Key steps to design DP algorithms

1. Identify subproblems

Key steps to design DP algorithms

1. Identify subproblems
2. Recurrence

   e.g. $L(j) = 1 + \max\{L(i) : a_i < a_j\}$

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 $x = ACGTA$, $y = ATCTG$.

Edit distance

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

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

Edit distance

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

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

Question: how far away are x and y?

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

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**:

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

3

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

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

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

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

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

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

cost : 5

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**:

$$\begin{array}{ccccccc} \text{x:} & A & - & C & G & T & A \\ & & \updownarrow & & \updownarrow & & \updownarrow \\ \text{y:} & A & T & C & - & T & G \end{array}$$

cost : 3 (optimal)

$$\begin{array}{cccccccc} \text{x:} & A & C & - & - & G & T & A \\ & & \updownarrow & \updownarrow & \updownarrow & & \updownarrow & \updownarrow \\ \text{y:} & A & T & C & T & G & - & - \end{array}$$

cost : 5

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

$$x = x_1 x_2 \cdots x_m \quad \text{and} \quad y = y_1 y_2 \cdots y_n$$

Edit distance — subproblem

Consider two strings

$$x = x_1 x_2 \cdots x_m \quad \text{and} \quad y = y_1 y_2 \cdots y_n$$

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

Edit distance — subproblem

Consider two strings

$$x = x_1 x_2 \cdots x_m \quad \text{and} \quad y = y_1 y_2 \cdots y_n$$

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

Define

$$\boxed{E(i,j)} = d(x_1 \cdots x_i, y_1 \cdots y_j) \qquad \text{Goal: } \boxed{d(x, y)}$$

$$\boxed{E(m,n)} = d(\underbrace{x_1 \cdots x_m}_{x}, \underbrace{y_1 \cdots y_n}_{y}) = \boxed{d(x, y)}$$

Edit distance — subproblem

Consider two strings

$$x = x_1 x_2 \cdots x_m \quad \text{and} \quad y = y_1 y_2 \cdots y_n$$

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: $E(m, n)$

Edit distance — subproblem

Consider two strings

$$x = x_1 x_2 \cdots x_m \quad \text{and} \quad y = y_1 y_2 \cdots y_n$$

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

Define

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

$E(i,0)$

$= d(x_1 \cdots x_i, -)$

$= i$

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:

Recurrence (I)

Look at the rightmost column:

Case 1 $\boxed{\begin{matrix} x_1 & \cdots & x_{i-1} \\ y_1 & \cdots & y_j \end{matrix}}$ $\begin{matrix} x_i \\ - \end{matrix}$

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

Recurrence (I)

Look at the rightmost column:

Case 1 $\begin{matrix} x_1 & \cdots & x_{i-1} & x_i \\ y_1 & \cdots & y_j & - \end{matrix}$

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

Recurrence (I)

Look at the rightmost column:

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

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

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

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

Recurrence (I)

Look at the rightmost column:

Case 1 $\begin{matrix} x_1 & \cdots & x_{i-1} & \color{red}{x_i} \\ y_1 & \cdots & y_j & \color{red}{\text{-}} \end{matrix}$

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

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

Case 2 $\begin{matrix} x_1 & \cdots & x_i & \color{red}{\text{-}} \\ y_1 & \cdots & y_{j-1} & \color{red}{y_j} \end{matrix}$

Recurrence (I)

Look at the rightmost column:

Case 1 $\begin{matrix} x_1 & \cdots & x_{i-1} & x_i \\ y_1 & \cdots & y_j & - \end{matrix}$

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

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

Case 2 $\begin{matrix} x_1 & \cdots & x_i & - \\ y_1 & \cdots & y_{j-1} & y_j \end{matrix}$

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

Recurrence (I)

Look at the rightmost column:

Case 1 $\begin{matrix} x_1 & \cdots & x_{i-1} & x_i \\ y_1 & \cdots & y_j & - \end{matrix}$

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

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

Case 2 $\begin{matrix} x_1 & \cdots & x_i & - \\ y_1 & \cdots & y_{j-1} & y_j \end{matrix}$

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

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

Recurrence (I)

Look at the rightmost column:

Case 1 $\begin{matrix} x_1 & \cdots & x_{i-1} & x_i \\ y_1 & \cdots & y_j & - \end{matrix}$

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

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

Case 2 $\begin{matrix} x_1 & \cdots & x_i & - \\ y_1 & \cdots & y_{j-1} & y_j \end{matrix}$

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

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

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

if $x_i = y_j$ no extra cost

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

if $x_i \neq y_j$ +1

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

Recurrence (I)

Look at the rightmost column:

$E(i,j) = ?$

Case 1 $\begin{matrix} x_1 & \cdots & x_{i-1} & x_i \\ y_1 & \cdots & y_j & - \end{matrix}$

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

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

Case 2 $\begin{matrix} x_1 & \cdots & x_i & - \\ y_1 & \cdots & y_{j-1} & y_j \end{matrix}$

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

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

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

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

Recurrence (II)

The recurrence:

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

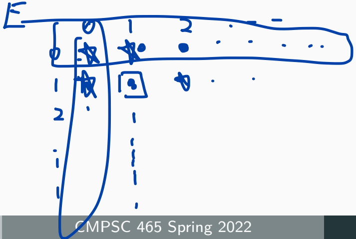
where

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

Base case :

$E(i,0) = i$

$E(0,j) = j$



Recurrence (II)

The recurrence:

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

where

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

Optimal solution: $E(m,n)$

Recurrence (II)

The recurrence:

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

where

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

Optimal solution: $E(m,n)$

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

Filling the table

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

Filling the table

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

$$\begin{array}{ccccccccc}
E(0,0) & & E(0,1) & \cdots & & E(0,n-1) & & E(0,n) \\
& \searrow & \downarrow & & & & & \\
E(1,0) & \rightarrow & E(1,1) & \cdots & & \cdots & & \cdots \\
\vdots & & & & & & & \\
& & & & \searrow & \downarrow & \searrow & \downarrow \\
E(m-1,0) & & & & \rightarrow & E(m-1,n-1) & \rightarrow & E(m-1,n) \\
& & & & \searrow & \downarrow & \searrow & \downarrow \\
E(m,0) & & & & \rightarrow & E(m,n-1) & \rightarrow & E(m,n)
\end{array}$$

Running example

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

Running example

$x = ACGTA$ and $y = ATCTG$

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

Base case

Base case