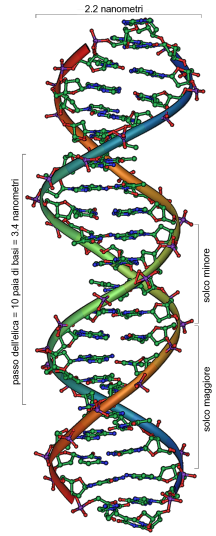
LA PIU' LUNGA SOTTOSEQUENZA COMUNE

(LONGEST COMMON SUBSEQUENCE : LCS)

- APPLICAZIONI BIOLOGICHE
  - CONFRONTO DNA DI DIVERSI ORGANISMI

- STRUTTURA DNA : FORMATA DA UNA STRINGA DI MOLECOLE DETTE BASI (ca. 3.2×10^9)



- BASI :
  - ADENINA (A)
  - CITOSINA (C)
  - GUANINA (G)
  - TIMINA (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

ES.

$S_1 = \text{ACCGGTCGAGTGCGCGGAAGCCGGCCGAA}$

$S_2 = \text{GTCGTTCGGAATGCCGTTGCTCTGTAAA}$

$S_3 = \text{GTCGTCGGAAGCCGGCCGAA}$

VI SONO VARI CRITERI DI SOMIGLIANZA:

- UNO DEI DUE FILAMENTI E' SOTTOSTRINGA DELL'ALTRO [STRING MATCHING]
- IL NUMERO DI MODIFICHE DA FARE AD UNO DEI DUE FILAMENTI DI DNA PER OTTENERE L'ALTRO E' "PICCOLO" [DISTANZA DI EDIT]
- C'E' UN "LUNGO" FILAMENTO, S_3, LE CUI BASI COMPAIONO SIA IN S_1 CHE IN S_2 (NELLO STESSO ORDINE, ANCHE SE IN POSIZIONI NON NECESSARIAMENTE CONSECUTIVE) [LCS]

DEF UNA **SOTTOSEQUENZA** DI UNA SEQUENZA DATA X E' UNA SEQUENZA OTTENUTA CANCELLANDO DA X ZERO O PIÙ ELEMENTI (MANTENENDO L'ORDINE).

PIÙ PRECISAMENTE, SE $X = x_1 x_2 \dots x_m$, UNA SEQUENZA $Z = z_1 z_2 \dots z_k$ E' UNA SOTTOSEQUENZA DI X SE ESISTE UNA SEQUENZA $\langle i_1, i_2, \dots, i_k \rangle$ DI INDICI TALE CHE:

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

$X = \dots x_{i_1} \dots x_{i_2} \dots x_{i_{k-1}} \dots x_{i_k} \dots$

$Z = z_1 \quad z_2 \dots z_{k-1} \quad z_k$

ESEMPIO SIA $X =$ A B C B D A B (posizioni 1 2 3 4 5 6 7).

ALLORA $Z =$ B C D B (posizioni 1 2 3 4) E' 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 E' UNA SOTTOSEQUENZA COMUNE DI X E Y SE Z E' UNA SOTTOSEQUENZA DI ENTRAMBE LE SEQUENZE X E Y

ESEMPIO DATE $X = \underset{1}{A}\ \underset{2}{B}\ \underset{3}{C}\ \underset{4}{B}\ \underset{5}{D}\ \underset{6}{A}\ \underset{7}{B}$ E

$Y = B\ D\ C\ A\ B\ A$,

- LA SEQUENZA $Z = B\ C\ A$ E' UNA SOTTOSEQUENZA COMUNE DI X E Y.
- TUTTAVIA $B\ C\ A$ NON E' LA PIÙ LUNGA SOTTOSEQUENZA COMUNE DI X E Y, IN QUANTO

  $B\ C\ B\ A$, $B\ C\ A\ B$, $B\ D\ A\ B$ 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.

