

LA PIU' LUNGA SOTTOSEQUENZA COMUNE (LONGEST COMMON SUBSEQUENCE : LCS)

• APPLICAZIONI BIOLOGICHE

- CONFRONTO **DNA** DI DIVERSI ORGANISMI
- STRUTTURA **DNA** : FORMATA DA UNA STRINGA DI MOLECOLE
DETTE **BASI**
 - **BASI** :
 - **A**DENINA (A)
 - **C**ITOSINA (C)
 - **G**UANINA (G)
 - **T**IMINA (T)
- QUINDI IL **DNA** DI UN ORGANISMO PUO' ESSERE
RAPPRESENTATO COME UNA STRINGA NELL'ALFABETO $\{A, C, G, T\}$
- LA **CORRELAZIONE** TRA DUE ORGANISMI PUO' ESSERE
"MISURATA" CON IL GRADO DI **SOMIGLIANZA** DEI LORO **DNA**

DEF UNA SOTTOSEQUENZA DI UNA SEQUENZA DATA X È
UNA SEQUENZA OTTENUTA CANCELLANDO DA X ZERO
O PIÙ ELEMENTI (MANTENENDO L'ORDINE).
PIÙ PRECISAMENTE, SE $X = \langle x_1, x_2, \dots, x_m \rangle$, UNA
SEQUENZA $Z = \langle z_1, z_2, \dots, z_k \rangle$ È UNA SOTTOSEQUENZA
DI X SE ESISTE UNA SEQUENZA $\langle i_1, i_2, \dots, i_k \rangle$
DI INDICI TALE CHE:

- $1 \leq i_1 < i_2 < \dots < i_k \leq m$
- $z_j = x_{i_j}$, PER OGNI $j = 1, 2, \dots, k$

ESEMPIO SIA $X = \langle A, B, C, B, D, A, B \rangle$.

ALLORA $Z = \langle B, C, D, B \rangle$ È UNA SOTTOSEQUENZA DI X
CORRISPONDENTE ALLA SEQUENZA DI INDICI $\langle 2, 3, 5, 7 \rangle$

DEF DATE DUE SEQUENZE X E Y , DICIAMO CHE Z È
UNA SOTTOSEQUENZA COMUNE DI X E Y SE Z
È UNA SOTTOSEQUENZA DI ENTRAMBE LE SEQUENZE X E Y

ESEMPIO DATE $X = \langle A, B, C, B, D, A, B \rangle$ E
 $Y = \langle B, D, C, A, B, A \rangle$,
• LA SEQUENZA $Z = \langle B, C, A \rangle$ È UNA SOTTOSEQUENZA
COMUNE DI X E Y .
• TUTTAVIA $\langle B, C, A \rangle$ NON È LA PIÙ LUNGA
SOTTOSEQUENZA COMUNE DI X E Y , IN QUANTO
 $\langle B, C, B, A \rangle$, $\langle B, C, A, B \rangle$, $\langle B, D, A, B \rangle$ SONO
SOTTOSEQUENZE COMUNI DI X E Y (DI LUNGHEZZA
MASSIMA).

PROBLEMA DELLA PIÙ LUNGA SOTTOSEQUENZA COMUNE:

DATE DUE SEQUENZE X E Y DETERMINARE UNA SOTTOSEQUENZA DI LUNGHEZZA MASSIMA (LCS) CHE SIA COMUNE A X E Y .

SOLUZIONE MEDIANTE RICERCA ESAUSTIVA

E' ESPOENZIALE IN $\min(|X|, |Y|)$, IN QUANTO UNA SEQUENZA DI LUNGHEZZA m HA ESATTAMENTE 2^m SOTTOSEQUENZE.

- IL PROBLEMA DELLA LCS PUÒ ESSERE RISOLTO IN MODO EFFICIENTE UTILIZZANDO LA PROGRAMMAZIONE DINAMICA.

FASE 1: CARATTERIZZAZIONE DELLA PIÙ LUNGA SOTTOSEQUENZA COMUNE

NOTAZIONE DATA UNA SEQUENZA $X = \langle x_1, x_2, \dots, x_m \rangle$
DEFINIAMO $X_i = \langle x_1, x_2, \dots, x_i \rangle$, PER $i = 0, 1, 2, \dots, m$.

INOLTRE PER OGNI SIMBOLO a DELL'ALFABETO
PONIAMO $Xa = \langle x_1, x_2, \dots, x_m, a \rangle$

ESEMPIO: SE $X = \langle A, B, C, B, D, A, B \rangle$, ALLORA

$X_1 = \langle A \rangle$, $X_4 = \langle A, B, C, B \rangle$, $X_0 = \langle \rangle$ (SEQUENZA VUOTA)

$X_3 D = \langle A, B, C, D \rangle$, ECC.

TEOREMA (SOTTOSTRUTTURA OTTIMA DI UNA LCS)

SIANO DATE DUE SEQUENZE $X = \langle x_1, x_2, \dots, x_m \rangle$ E
 $Y = \langle y_1, y_2, \dots, y_n \rangle$, TALI CHE $m \geq 1$ E $n \geq 1$,
E SIA $Z = \langle z_1, z_2, \dots, z_k \rangle$ UNA LCS DI X E Y .

