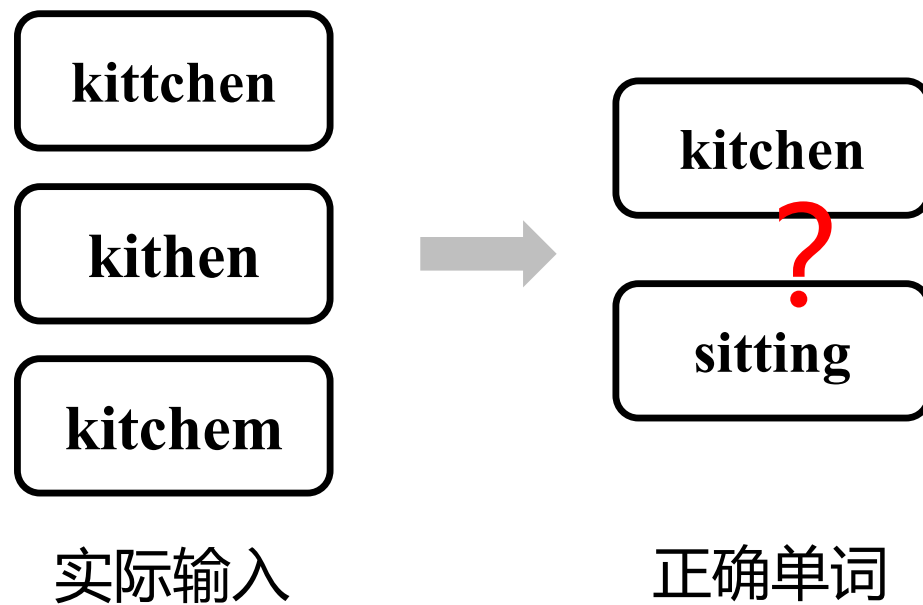
---

- 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



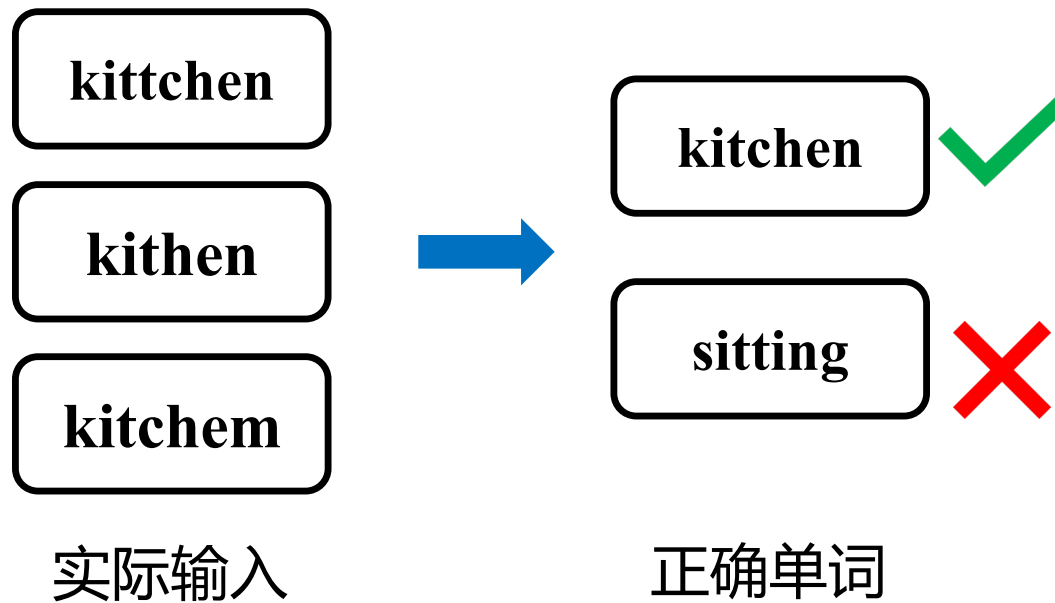
- 输入法自动更正



# 问题背景

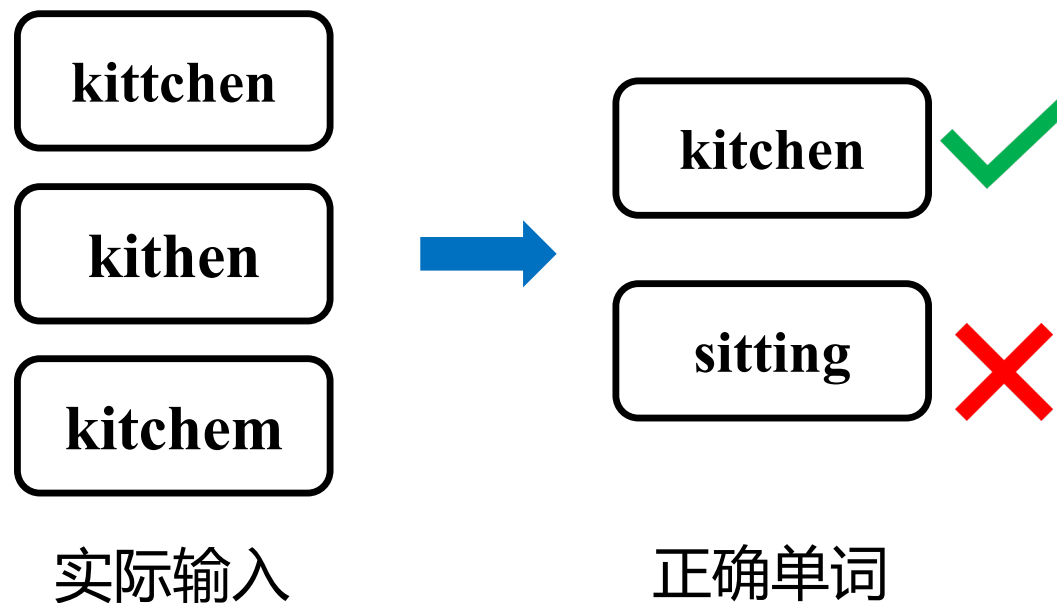


- 输入法自动更正



## 问题背景

- 输入法自动更正



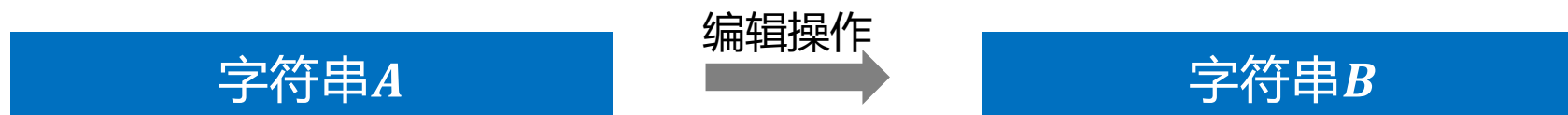
问题：如何衡量序列的相似程度？



# 编辑操作

---

- 基本思想

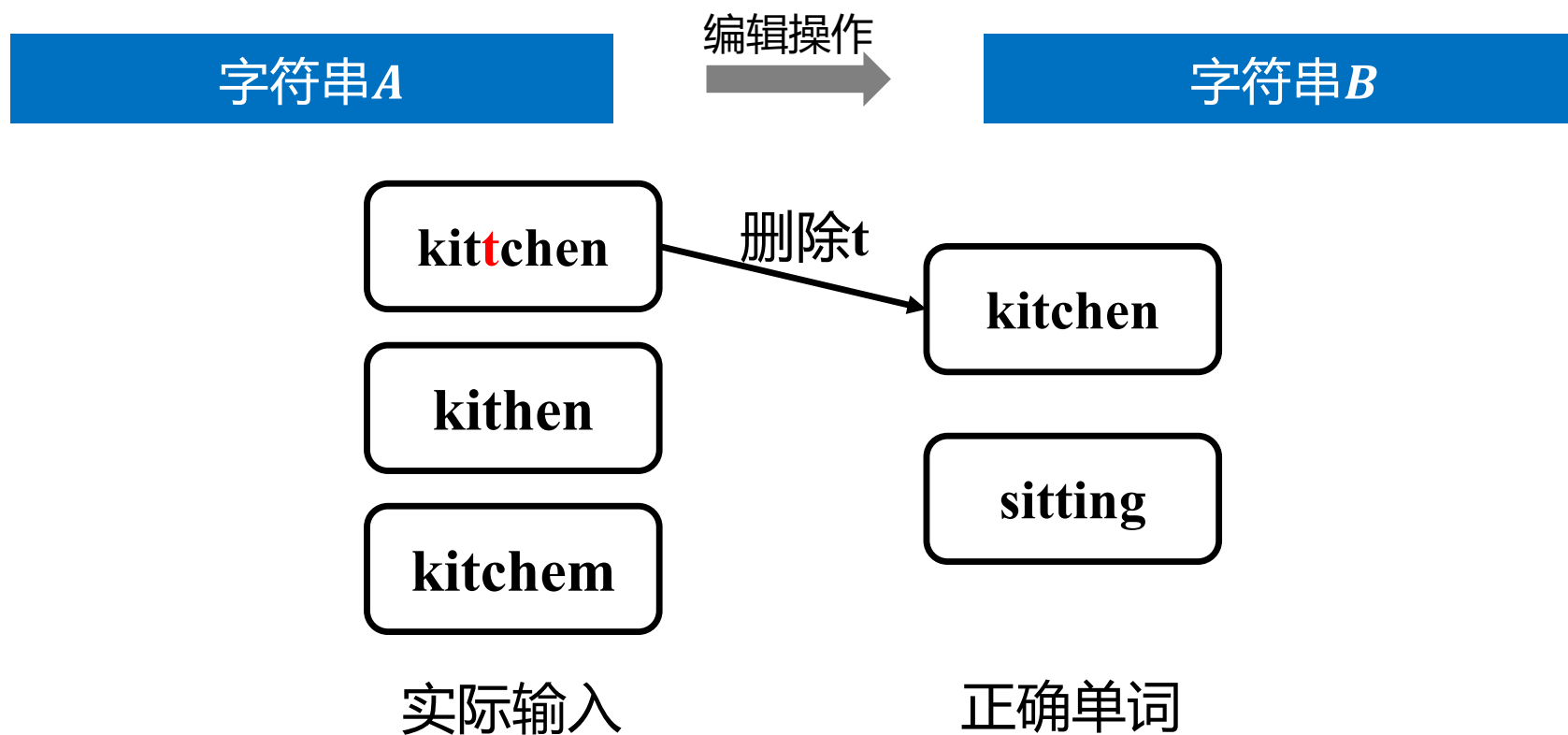


- 编辑操作：删除、插入、替换

# 编辑操作



- 基本思想



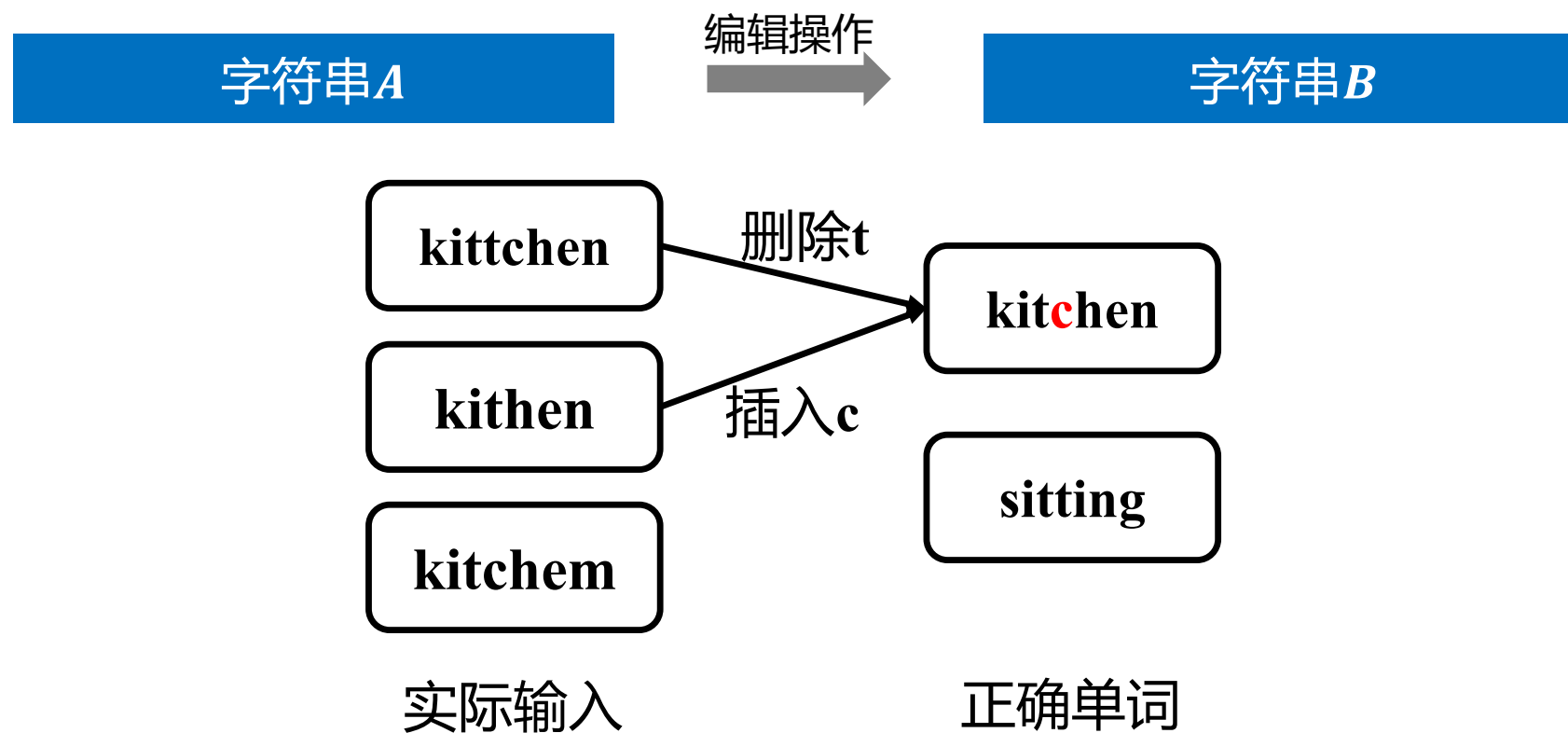
- 编辑操作：删除、插入、替换



# 编辑操作



- 基本思想

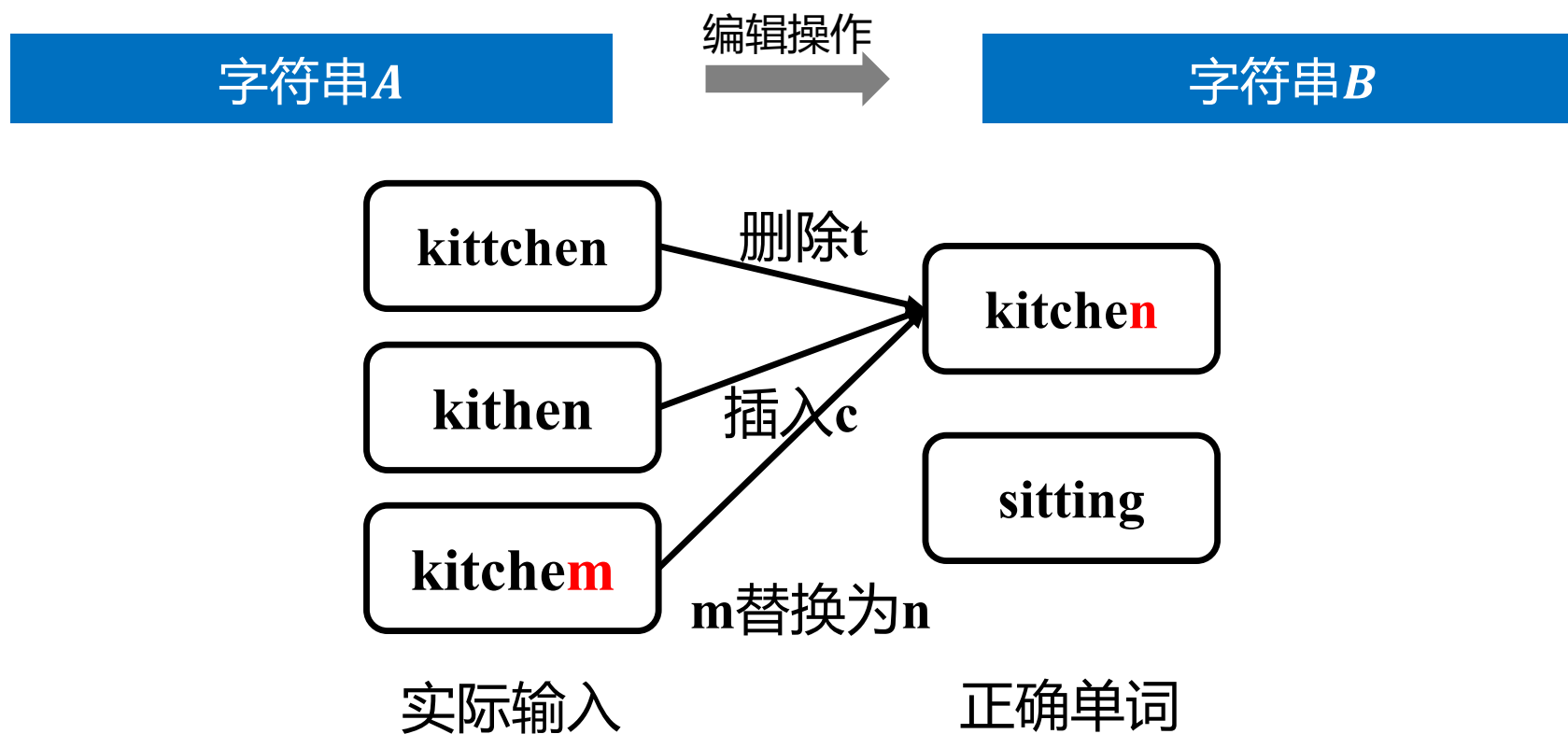


- 编辑操作：删除、插入、替换

# 编辑操作



- 基本思想

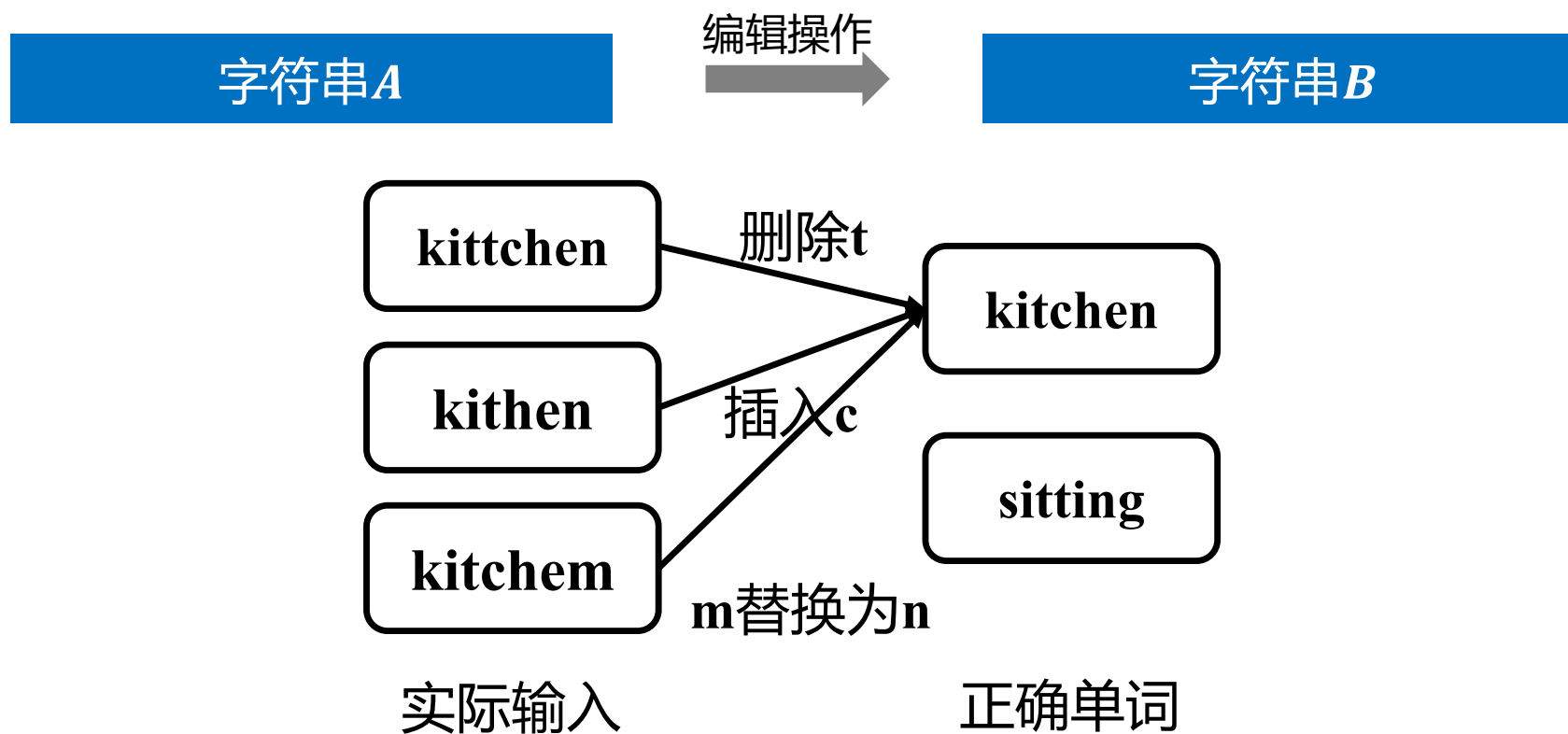


- 编辑操作：删除、插入、替换



# 编辑操作

- 基本思想

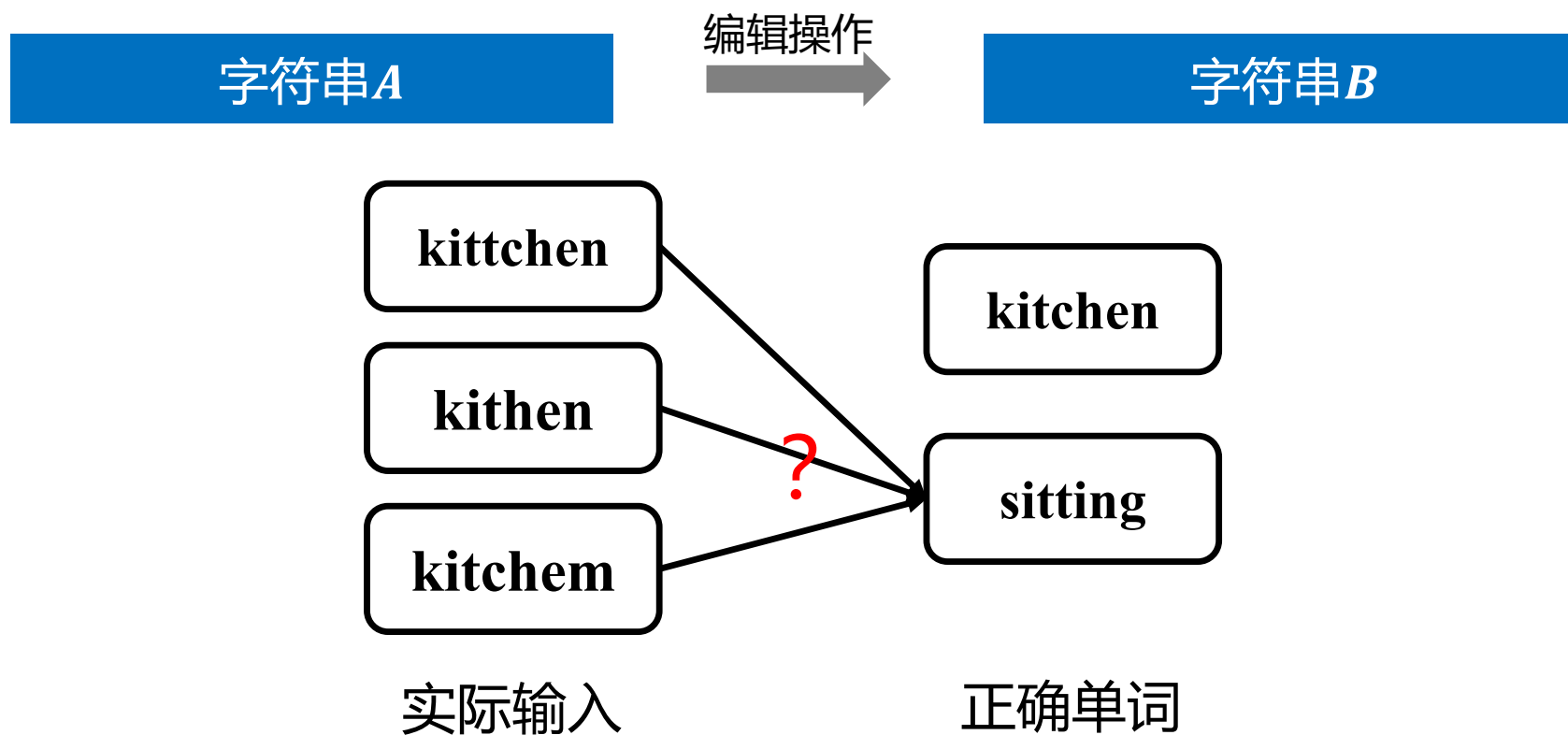


- 编辑操作：删除、插入、替换

# 编辑操作

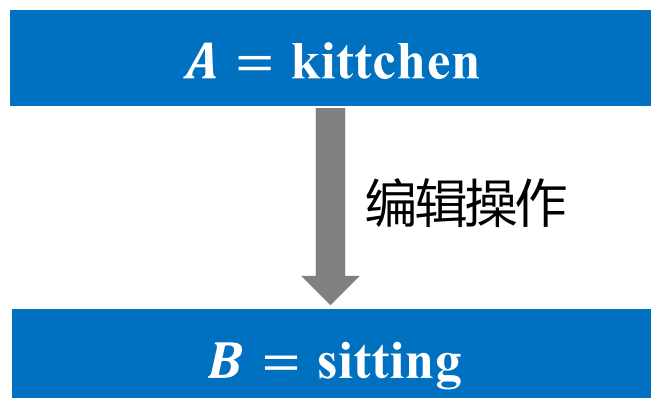


- 基本思想



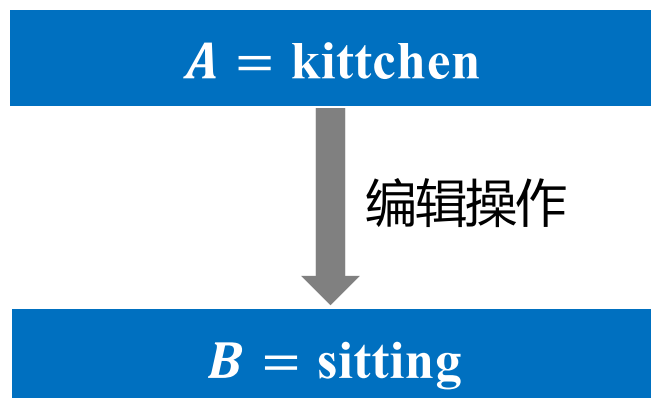
- 编辑操作：删除、插入、替换

# 编辑操作示例



操作名称	操作示例
删除	<b>kittchen</b> → kitchen
插入	kithen → kit <b>ch</b> en
替换	kitchem → kitchen <b>n</b>

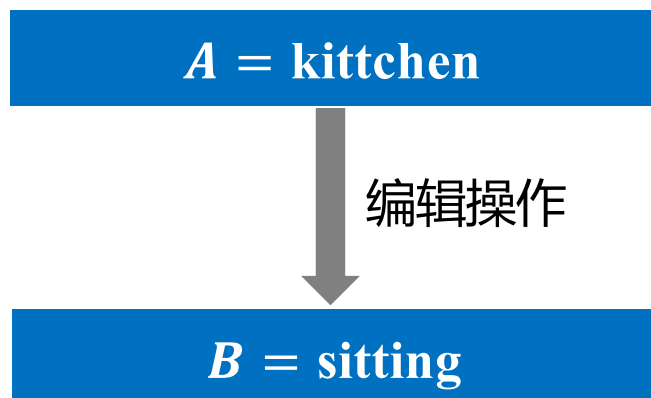
# 编辑操作示例



操作名称	操作示例
删除	kitt <b>ch</b> en → kitchen
插入	kithen → kit <b>ch</b> en
