

# **CMPSC 465**

## **Data Structures and Algorithms**

### **Spring 2022**

---

Instructor: Chunhao Wang

# Dynamic Programming

---

# Dynamic Programming

---

## Prelude

# Key steps to design DP algorithms

1. Identify subproblems
2. Recurrence  
e.g.  $L(j) = 1 + \max\{L(i) : a_i < a_j\}$
3. Base case

# Dynamic Programming

---

## Edit Distance (Textbook Section 6.3)

# Edit distance

Motivation: consider DNA sequences  $x = ACGTA$ ,  $y = ATCTG$ .




Note  $|x| \neq |y|$  in general

Question: how far away are  $x$  and  $y$ ?






## Definition

The **edit distance** between  $x$  and  $y$ , denoted by  $d(x, y)$ , is the minimum number of insertions, deletions, and substitutions needed to transform  $x$  to  $y$

Consider the following **alignments**:

x:	A	-	C	G	T	A
						
y:	A	T	C	-	T	G

cost : 3 (optimal)

x:	A	C	-	-	G	T	A
							
y:	A	T	C	T	G	-	-

cost : 5

So  $d(x, y) = 3$

## Edit distance — subproblem

Consider two strings

$$x = x_1x_2 \cdots x_m \quad \text{and} \quad y = y_1y_2 \cdots y_n$$

Subproblem: consider prefix  $x_1 \cdots x_i$  and  $y_1 \cdots y_j$  ( $i \leq m, j \leq n$ )

Define

$$E(i, j) = d(x_1 \cdots x_i, y_1 \cdots y_j)$$

Optimal solution:  $E(m, n)$

How to use the solution to the subproblems to solve  $E(i, j)$ ?

# Recurrence (I)

Look at the rightmost column:

$$\text{Case 1} \quad \begin{array}{cccc} x_1 & \cdots & x_{i-1} & x_i \\ y_1 & \cdots & y_j & - \end{array}$$

Contributes 1 to the cost plus the cost of alignment  $\begin{array}{ccc} x_1 & \cdots & x_{i-1} \\ y_1 & \cdots & y_j \end{array}$

$$E(i, j) = 1 + E(i - 1, j)$$

$$\text{Case 2} \quad \begin{array}{cccc} x_1 & \cdots & x_i & - \\ y_1 & \cdots & y_{j-1} & y_j \end{array}$$

Contributes 1 to the cost plus the cost of alignment  $\begin{array}{ccc} x_1 & \cdots & x_i \\ y_1 & \cdots & y_{j-1} \end{array}$

$$E(i, j) = 1 + E(i, j - 1)$$

$$\text{Case 3} \quad \begin{array}{cccc} x_1 & \cdots & x_{i-1} & x_i \\ y_1 & \cdots & y_{j-1} & y_j \end{array}$$

$$E(i, j) = \begin{cases} E(i - 1, j - 1) & \text{if } x_i = y_j \\ 1 + E(i - 1, j - 1) & \text{otherwise} \end{cases}$$



## Recurrence (II)

**The recurrence:**

$$E(i, j) = \min\{1 + E(i - 1, j), 1 + E(i, j - 1), \text{diff}(i, j) + E(i - 1, j - 1)\},$$

where

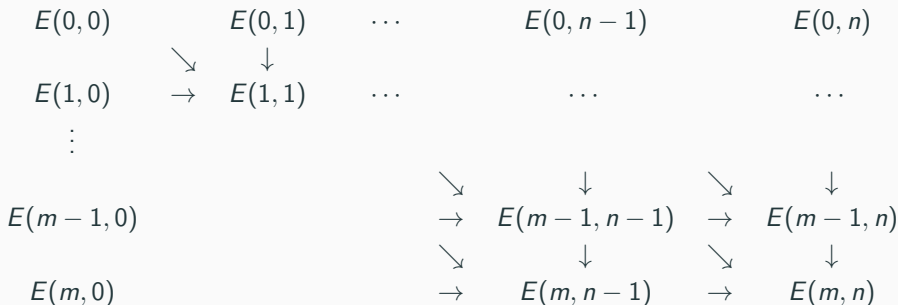
$$\text{diff}(i, j) = \begin{cases} 1 & \text{if } x_i \neq y_j \\ 0 & \text{otherwise} \end{cases}$$

**Optimal solution:**  $E(m, n)$

**Base case:**  $E(0, 0) = 0$ ,  $E(i, 0) = i$ ,  $E(0, j) = j$

# Filling the table

$$E(i, j) = \min\{1 + E(i - 1, j), 1 + E(i, j - 1), \text{diff}(i, j) + E(i - 1, j - 1)\},$$



# Running example

$x = \text{ACGTA}$  and  $y = \text{ATCTG}$

		A	T	C	T	G
	0	1	2	3	4	5
A	1	0	1	2	3	4
C	2	1	1	1	2	3
G	3	2	2	2	2	2
T	4	3	2	3	2	3
A	5	4	3	3	3	3