1. SE $x_m = y_m$, ALLORA

- $z_k = x_m = y_m$ E

- Z_{k-1} E' UNA LCS DI X_{m-1} E Y_{m-1}

2. SE $x_m \neq y_m$, ALLORA

$z_k \neq x_m \implies Z$ E' UNA LCS DI X_{m-1} E Y

3. SE $x_m \neq y_m$, ALLORA

$z_k \neq y_m \implies Z$ E' UNA LCS DI X E Y_{n-1} .

DIM. (1) SE $z_k \neq x_m$ ALLORA Z SAREBBE UNA
SOTTOSEQUENZA COMUNE DI X_{m-1} E Y_{m-1}
E QUINDI Z_{x_m} SAREBBE UNA SOTTOSEQUENZA
COMUNE DI X E Y , ASSURDO.

QUINDI $z_k = x_m = y_m$ E PERTANTO Z_{k-1} E'
UNA SOTTOSEQUENZA COMUNE DI X_{m-1} E Y_{m-1} .
SE Z_{k-1} NON FOSSE DI LUNGEZZA MASSIMA,
ESISTEREBBE W LCS DI X_{m-1} E Y_{m-1} TALE
CHE $|W| > |Z_{k-1}|$.

MA, ALLORA W_{x_m} SAREBBE UNA SOTTOSEQUENZA
COMUNE DI X E Y , ASSURDO IN QUANTO
 $|W_{x_m}| > |Z|$ E Z E' UNA LCS DI
 X E Y

(2) SE $z_k \neq x_m$ ALLORA Z E' UNA SOTTOSEQUENZA
COMUNE DI X_{m-1} E Y .

SE ESISTESSE UNA SOTTOSEQUENZA COMUNE W
DI X_{m-1} E Y TALE CHE $|W| > |Z|$, CIO'
SAREBBE ASSURDO IN QUANTO W SAREBBE A
MAGGIOR RAGIONE UNA SOTTOSEQUENZA
COMUNE DI X E Y (DI LUNGHEZZA MAGGIORE
DI QUELLA DELLA LCS Z).

(3) ANALOGO AL CASO (2)

SPAZIO DEI SOTTOPROBLEMI

DAL PRECEDENTE TEOREMA SEGUE CHE LO SPAZIO DEI SOTTOPROBLEMI È $\{(X_i, Y_j) : 0 \leq i \leq m, 0 \leq j \leq m\}$ LA CUI CARDINALITÀ È $O(m \cdot m)$.

FASE 2: SOLUZIONE RICORSIVA

DEFINIAMO $c[i, j]$, PER $0 \leq i \leq m$ E $0 \leq j \leq m$, COME LA LUNGHEZZA DI UNA LCS DI X_i E Y_j .
IN VIRTU' DELLA SOTTOSTRUTTURA OTTIMA SI HA

$$c[i, j] = \begin{cases} 0 & \text{SE } i=0 \text{ O } j=0 \\ c[i-1, j-1] + 1 & \text{SE } i, j > 0 \text{ E } x_i = y_j \\ \max(c[i, j-1], c[i-1, j]) & \text{SE } i, j > 0 \text{ E } x_i \neq y_j \end{cases}$$

- NUMERO n_1 DI SOTTOPROBLEMI UTILIZZATI IN UNA SOLUZIONE OTTIMA $= 1$
- NUMERO n_2 DI SCELTE PER DETERMINARE QUALI SOTTOPROBLEMI UTILIZZARE ≤ 2

FASE 3: CALCOLO DELLA LUNGHEZZA DI UNA LCS

LCS_LENGTH (X, Y)

$m := \text{length}[X]$

$n := \text{length}[Y]$

for $i := 1$ to m do

$c[i, 0] := 0$

for $j := 0$ to n do

$c[0, j] := 0$

CASO BASE

for $i := 1$ to m do

for $j := 1$ to n do

if $x_i = y_j$ then

$c[i, j] := c[i-1, j-1] + 1$; $b[i, j] := "\nearrow"$

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

$c[i, j] := c[i-1, j]$; $b[i, j] := "\uparrow"$

else $c[i, j] := c[i, j-1]$; $b[i, j] := "\leftarrow"$

return c, b

DEFINIZIONE
RICORSIVA

COMPLESSITA' DI LCS_LENGTH: $\Theta(mn)$

ESEMPIO: DATI $X = \langle A, B, C, B, D, A, B \rangle$
 $Y = \langle B, D, C, A, B, A \rangle$

LCS_LENGTH (X, Y)

		0	1	2	3	4	5	6
	y_j		B	D	C	A	B	A
0	x_i	0	0	0	0	0	0	0
1	A	0	↑0	↑0	↑0	↖1	↖1	↖1
2	B	0	↖1	↖1	↖1	↑1	↖2	↖2
3	C	0	↑1	↑1	↖2	↖2	2	2
4	B	0	↖1	↑1	↑2	↑2	↖3	↖3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

CALCOLA LA TABELLA.