替换	kitchem → kitchen <b>n</b>



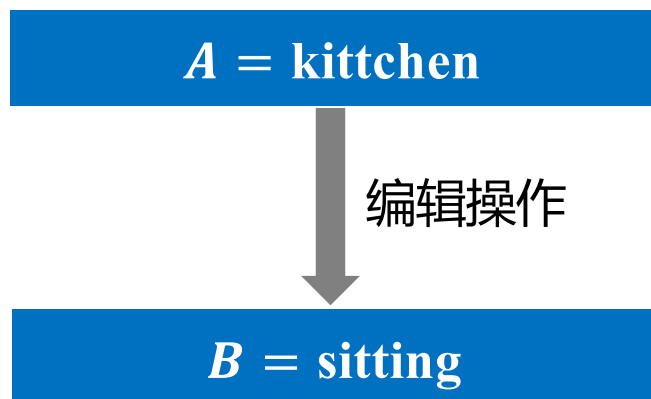
# 编辑操作示例



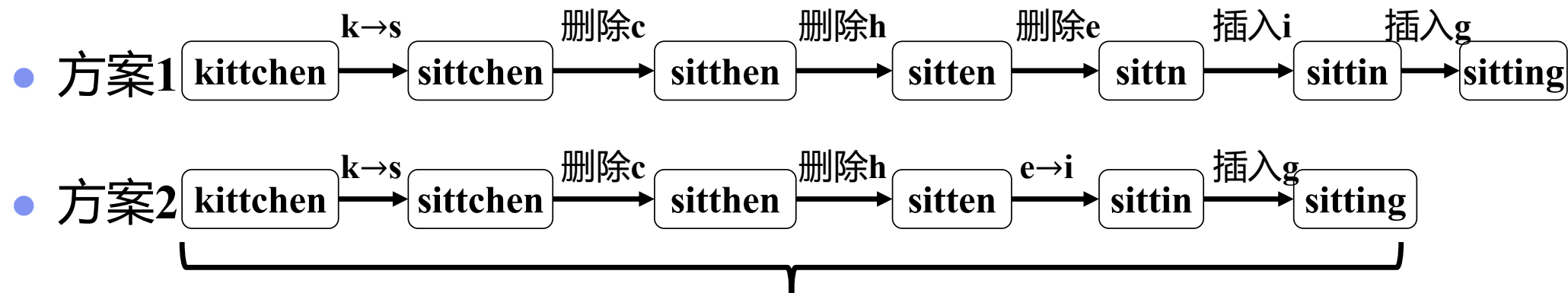
操作名称	操作示例
删除	kitt <b>ch</b> en → kitchen
插入	kithen → kit <b>ch</b> en
替换	kitchem → kitchen <b>n</b>

- 方案1 <sup>6次</sup>  
kitt**ch**en  $\xrightarrow{k \rightarrow s}$  sitt**ch**en  $\xrightarrow{\text{删除}c}$  sitt**h**en  $\xrightarrow{\text{删除}h}$  sitt**e**n  $\xrightarrow{\text{删除}e}$  sitt**n**  $\xrightarrow{\text{插入}i}$  sitt**i**n  $\xrightarrow{\text{插入}g}$  sitt**ing**
- 方案2 <sup>5次</sup>  
kitt**ch**en  $\xrightarrow{k \rightarrow s}$  sitt**ch**en  $\xrightarrow{\text{删除}c}$  sitt**h**en  $\xrightarrow{\text{删除}h}$  sitt**e**n  $\xrightarrow{e \rightarrow i}$  sitt**i**n  $\xrightarrow{\text{插入}g}$  sitt**ing**

# 编辑操作示例



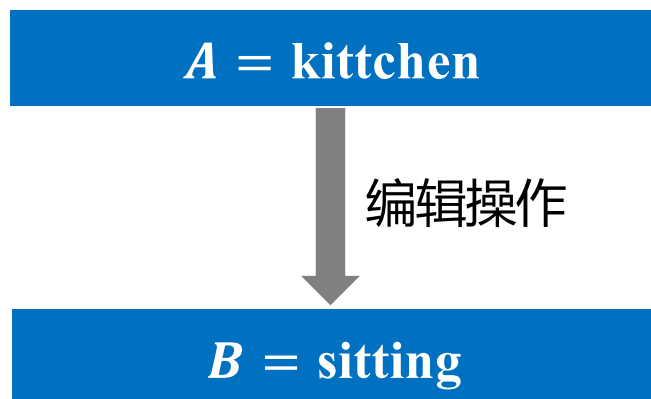
操作名称	操作示例
删除	kitt <b>ch</b> en → kitchen
插入	kithen → kit <b>ch</b> en
替换	kitchem → kitchen <b>n</b>



编辑距离：编辑操作次数



# 编辑操作示例



操作名称	操作示例
删除	kitt <b>ch</b> en → kittchen
插入	kithen → kitt <b>ch</b> en
替换	kitchem → kittchen <b>n</b>