1 2 3 4 5 6

SOLUZIONE MEDIANTE RICERCA ESAUSTIVA

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

COMPLESSITÀ: $O(\max(|X|,|Y|) \cdot 2^{\min(|X|,|Y|)})$

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

FASE 1: CARATTERIZZAZIONE DELLA PIÙ LUNGA SOTTOSEQUENZA COMUNE

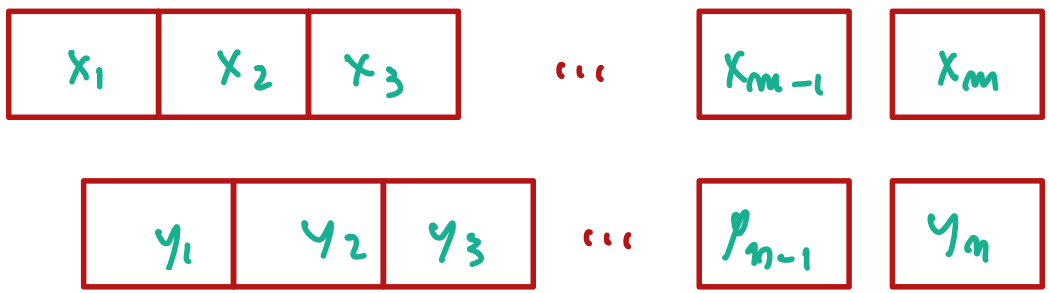
NOTAZIONE DATA UNA SEQUENZA $X = x_1 x_2 \dots x_m$, PONIAMO $X_i := x_1 x_2 \dots x_i$, PER $i = 0, 1, 2, \dots, m$.

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

ESEMPIO: SE $X = ABCBDAB$, ALLORA

$X_1 = A$, $X_4 = ABCB$, $X_0 = \varepsilon$ (SEQUENZA VUOTA)

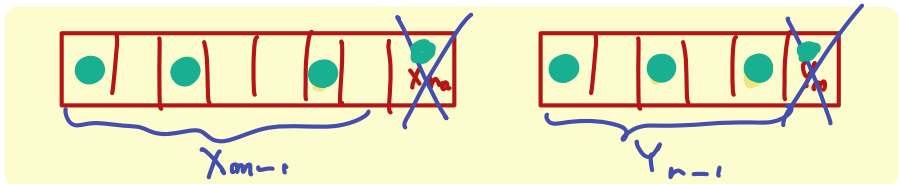
$X_3 D = ABCD$, ECC.



TEOREMA (SOTTOSTRUTTURA OTTIMA DI UNA LCS)

SIANO DATE DUE SEQUENZE $X = x_1 x_2 \dots x_m$ E $Y = y_1 y_2 \dots y_n$ TALI CHE $m \geq 1$ E $n \geq 1$,

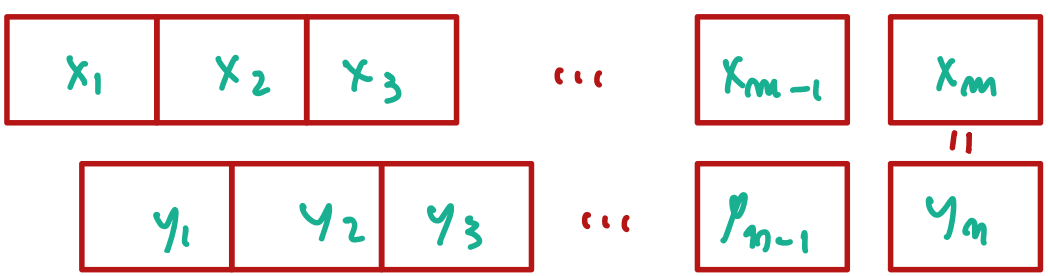
E SIA $Z = z_1 z_2 \dots z_k$ UNA LCS DI X E Y.



1. SE $x_m = y_n$, ALLORA
   - $z_k = x_m = y_n$ E
   - Z_{k-1} E' UNA LCS DI X_{m-1} E Y_{n-1}

2. SE $x_m \neq y_n$, ALLORA
   $z_k \neq x_m \implies Z$ E' UNA LCS DI X_{m-1} E Y

3. SE $x_m \neq y_n$, ALLORA
   $z_k \neq y_n \implies Z$ E' UNA LCS DI X E Y_{n-1}.



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

$x_m = y_m$

E QUINDI Zx_m E' UNA SOTTOSEQUENZA COMUNE DI X E Y, ASSURDO.

QUINDI $z_k = x_m = y_n$ E PERTANTO Z_{k-1} E' UNA SOTTOSEQUENZA COMUNE DI X_{m-1} E Y_{n-1}.

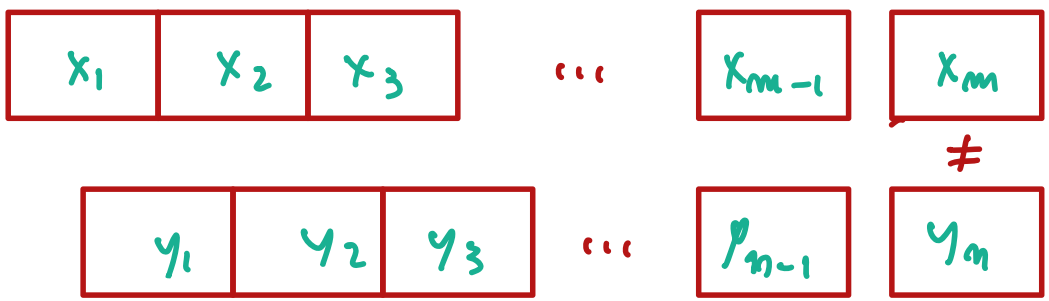
SE Z_{k-1} NON FOSSE DI LUNGEZZA MASSIMA, ESISTEREBBE W LCS DI X_{m-1} E Y_{n-1} TALE CHE $|W| > |Z_{k-1}|$.