DA QUESTA SI
EVINCE SUBITO CHE
LA LUNGHEZZA DI UNA
LCS E' 4

FASE 4: COSTRUZIONE DI UNA LCS

PRINT-LCS (b, X, i, j)

if $i=0$ o $j=0$ then
return

if $b[i, j] = "↖"$ then
PRINT-LCS (b, X, i-1, j-1)

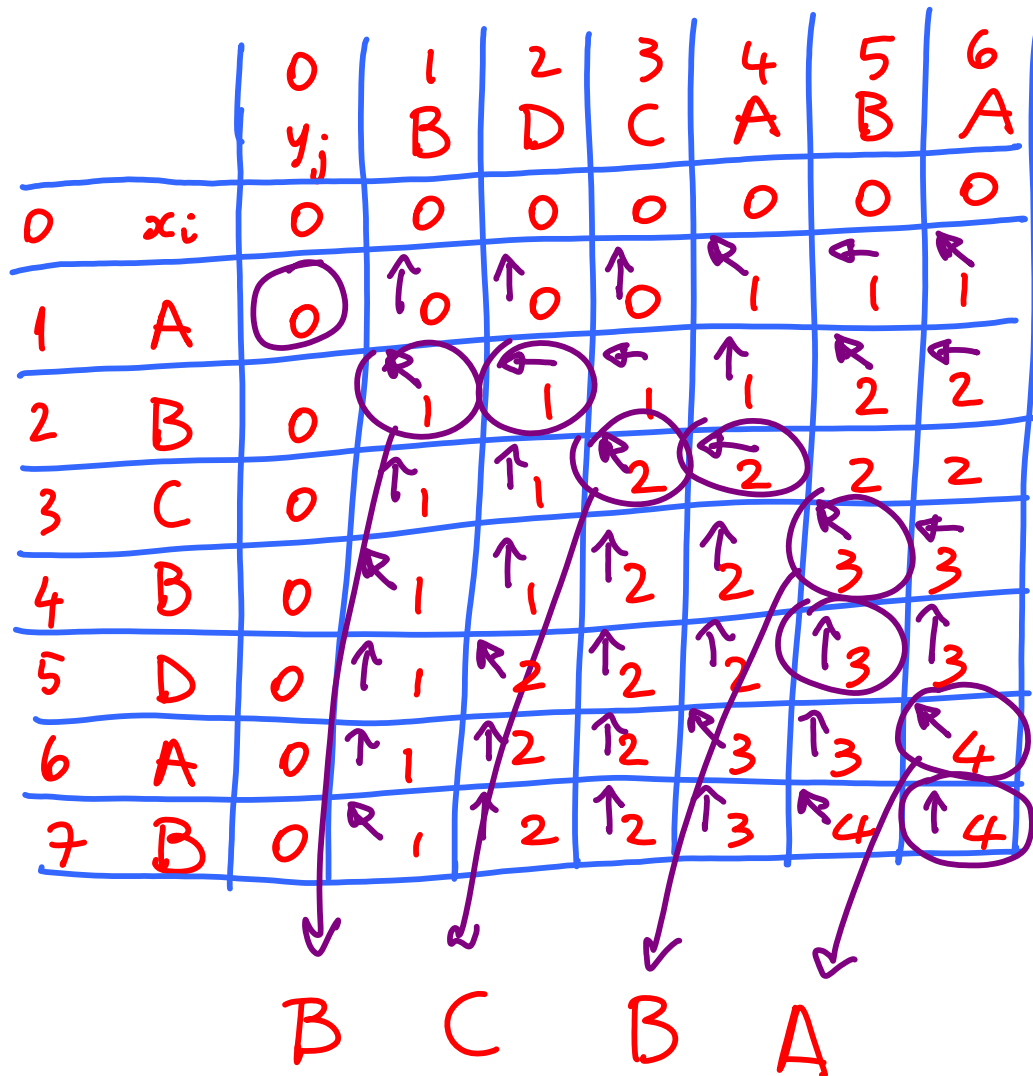
stampa x_i

elseif $b[i, j] = "↑"$ then
PRINT-LCS (b, X, i-1, j)

else
PRINT-LCS (b, X, i, j-1)

COMPLESSITA': $O(m+n)$

PRINT-LCS (b, X, m, m)



E' UNA LCS

ESERCIZI

15.4-4 SPIEGARE COME CALCOLARE LA LUNGHEZZA DI UNA LCS UTILIZZANDO SOLTANTO $2 \min(m, n)$ POSIZIONI NELLA TABELLA c PIÙ UNO SPAZIO $O(1)$ AGGIUNTIVO, RISOLVERE LO STESSO PROBLEMA UTILIZZANDO $\min(m, n)$ POSIZIONI PIÙ UNO SPAZIO $O(1)$ AGGIUNTIVO,

15.4-5 PROGETTARE UN ALGORITMO $O(n^2)$ PER TROVARE UNA PIÙ LUNGA SOTTOSEQUENZA CRESCENTE DI UNA SEQUENZA DI n NUMERI