- 方案1: kittchen  $\xrightarrow{k \rightarrow s}$  sittchen  $\xrightarrow{\text{删除}c}$  sitthen  $\xrightarrow{\text{删除}h}$  sitten  $\xrightarrow{\text{删除}e}$  sittn  $\xrightarrow{\text{插入}i}$  sittin  $\xrightarrow{\text{插入}g}$  sitting
- 方案2: kittchen  $\xrightarrow{k \rightarrow s}$  sittchen  $\xrightarrow{\text{删除}c}$  sitthen  $\xrightarrow{\text{删除}h}$  sitten  $\xrightarrow{e \rightarrow i}$  sittin  $\xrightarrow{\text{插入}g}$  sitting

问题：如何求出最少的编辑操作数（最小编辑距离）？

# 问题定义



## 编辑距离问题

### Minimum Edit Distance, MED

输入

- 长度为 $n$ 的字符串 $s$  , 长度为 $m$ 的字符串 $t$

输出

- 求出一组编辑操作 $O = \langle e_1, e_2, \dots, e_d \rangle$  , 令

$$\min |O|$$

优化目标

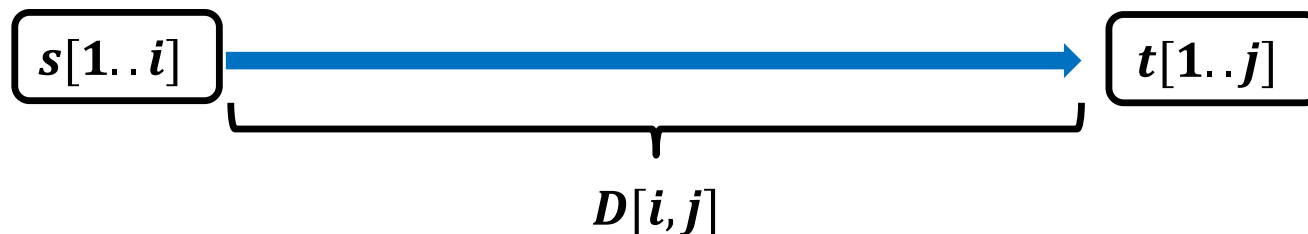
$s.t.$  字符串 $s$ 经过 $O$ 的操作后满足 $s = t$

约束条件

# 问题结构分析

- 给出问题表示

- $D[i, j]$  : 字符串  $s[1..i]$  变为  $t[1..j]$  的最小编辑距离



- 明确原始问题

- $D[n, m]$  : 字符串  $s[1..n]$  变为  $t[1..m]$  的最小编辑距离

问题结构分析



递推关系建立



自底向上计算



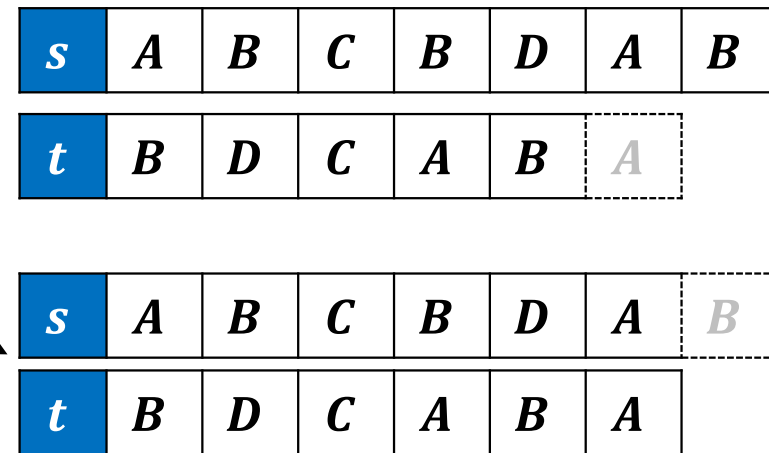
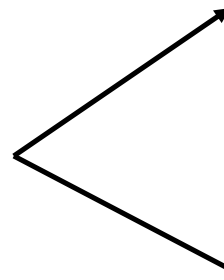
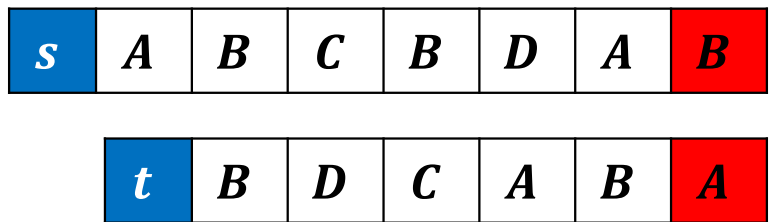
最优方案追踪