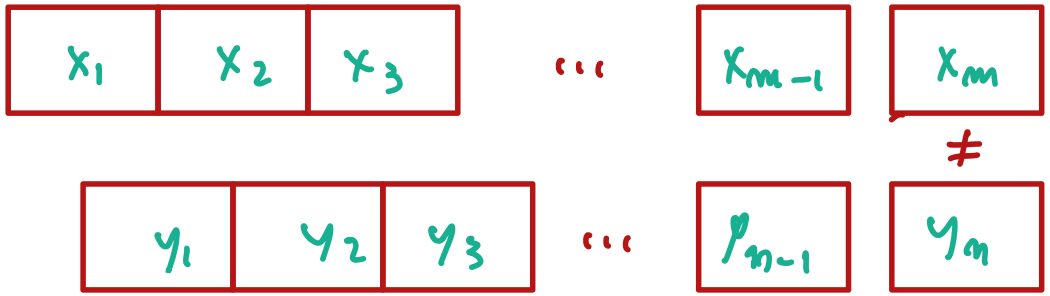
MA, ALLORA Wx_m SAREBBE UNA SOTTOSEQUENZA COMUNE DI X E Y, ASSURDO IN QUANTO $|Wx_m| > |Z|$ E Z E' UNA LCS DI X E Y

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

$x_m \neq y_m$

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) SE $z_k \neq y_m$ ALLORA Z E' UNA SOTTOSEQUENZA COMUNE DI X E Y_{m-1}.

$x_m \neq y_m$

SE ESISTESSE UNA SOTTOSEQUENZA COMUNE W DI X E Y_{m-1} TALE CHE $|W| > |Z|$, POICHE' W SAREBBE ANCHE UNA SOTTOSEQUENZA COMUNE X E Y, VERREBBE CONTRADDETTA LA MASSIMALITA' DI $|Z|$.

SPAZIO DEI SOTTOPROBLEMI

$$(X,Y) = (X_m, Y_n) \longmapsto (X_{m-1}, Y_{n-1}) \,/\, (X_m, Y_{n-1}) \,/\, (X_{m-1}, Y_n)$$

$$(X_{m-1}, Y_{n-1}) \longmapsto (X_{m-2}, Y_{n-2}) \,/\, (X_{m-1}, Y_{n-2}) \,/\, (X_{m-2}, Y_{n-1})$$

$$(X_m, Y_{n-1}) \longmapsto (X_{m-1}, Y_{n-2}) \,/\, (X_m, Y_{n-2}) \,/\, (X_{m-1}, Y_{n-1})$$

$$(X_{m-1}, Y_n) \longmapsto (X_{m-2}, Y_{n-1}) \,/\, (X_{m-1}, Y_{n-1}) \,/\, (X_{m-2}, Y_n)$$

$$\vdots$$

$$\{ (X_i, Y_j) \;:\; 0 \le i \le m \,,\; 0 \le j \le n \}$$

SPAZIO DEI SOTTOPROBLEMI

DAL PRECEDENTE TEOREMA SEGUE CHE LO SPAZIO DEI SOTTOPROBLEMI E' $\{(X_i, Y_j) : 0 \le i \le m, 0 \le j \le n\}$ LA CUI CARDINALITA' E' $O(mn)$.

FASE 2: SOLUZIONE RICORSIVA

DEFINIAMO $c[i,j]$, PER $0 \le i \le m$ E $0 \le j \le n$, 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 := length[X]
  n := length[Y]
  for i := 1 to m do
      c[i,0] := 0
  for j := 0 to n do
      c[0,j] := 0