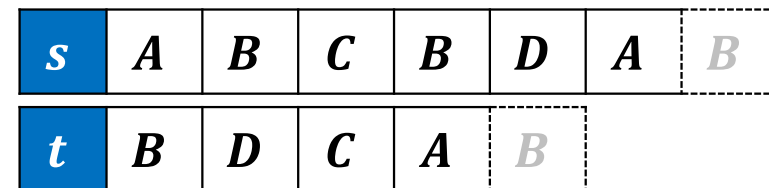
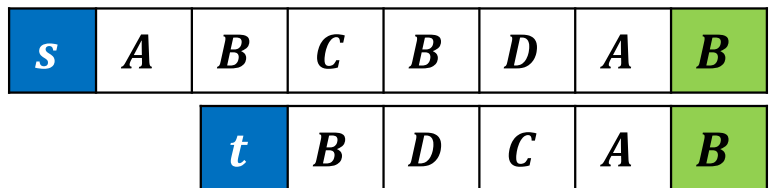
# 递推关系建立：回顾与启发

- 最长公共子序列

- 如果  $s_i \neq t_j$



- 如果  $s_i = t_j$



考察末尾元素

# 递推关系建立

- 考察末尾元素

- 删除

<i>s</i>	A	B	C	B	D	A	<b>B</b>
----------	---	---	---	---	---	---	----------

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



?

- 插入

<i>s</i>	A	B	C	B	D	A	B	?
----------	---	---	---	---	---	---	---	---

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



?

- 替换

								?
<i>s</i>	A	B	C	B	D	A	<b>B</b>	

<i>t</i>	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

- 考察末尾元素：删除

<b>s</b>	A	B	C	B	D	A	<b>B</b>
----------	---	---	---	---	---	---	----------

<b>t</b>	B	D	C	A	B	A
----------	---	---	---	---	---	---



<b>s</b>	A	B	C	B	D	A	B
----------	---	---	---	---	---	---	---

<b>t</b>	B	D	C	A	B	A
----------	---	---	---	---	---	---

问题结构分析



递推关系建立



自底向上计算

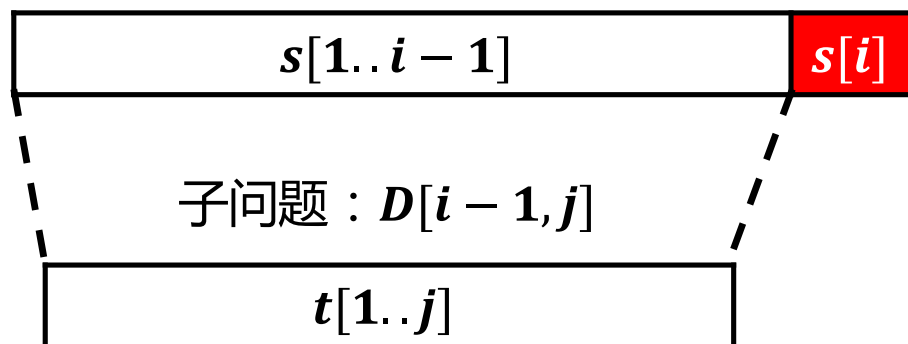
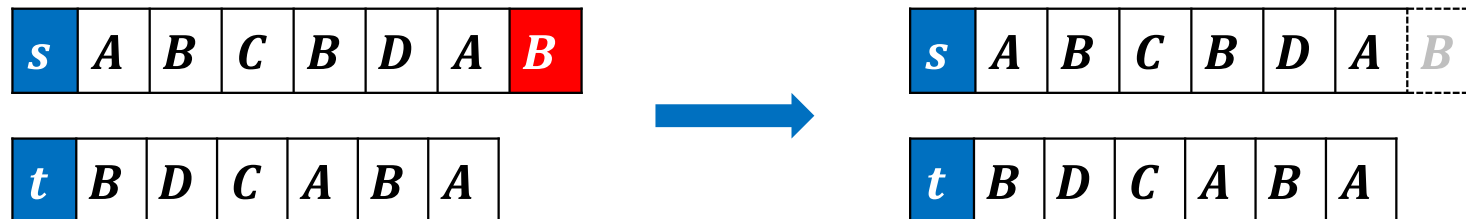


最优方案追踪



## 递推关系建立：分析最优（子）结构

- 考察末尾元素：删除



问题结构分析

递推关系建立

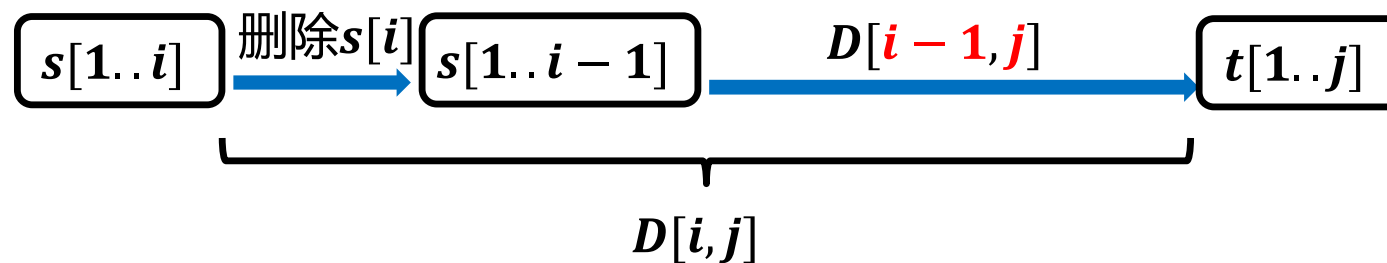
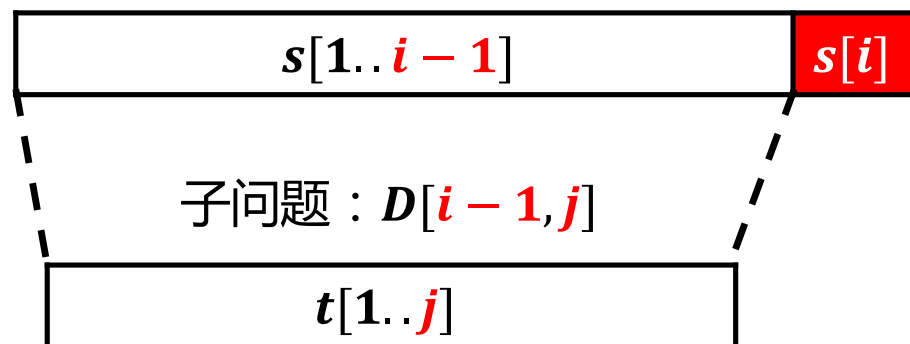
自底向上计算

最优方案追踪



## 递推关系建立：分析最优（子）结构

- 考察末尾元素：删除



问题结构分析

递推关系建立

自底向上计算

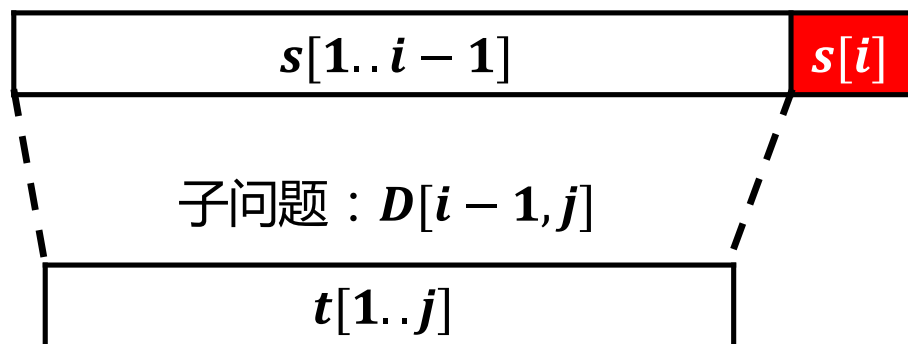
最优方案追踪





## 递推关系建立：分析最优（子）结构

- 考察末尾元素：删除



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

问题结构分析

递推关系建立

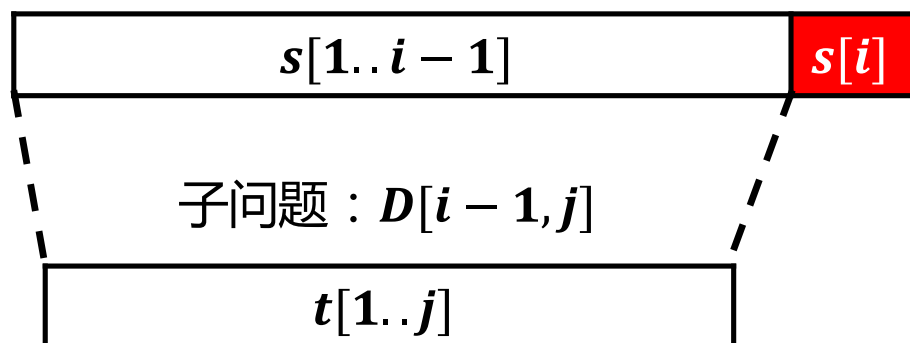
自底向上计算

最优方案追踪



## 递推关系建立：分析最优（子）结构

- 考察末尾元素：删除



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

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

- 考察末尾元素：插入

<i>s</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>B</i>	<i>D</i>	<i>A</i>	<i>B</i>	?
----------	----------	----------	----------	----------	----------	----------	----------	---



<i>t</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</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

- 考察末尾元素：插入

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

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



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

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

问题结构分析



递推关系建立



自底向上计算

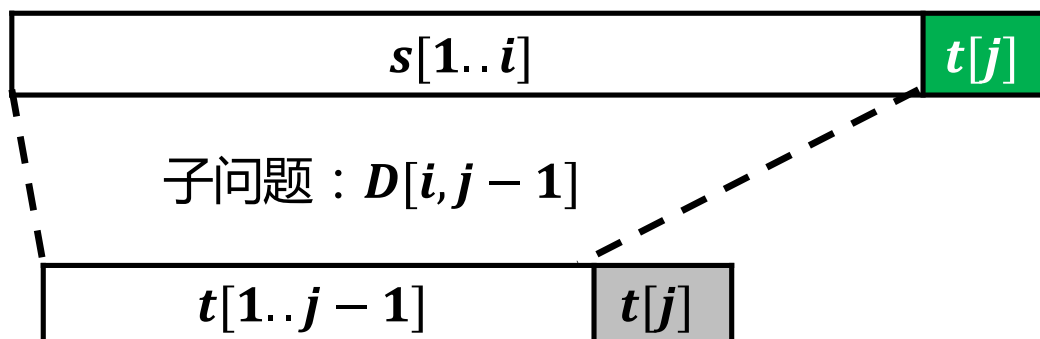
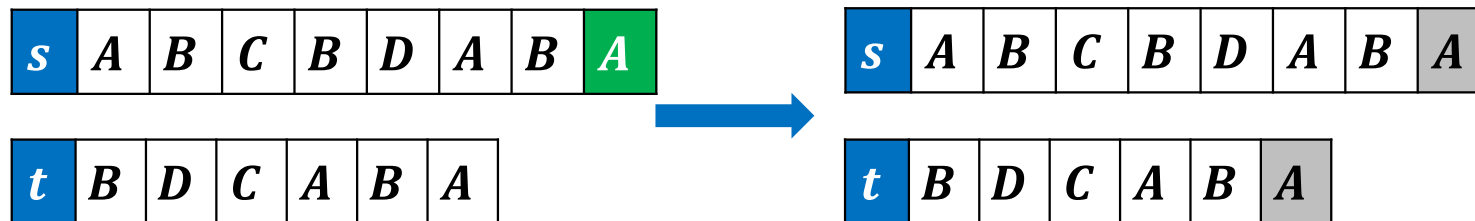


最优方案追踪



## 递推关系建立：分析最优（子）结构

- 考察末尾元素：插入



问题结构分析

递推关系建立

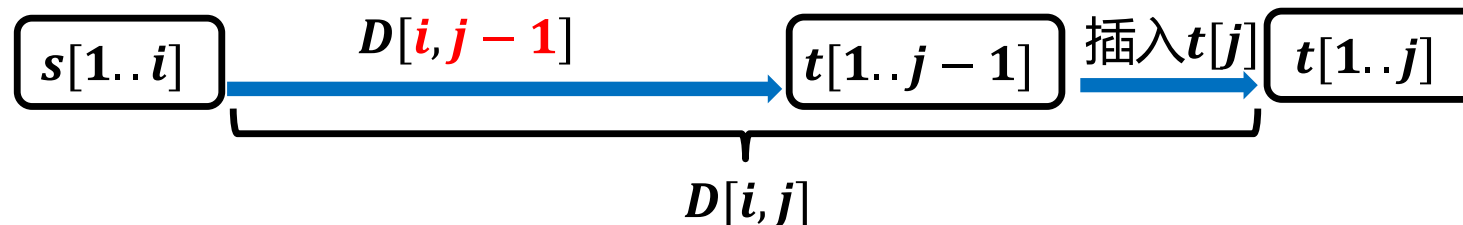
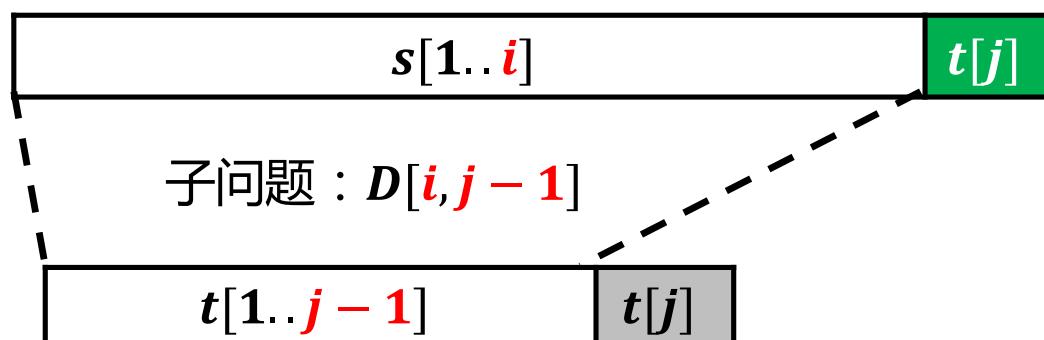
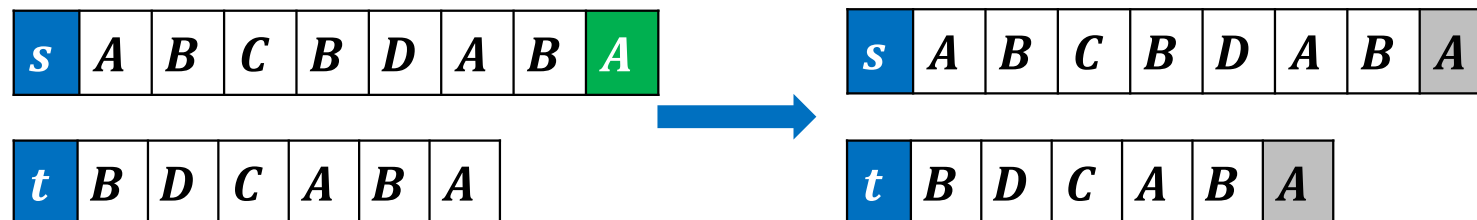
自底向上计算

最优方案追踪



# 递推关系建立：分析最优（子）结构

- 考察末尾元素：插入



问题结构分析

递推关系建立

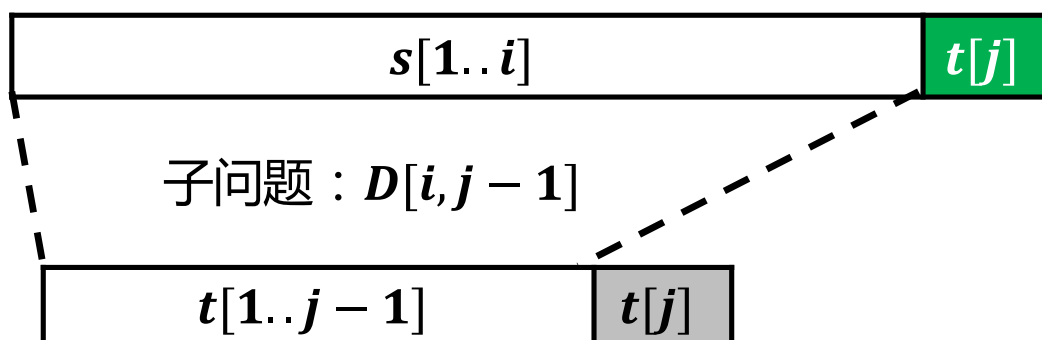
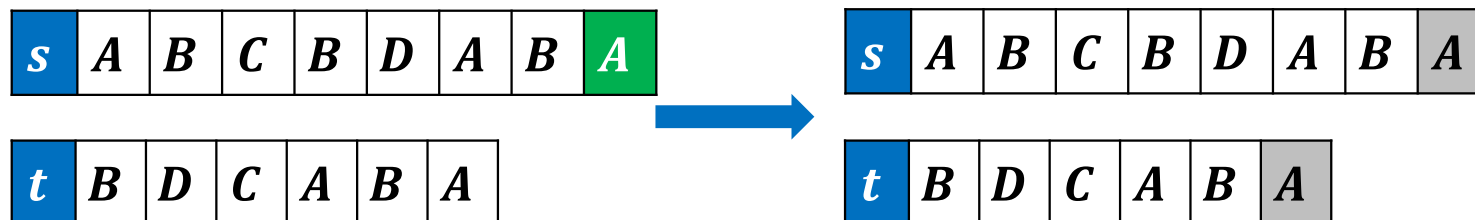
自底向上计算

最优方案追踪



## 递推关系建立：分析最优（子）结构

- 考察末尾元素：插入



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

问题结构分析

递推关系建立

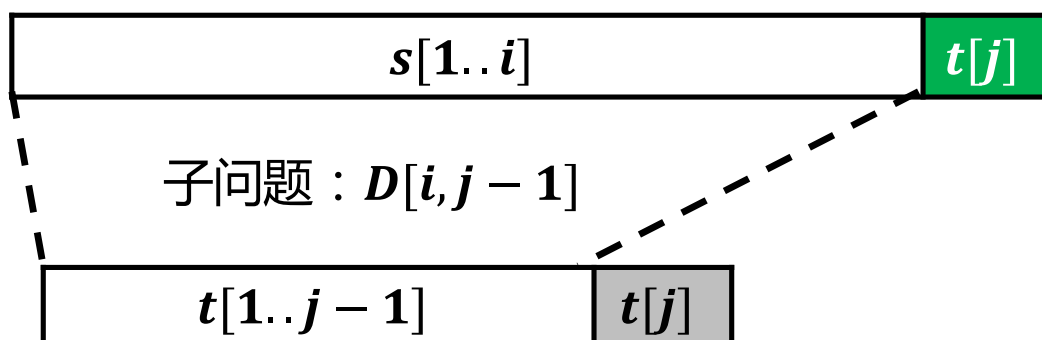
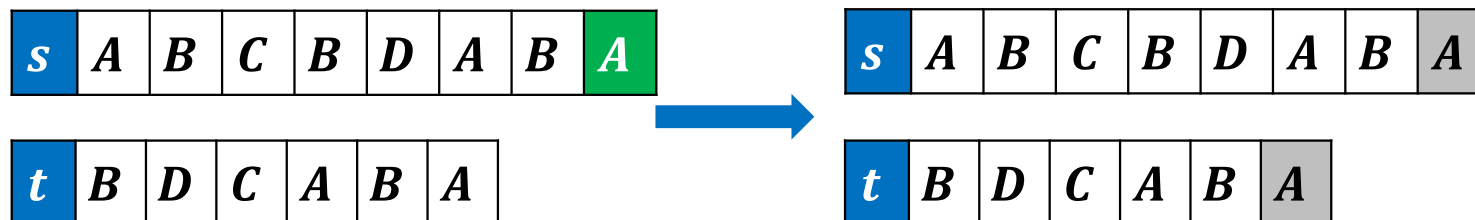
自底向上计算

最优方案追踪



## 递推关系建立：分析最优（子）结构

- 考察末尾元素：插入



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

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：分析最优（子）结构

- 考察末尾元素：替换

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



<i>t</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</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

- 考察末尾元素：替换

<i>s</i>	A	B	C	B	D	A	?
----------	---	---	---	---	---	---	---



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

问题结构分析



递推关系建立



自底向上计算

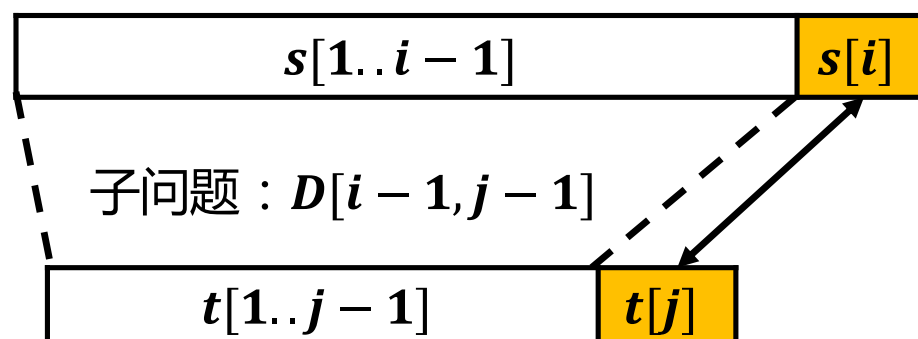


最优方案追踪



# 递推关系建立：分析最优（子）结构

- 考察末尾元素：替换



问题结构分析

递推关系建立

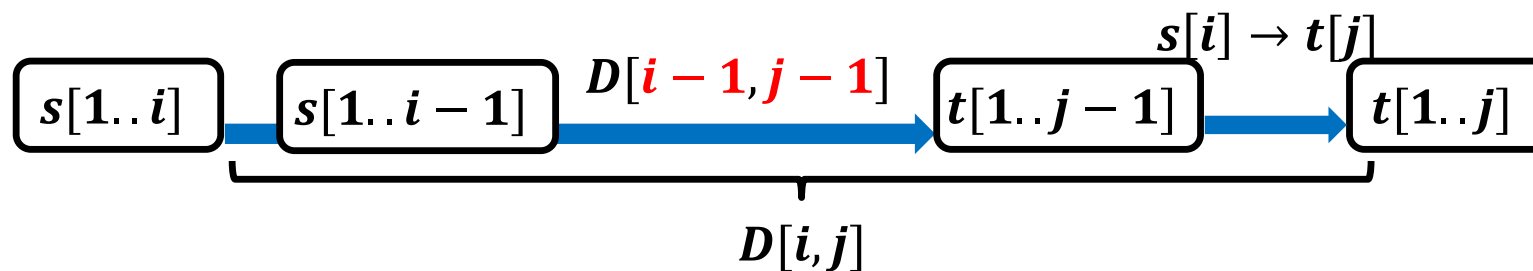
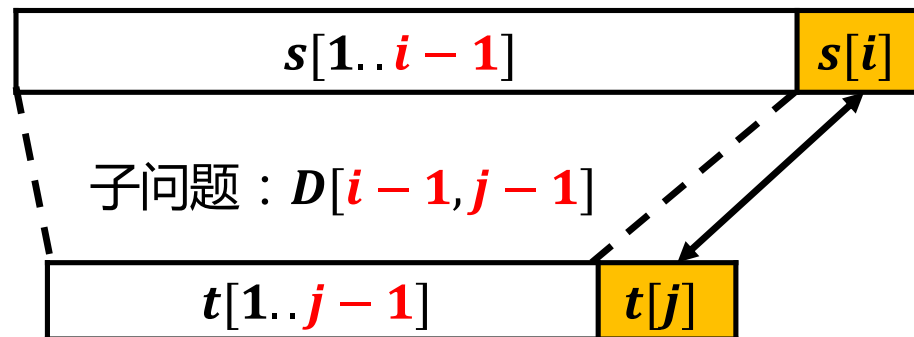
自底向上计算