  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] := "↖"
          else if c[i-1,j] ≥ c[i,j-1] then
              c[i,j] := c[i-1,j] ;  b[i,j] := "↑"
          else c[i,j] := c[i,j-1] ;  b[i,j] := "←"
  return c, b
```

CASO BASE

$O(m+n)$

DEFINIZIONE RICORSIVA

$O(mn)$

c[i,j]

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	↖	←	←	←	←
2	0	↑	↖	←	←	←
3	0	↑	↑	↖	←	←
4	0	↑	↑	↑	↖	←

COMPLESSITÀ DI LCS_LENGTH: $\Theta(mn)$

ESEMPIO: DATI $X = A\ B\ C\ B\ D\ A\ B$

$Y = B\ D\ C\ A\ B\ A$.

LCS_LENGTH (X, Y) CALCOLA LA TABELLA.

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

$Z = B\ C\ B\ A$

$B\ C\ A\ B$

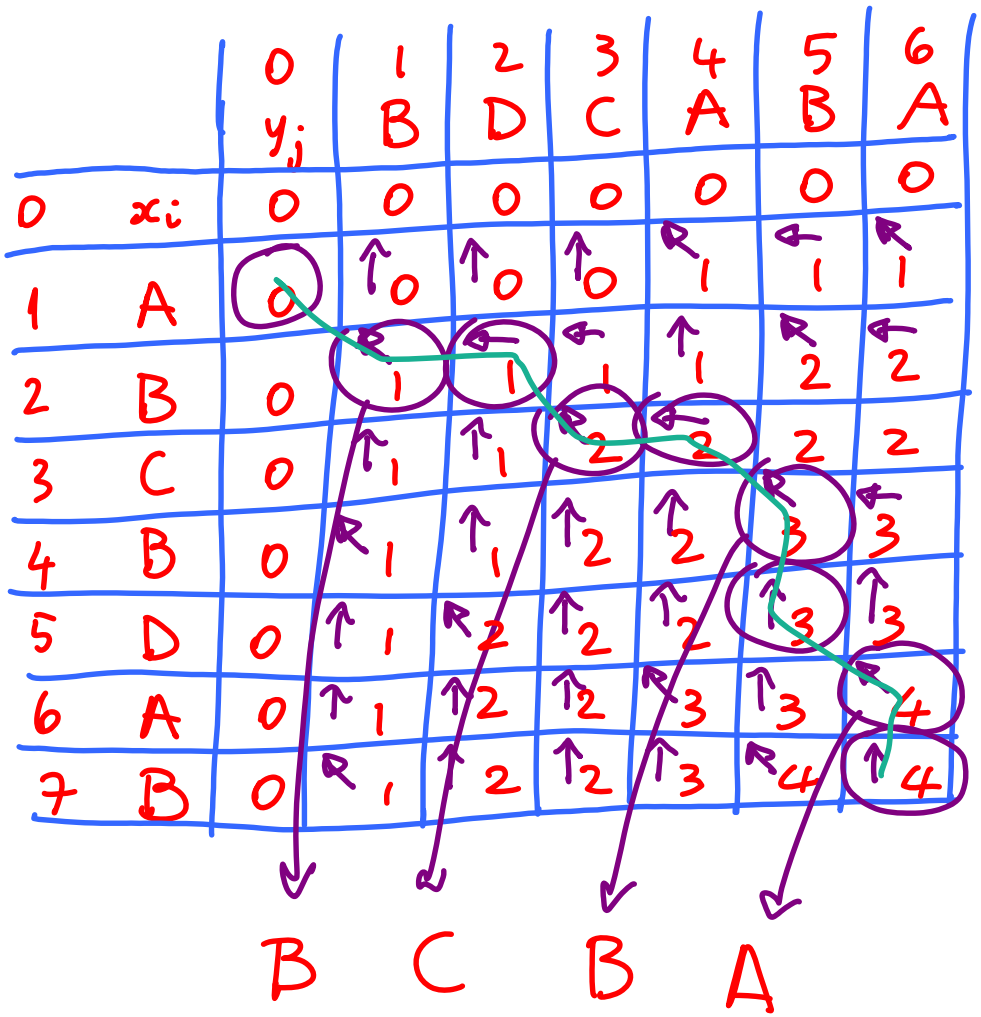
$B\ D\ A\ B$

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)
```

COMPLESSITÀ: $O(m+n)$

PRINT_LCS (b, X, m, n)



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

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.

		B	D	C	A	B	A
A							
B							
C							
B							
D							
A							
B							

0	0	0	0	0	0	0
0	0	0	0	1	1	1
0	1	1	1	1	2	2
0	1	1	2	2	2	2
0	1	1	2	2	3	3
0	1	2	2	2	3	3
0	1	2	2	3	3	4
0	1	2	2	3	4	4

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.

		B	D	C	A	B	A
A	0	1	1	1	1	2	2
B							
C							
B							
D							
A							
B							

0	1	1	2	2	2	2

0	0	0	0	1	1	1
0	1	1	1	1	2	2
0	1	1	2	2	2	2
0	1	1	1	1	3	3
0	1	2	2	2	3	3

0	1	2	2	3	4	4

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

SIA S UNA SEQUENZA DI NUMERI

- ORDINARE LA SEQUENZA S ELIMINANDO LE RIPETIZIONI.

SIA S_1 LA SEQUENZA OTTENUTA

- CERCARE LA LCS DI S ED S_1.

COMPLESSITÀ: $O(n^2)$