最优方案追踪



# 递推关系建立：分析最优（子）结构

- 考察末尾元素：替换



问题结构分析

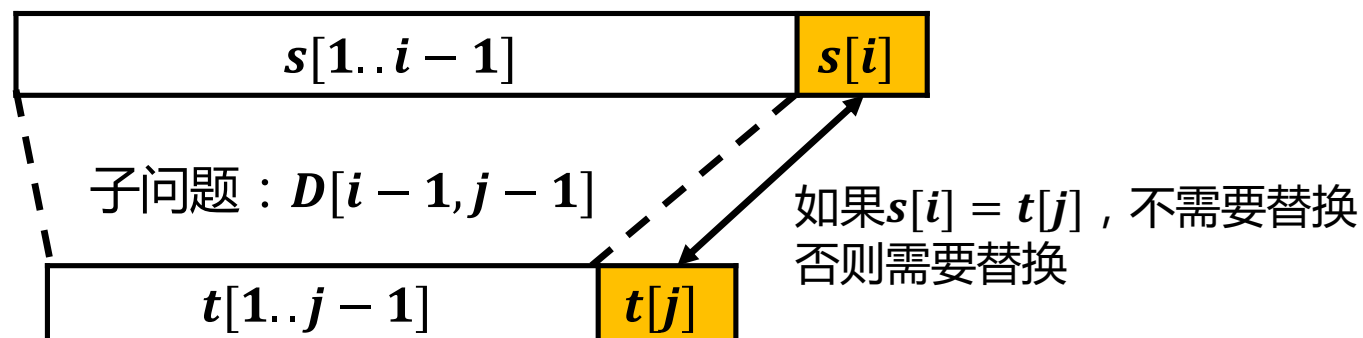
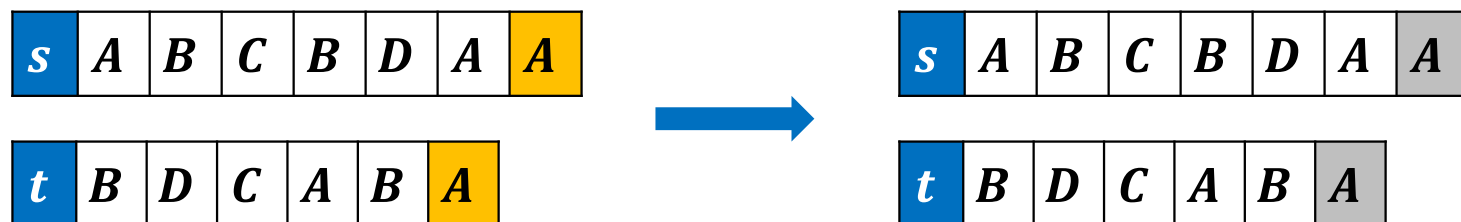
递推关系建立

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

- 考察末尾元素：替换



- $$D[i, j] = D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases}$$

问题结构分析

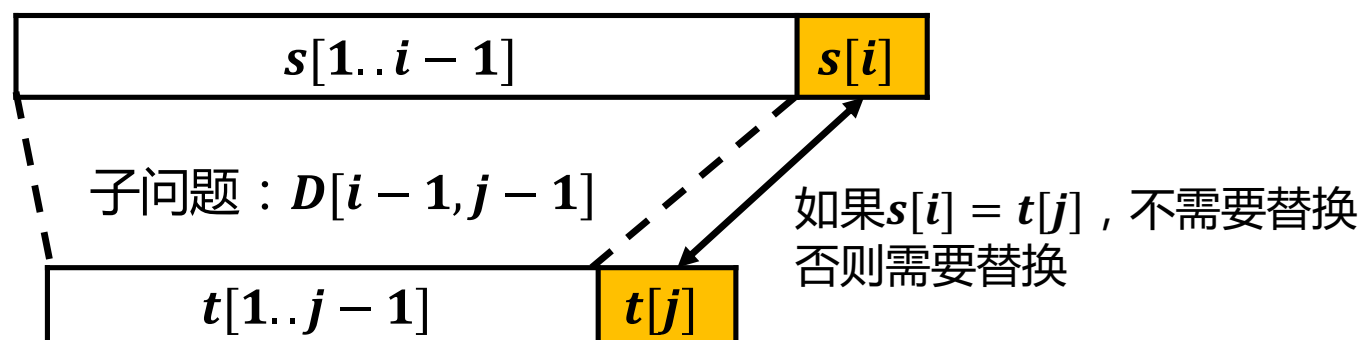
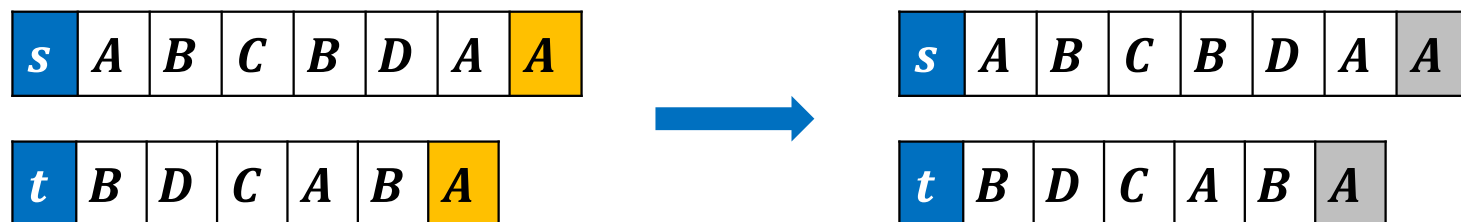
递推关系建立

自底向上计算

最优方案追踪

# 递推关系建立：分析最优（子）结构

## 考察末尾元素：替换



$$D[i, j] = D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases}$$

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

- 综合上面三种方式

- $$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

删除  
插入  
替换

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

- 最小编辑距离 vs. 最长公共子序列

- $$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

删除  
插入  
替换

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

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

- $D[i, 0] = i$ 
  - 把长度为 $i$ 的串变为空串至少需要 $i$ 次操作（删除）
- $D[0, j] = j$ 
  - 把空串变为长度为 $j$ 的串至少需要 $j$ 次操作（插入）

$i \backslash j$	0	1	2	...	$m-1$	$m$
0	0	1	2	...	$m-1$	$m$
1	1					
2	2					
...	...					
$n-1$	$n-1$					
$n$	$n$					

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

- 递推公式

- $$D[i, j] = \min \begin{cases} D[i-1, j] + 1 & \text{删除} \\ D[i, j-1] + 1 & \text{插入} \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} & \text{替换} \end{cases}$$

$i \backslash j$	0	1	2	...	$m-1$	$m$
0	0	1	2	...	$m-1$	$m$
1	1					
2	2					
...	...					
$n-1$	$n-1$					
$n$	$n$					

Diagram illustrating the dynamic programming table structure and the calculation of  $D[i, j]$  based on the recurrence relation. The table shows indices  $i$  (rows) and  $j$  (columns). The value  $D[i, j]$  is determined by the minimum of three cases: deletion ( $D[i-1, j] + 1$ ), insertion ( $D[i, j-1] + 1$ ), and replacement ( $D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases}$ ). Arrows indicate the dependencies for calculating  $D[i, j]$  from its neighbors.

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：依次计算问题

## 递推公式

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 & \text{删除} \\ D[i, j-1] + 1 & \text{插入} \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} & \text{替换} \end{cases}$$

$i \backslash j$	0	1	2	...	$m-1$	$m$
0	0	1	2	...	$m-1$	$m$
1	1					
2	2					
...	...					
$n-1$	$n-1$					
$n$	$n$					

问题结构分析



递推关系建立



自底向上计算



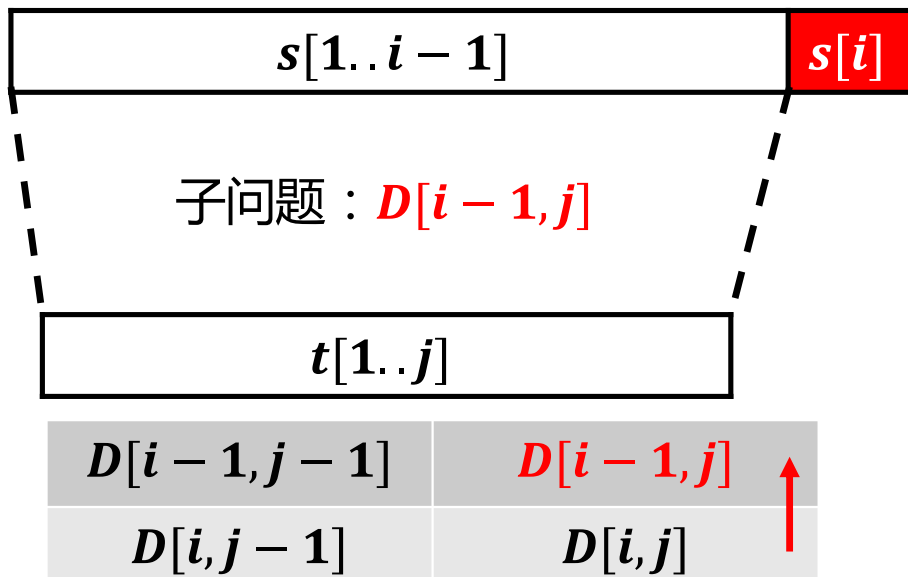
最优方案追踪





## 最优方案追踪：记录决策过程

- 追踪数组  $Rec$ ，记录子问题来源



$Rec[i, j]$	子问题来源	操作
$U$	上侧，即 $D[i-1, j]$	删除 $s[i]$

问题结构分析



递推关系建立



自底向上计算

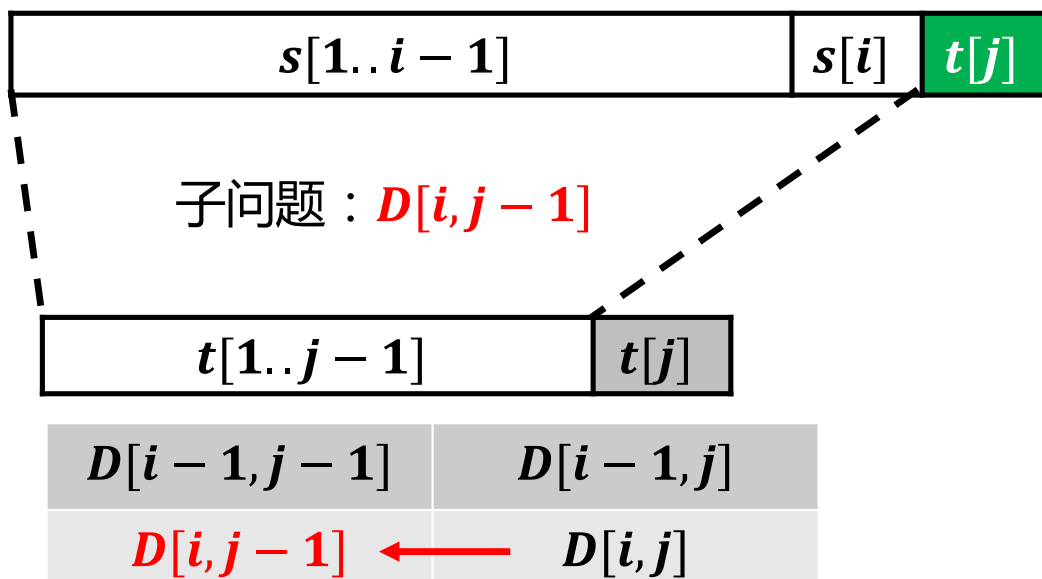


最优方案追踪



# 最优方案追踪：记录决策过程

- 追踪数组  $Rec$ ，记录子问题来源



问题结构分析



递推关系建立



自底向上计算



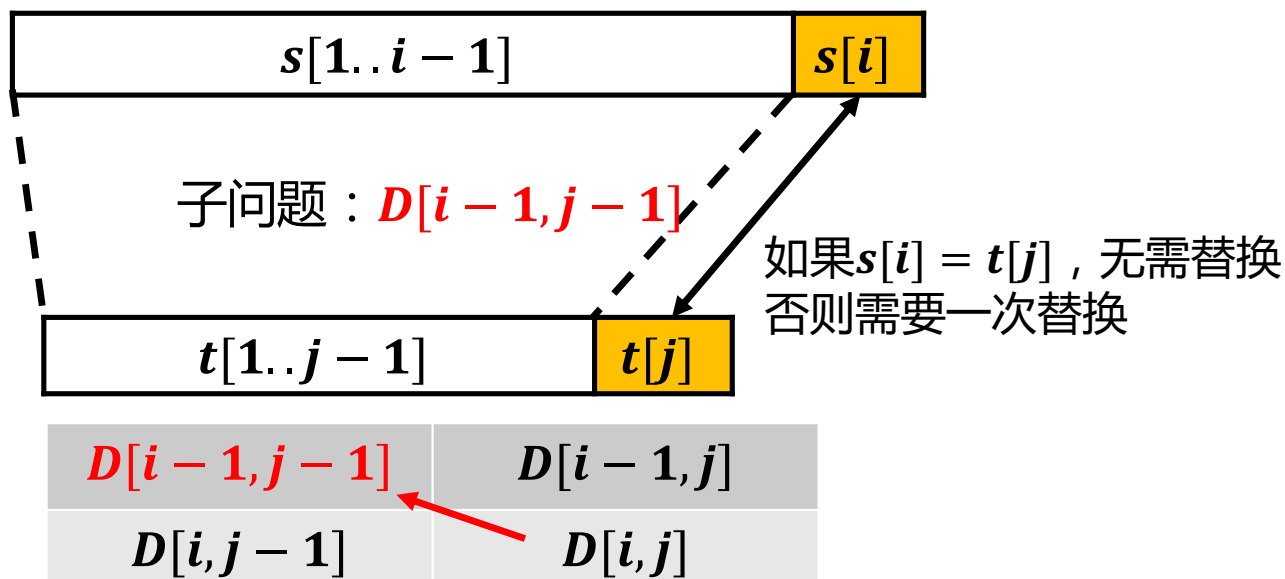
最优方案追踪

$Rec[i, j]$	子问题来源	操作
$U$	上侧，即 $D[i - 1, j]$	删除 $s[i]$
$L$	左侧，即 $D[i, j - 1]$	插入 $t[j]$



# 最优方案追踪：记录决策过程

- 追踪数组  $Rec$ ，记录子问题来源



问题结构分析

递推关系建立

自底向上计算

最优方案追踪

$Rec[i, j]$	子问题来源	操作
$U$	上侧，即 $D[i-1, j]$	删除 $s[i]$
$L$	左侧，即 $D[i, j-1]$	插入 $t[j]$
$LU$	左上，即 $D[i-1, j-1]$	用 $t[j]$ 替换 $s[i]$ /无操作

# 最优方案追踪：输出最优方案

- 根据数组  $Rec$ ，输出最少编辑操作

$i \backslash j$	0	1	2	...	$m$
0					
1					
...					
$n$					

$Rec[i, j] = L$   
 $Rec[i, j] = LU$   
 $Rec[i, j] = U$

问题结构分析

递推关系建立

自底向上计算

最优方案追踪

$Rec[i, j]$	子问题来源	操作
$U$	上侧，即 $D[i - 1, j]$	删除 $s[i]$
$L$	左侧，即 $D[i, j - 1]$	插入 $t[j]$
$LU$	左上，即 $D[i - 1, j - 1]$	用 $t[j]$ 替换 $s[i]$ /无操作

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1						
2	2						
3	3						
4	4						
5	5						
6	6						
7	7						

初始化

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U						
2	U						
3	U						
4	U						
5	U						
6	U						
7	U						



# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1						
2	2						
3	3						
4	4						
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U						
2	U						
3	U						
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 & \text{删除A} \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

$D$

$D[i - 1, j]$

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

*Rec*

<i>i \ j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U						
2	U						
3	U						
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 & \text{删除A} \\ D[i, j-1] + 1 & \text{插入B} \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

**D**

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1						
2	2						
3	3						
4	4						
5	5						
6	6						
7	7						

Annotations:  $D[i-1, j]$  points to cell (0,1) and  $D[i, j-1]$  points to cell (2,1).

**Rec**

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U						
2	U						
3	U						
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A			

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 & \text{删除A} \\ D[i, j-1] + 1 & \text{插入B} \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} & \text{A替换为B} \end{cases}$$

**D**

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1						
2	2						
3	3						
4	4						
5	5						
6	6						
7	7						

**Rec**

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U						
2	U						
3	U						
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	B

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

**D**

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	4	5	6
2	2	2	3	4	5	6	7
3	3	3	4	5	6	7	8
4	4	4	5	6	7	8	9
5	5	5	6	7	8	9	10
6	6	6	7	8	9	10	11
7	7	7	8	9	10	11	12

**Rec**

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU					
2	U						
3	U						
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

**D**

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2				
2	2						
3	3						
4	4						
5	5						
6	6						
7	7						

**Rec**

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU				
2	U						
3	U						
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3			
2	2						
3	3						
4	4						
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU			
2	U						
3	U						
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
s	A	B	C	B	D	A	B
t	B	D	C	A	B	A	B

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

$D$   $D[i-1, j]$

无需替换

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



# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	
2	2						
3	3						
4	4						
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	
2	U						
3	U						
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2						
3	3						
4	4						
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U						
3	U						
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1					
3	3						
4	4						
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU					
3	U						
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2				
3	3						
4	4						
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU				
3	U						
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3			
3	3						
4	4						
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU			
3	U						
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4		
3	3						
4	4						
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU		
3	U						
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	
3	3						
4	4						
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	
3	U						
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3						
4	4						
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U						
4	U						
5	U						
6	U						
7	U						



# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2					
4	4						
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U					
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2				
4	4						
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU				
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2			
4	4						
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU			
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3		
4	4						
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L		
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	
4	4						
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4						
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U						
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3					
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU					
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3				
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU				
5	U						
6	U						
7	U						



# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3			
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU			
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3		
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU		
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5						
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U						
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3				
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU				
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	4			
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU			
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	4	4		
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU		
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	4	4	4	
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	
6	U						
7	U						



# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	4	4	4	4
6	6						
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U						
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	4	4	4	4
6	6	5					
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U					
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	4	4	4	4
6	6	5	4				
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U				
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	4	4	4	4
6	6	5	4	4			
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU			
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	4	4	4	4
6	6	5	4	4	4		
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU		
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	4	4	4	4
6	6	5	4	4	4	5	
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	4	4	4	4
6	6	5	4	4	4	5	4
7	7						

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	LU
7	U						

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	4	4	4	4
6	6	5	4	4	4	5	4
7	7	6					

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	LU
7	U	LU					



# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	4	4	4	4
6	6	5	4	4	4	5	4
7	7	6	5				

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	LU
7	U	LU	U				

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	4	4	4	4
6	6	5	4	4	4	5	4
7	7	6	5	5			

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	LU
7	U	LU	U	LU			

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	4	4	4	4
6	6	5	4	4	4	5	4
7	7	6	5	5	5		

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	LU
7	U	LU	U	LU	LU		

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	4	4	4	4
6	6	5	4	4	4	5	4
7	7	6	5	5	5	4	

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	LU
7	U	LU	U	LU	LU	LU	

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	4	4	4	4
6	6	5	4	4	4	5	4
7	7	6	5	5	5	4	5

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	LU
7	U	LU	U	LU	LU	LU	L

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*D*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	1	2	3	3	4	5
2	2	1	2	3	4	3	4
3	3	2	2	2	3	4	4
4	4	3	3	3	3	3	4
5	5	4	3	最优解			4
6	6	5	4				4
7	7	6	5	5	5	4	5

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	LU
7	U	LU	U	LU	LU	LU	L

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	LU
7	U	LU	U	LU	LU	LU	L

操作：

插入A

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	LU
7	U	LU	U	LU	LU	LU	L

操作：

无需操作  
插入A



# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	LU
7	U	LU	U	LU	LU	LU	L

操作：

无需操作  
无需操作  
插入A

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	LU
7	U	LU	U	LU	LU	LU	L

操作：

用C替换D  
无需操作  
无需操作  
插入A

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

操作：

用D替换B  
用C替换D  
无需操作  
无需操作  
插入A

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	LU
7	U	LU	U	LU	LU	LU	L

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

操作：

删除C  
用D替换B  
用C替换D  
无需操作  
无需操作  
插入A

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	LU
7	U	LU	U	LU	LU	LU	L

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	<b>B</b>	C	B	D	A	B
<i>t</i>	<b>B</b>	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

操作：

无需操作

删除C

用D替换B

用C替换D

无需操作

无需操作

插入A

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	<b>LU</b>	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	LU
7	U	LU	U	LU	LU	LU	L

# 算法实例



	1	2	3	4	5	6	7
<i>s</i>	A	B	C	B	D	A	B
<i>t</i>	B	D	C	A	B	A	

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$

操作：  
 删除A  
 无需操作  
 删除C  
 用D替换B  
 用C替换D  
 无需操作  
 无需操作  
 插入A

*Rec*

<i>i</i> \ <i>j</i>	0	1	2	3	4	5	6
0		L	L	L	L	L	L
1	U	LU	LU	LU	LU	L	LU
2	U	LU	LU	LU	LU	LU	L
3	U	U	LU	LU	L	L	LU
4	U	LU	LU	LU	LU	LU	L
5	U	U	LU	LU	LU	LU	LU
6	U	U	U	LU	LU	LU	LU
7	U	LU	U	LU	LU	LU	L



# 动态规划算法：伪代码

- Minimum-Edit-Distance( $s, t$ )

输入: 字符串 $s$ 和 $t$

输出:  $s$ 和 $t$ 的最小编辑距离

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

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

新建 $D[0..n, 0..m]$ ,  $Rec[0..n, 0..m]$ 两个二维数组

//初始化

for  $i \leftarrow 0$  to  $n$  do

$D[i, 0] \leftarrow i$

$Rec[i, 0] \leftarrow "U"$

end

for  $j \leftarrow 0$  to  $m$  do

$D[0, j] \leftarrow j$

$Rec[0, j] \leftarrow "L"$

end

初始化



# 动态规划算法：伪代码

## ● Minimum-Edit-Distance( $s, t$ )

//动态规划

for  $i \leftarrow 1$  to  $n$  do

  for  $j \leftarrow 1$  to  $m$  do

$c \leftarrow 0$

    if  $s_i \neq t_j$  then

$c \leftarrow 1$

    end

$replace \leftarrow D[i-1, j-1] + c$

$delete \leftarrow D[i-1, j] + 1$

$insert \leftarrow D[i, j-1] + 1$

    if  $replace = \min\{delete, insert, replace\}$  then

$D[i, j] \leftarrow D[i-1, j-1] + c$

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

    end

    else if  $insert = \min\{delete, insert, replace\}$  then

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

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

    end

    else

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

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

    end

  end

end

依次计算子问题





# 动态规划算法：伪代码

## ● Minimum-Edit-Distance( $s, t$ )

```
//动态规划
for  $i \leftarrow 1$  to  $n$  do
  for  $j \leftarrow 1$  to  $m$  do
     $c \leftarrow 0$ 
    if  $s_i \neq t_j$  then
       $c \leftarrow 1$ 
    end
     $replace \leftarrow D[i-1, j-1] + c$ 
     $delete \leftarrow D[i-1, j] + 1$ 
     $insert \leftarrow D[i, j-1] + 1$ 
    if  $replace = \min\{delete, insert, replace\}$  then
       $D[i, j] \leftarrow D[i-1, j-1] + c$ 
       $Rec[i, j] \leftarrow "LU"$ 
    end
    else if  $insert = \min\{delete, insert, replace\}$  then
       $D[i, j] \leftarrow D[i, j-1] + 1$ 
       $Rec[i, j] \leftarrow "L"$ 
    end
    else
       $D[i, j] \leftarrow D[i-1, j] + 1$ 
       $Rec[i, j] \leftarrow "U"$ 
    end
  end
end
end
```

替换/无需操作



# 动态规划算法：伪代码

## ● Minimum-Edit-Distance( $s, t$ )

```
//动态规划
for  $i \leftarrow 1$  to  $n$  do
  for  $j \leftarrow 1$  to  $m$  do
     $c \leftarrow 0$ 
    if  $s_i \neq t_j$  then
       $c \leftarrow 1$ 
    end
     $replace \leftarrow D[i-1, j-1] + c$ 
     $delete \leftarrow D[i-1, j] + 1$ 
     $insert \leftarrow D[i, j-1] + 1$ 
    if  $replace = \min\{delete, insert, replace\}$  then
       $D[i, j] \leftarrow D[i-1, j-1] + c$ 
       $Rec[i, j] \leftarrow "LU"$ 
    end
    else if  $insert = \min\{delete, insert, replace\}$  then
       $D[i, j] \leftarrow D[i, j-1] + 1$ 
       $Rec[i, j] \leftarrow "L"$ 
    end
    else
       $D[i, j] \leftarrow D[i-1, j] + 1$ 
       $Rec[i, j] \leftarrow "U"$ 
    end
  end
end
end
```

采用替换操作



# 动态规划算法：伪代码

## ● Minimum-Edit-Distance( $s, t$ )

```
//动态规划
for  $i \leftarrow 1$  to  $n$  do
  for  $j \leftarrow 1$  to  $m$  do
     $c \leftarrow 0$ 
    if  $s_i \neq t_j$  then
       $c \leftarrow 1$ 
    end
     $replace \leftarrow D[i-1, j-1] + c$ 
     $delete \leftarrow D[i-1, j] + 1$ 
     $insert \leftarrow D[i, j-1] + 1$ 
    if  $replace = \min\{delete, insert, replace\}$  then
       $D[i, j] \leftarrow D[i-1, j-1] + c$ 
       $Rec[i, j] \leftarrow "LU"$ 
    end
    else if  $insert = \min\{delete, insert, replace\}$  then
       $D[i, j] \leftarrow D[i, j-1] + 1$ 
       $Rec[i, j] \leftarrow "L"$ 
    end
    else
       $D[i, j] \leftarrow D[i-1, j] + 1$ 
       $Rec[i, j] \leftarrow "U"$ 
    end
  end
end
end
```

记录编辑距离和操作



# 动态规划算法：伪代码

## ● Minimum-Edit-Distance( $s, t$ )

```
//动态规划
for  $i \leftarrow 1$  to  $n$  do
  for  $j \leftarrow 1$  to  $m$  do
     $c \leftarrow 0$ 
    if  $s_i \neq t_j$  then
       $c \leftarrow 1$ 
    end
     $replace \leftarrow D[i-1, j-1] + c$ 
     $delete \leftarrow D[i-1, j] + 1$ 
     $insert \leftarrow D[i, j-1] + 1$ 
    if  $replace = \min\{delete, insert, replace\}$  then
       $D[i, j] \leftarrow D[i-1, j-1] + c$ 
       $Rec[i, j] \leftarrow "LU"$ 
    end
    else if  $insert = \min\{delete, insert, replace\}$  then
       $D[i, j] \leftarrow D[i, j-1] + 1$ 
       $Rec[i, j] \leftarrow "L"$ 
    end
    else
       $D[i, j] \leftarrow D[i-1, j] + 1$ 
       $Rec[i, j] \leftarrow "U"$ 
    end
  end
end
end
```

采取插入操作



# 动态规划算法：伪代码

## ● Minimum-Edit-Distance( $s, t$ )

```
//动态规划
for  $i \leftarrow 1$  to  $n$  do
  for  $j \leftarrow 1$  to  $m$  do
     $c \leftarrow 0$ 
    if  $s_i \neq t_j$  then
       $c \leftarrow 1$ 
    end
     $replace \leftarrow D[i-1, j-1] + c$ 
     $delete \leftarrow D[i-1, j] + 1$ 
     $insert \leftarrow D[i, j-1] + 1$ 
    if  $replace = \min\{delete, insert, replace\}$  then
       $D[i, j] \leftarrow D[i-1, j-1] + c$ 
       $Rec[i, j] \leftarrow "LU"$ 
    end
    else if  $insert = \min\{delete, insert, replace\}$  then
       $D[i, j] \leftarrow D[i, j-1] + 1$ 
       $Rec[i, j] \leftarrow "L"$ 
    end
    else
       $D[i, j] \leftarrow D[i-1, j] + 1$ 
       $Rec[i, j] \leftarrow "U"$ 
    end
  end
end
end
```

采取删除操作



# 最优方案追踪：伪代码

## ● Print-MED( $Rec, s, t, i, j$ )

输入: 矩阵 $Rec$ , 字符串 $s, t$ , 位置索引 $i$ 和 $j$

输出: 操作序列

if  $i = 0$  and  $j = 0$  then

    return  $NULL$

end

if  $Rec[i, j] = "LU"$  then

    Print-MED( $Rec, s, t, i - 1, j - 1$ )

    if  $s_i = t_j$  then

        print “无需操作”

    end

    else

        print “用 $t_j$ 替换 $s_i$ ”

    end

end

else if  $Rec[i, j] = "U"$  then

    Print-MED( $Rec, s, t, i - 1, j$ )

    print “删除 $s_i$ ”

end

else

    Print-MED( $Rec, s, t, i, j - 1$ )

    print “插入 $t_j$ ”

end

采取替换操作



# 最优方案追踪：伪代码

## • Print-MED( $Rec, s, t, i, j$ )

输入: 矩阵 $Rec$ , 字符串 $s, t$ , 位置索引 $i$ 和 $j$

输出: 操作序列

if  $i = 0$  and  $j = 0$  then

    | return  $NULL$

end

if  $Rec[i, j] = "LU"$  then

    | Print-MED( $Rec, s, t, i - 1, j - 1$ )

    | if  $s_i = t_j$  then

        | print “无需操作”

    end

    else

        | print “用 $t_j$ 替换 $s_i$ ”

    end

end

else if  $Rec[i, j] = "U"$  then

    | Print-MED( $Rec, s, t, i - 1, j$ )

    | print “删除 $s_i$ ”

end

else

    | Print-MED( $Rec, s, t, i, j - 1$ )

    | print “插入 $t_j$ ”

end

递归输出子问题方案



# 最优方案追踪：伪代码

## • Print-MED( $Rec, s, t, i, j$ )

输入: 矩阵 $Rec$ , 字符串 $s, t$ , 位置索引 $i$ 和 $j$

输出: 操作序列

if  $i = 0$  and  $j = 0$  then

    | return  $NULL$

end

if  $Rec[i, j] = "LU"$  then

    Print-MED( $Rec, s, t, i - 1, j - 1$ )

    if  $s_i = t_j$  then

        | print “无需操作”

    end

    else

        | print “用 $t_j$ 替换 $s_i$ ”

    end

end

else if  $Rec[i, j] = "U"$  then

    Print-MED( $Rec, s, t, i - 1, j$ )

    print “删除 $s_i$ ”

end

else

    Print-MED( $Rec, s, t, i, j - 1$ )

    print “插入 $t_j$ ”

end

替换/无操作





# 最优方案追踪：伪代码

## • Print-MED( $Rec, s, t, i, j$ )

输入: 矩阵 $Rec$ , 字符串 $s, t$ , 位置索引 $i$ 和 $j$

输出: 操作序列

if  $i = 0$  and  $j = 0$  then

    | return  $NULL$

end

if  $Rec[i, j] = "LU"$  then

    Print-MED( $Rec, s, t, i - 1, j - 1$ )

    if  $s_i = t_j$  then

        | print “无需操作”

    end

    else

        | print “用 $t_j$ 替换 $s_i$ ”

    end

end

( else if  $Rec[i, j] = "U"$  then

    Print-MED( $Rec, s, t, i - 1, j$ )

    print “删除 $s_i$ ”

end

else

    Print-MED( $Rec, s, t, i, j - 1$ )

    print “插入 $t_j$ ”

end

采取删除操作



# 最优方案追踪：伪代码

## ● Print-MED( $Rec, s, t, i, j$ )

输入: 矩阵 $Rec$ , 字符串 $s, t$ , 位置索引 $i$ 和 $j$

输出: 操作序列

if  $i = 0$  and  $j = 0$  then

    | return  $NULL$

end

if  $Rec[i, j] = "LU"$  then

    Print-MED( $Rec, s, t, i - 1, j - 1$ )

    if  $s_i = t_j$  then

        | print “无需操作”

    end

    else

        | print “用 $t_j$ 替换 $s_i$ ”

    end

end

else if  $Rec[i, j] = "U"$  then

    Print-MED( $Rec, s, t, i - 1, j$ )

    print “删除 $s_i$ ”

end

else

    Print-MED( $Rec, s, t, i, j - 1$ )

    print “插入 $t_j$ ”

end

采取插入操作



# 时间复杂度分析

//动态规划

```
for i ← 1 to n do
  for j ← 1 to m do
    c ← 0
    if  $s_i \neq t_j$  then
      | c ← 1
    end
    replace ←  $D[i - 1, j - 1] + c$ 
    delete ←  $D[i - 1, j] + 1$ 
    insert ←  $D[i, j - 1] + 1$ 
    if replace = min{delete, insert, replace} then
      |  $D[i, j] \leftarrow D[i - 1, j - 1] + c$ 
      | Rec[i, j] ← “LU”
    end
    else if insert = min{delete, insert, replace} then
      |  $D[i, j] \leftarrow D[i, j - 1] + 1$ 
      | Rec[i, j] ← “L”
    end
    else
      |  $D[i, j] \leftarrow D[i - 1, j] + 1$ 
      | Rec[i, j] ← “U”
    end
  end
end
end
```

时间复杂度： $O(mn)$

$O(m)$   $O(mn)$

# 小结



## 最长公共子序列

如果  $s_i \neq t_j$

<b>s</b>	A	B	C	B	D	A	<b>B</b>
----------	---	---	---	---	---	---	----------

<b>t</b>	B	D	C	A	B	<b>A</b>
----------	---	---	---	---	---	----------

如果  $s_i = t_j$

<b>s</b>	A	B	C	B	D	A	<b>B</b>
----------	---	---	---	---	---	---	----------

<b>t</b>	B	D	C	A	<b>B</b>
----------	---	---	---	---	----------

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

## 最小编辑距离

<b>s</b>	A	B	C	B	D	A	<b>B</b>
----------	---	---	---	---	---	---	----------

删除

<b>t</b>	B	D	C	A	B	A
----------	---	---	---	---	---	---

<b>s</b>	A	B	C	B	D	A	B	<b>?</b>
----------	---	---	---	---	---	---	---	----------

插入

<b>t</b>	B	D	C	A	B	A
----------	---	---	---	---	---	---

<b>s</b>	A	B	C	B	D	A	<b>B</b>	<b>?</b>
----------	---	---	---	---	---	---	----------	----------

替换

<b>t</b>	B	D	C	A	B	A
----------	---	---	---	---	---	---

$$D[i, j] = \min \begin{cases} D[i-1, j] + 1 \\ D[i, j-1] + 1 \\ D[i-1, j-1] + \begin{cases} 0, & \text{if } s[i] = t[j] \\ 1, & \text{if } s[i] \neq t[j] \end{cases} \end{cases}$$



# 谢谢

