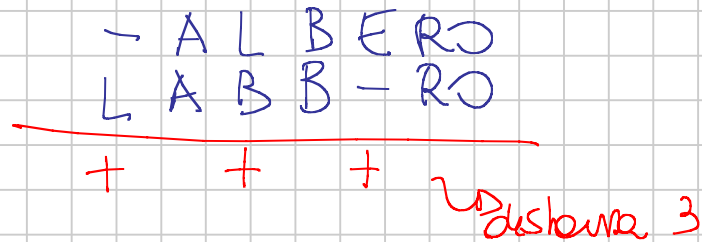


EDIT DISTANCE

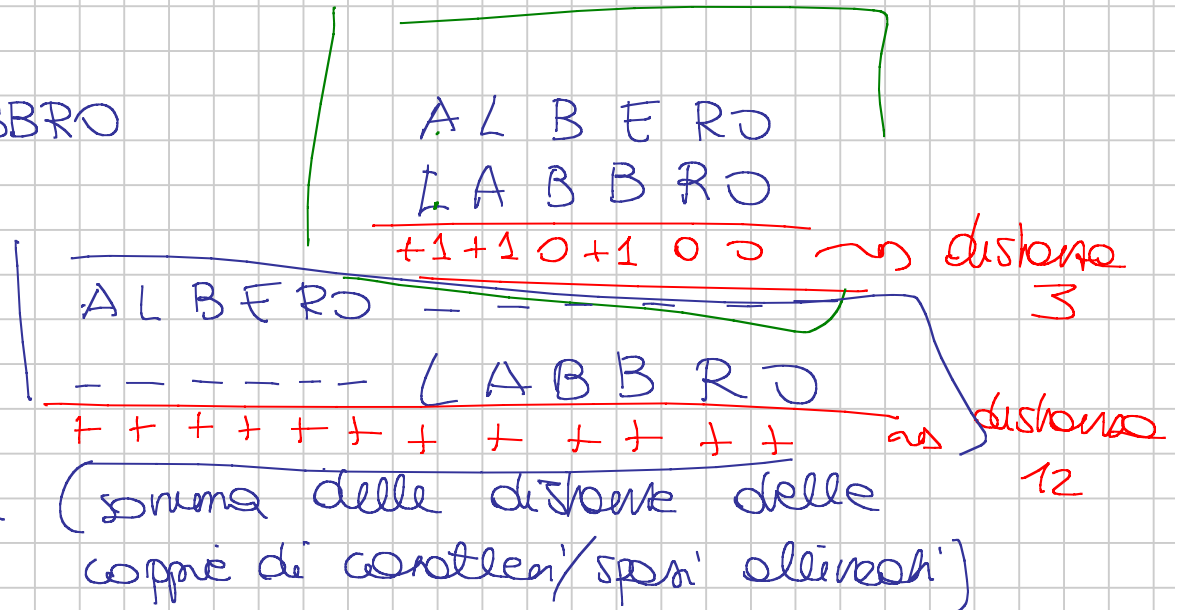
trovare un ALLINEAMENTO OTTIMO fra due sequenze X e Y

ogni carattere o "spazio" di X si fa corrispondere ad un carattere o spazio in Y.

X = ALBERO Y = LABBRO



Allineamenti → distanza



Calcolo della distanza tra caratter/spazi:

"MATCH" (caratteri uguali) \leadsto distanza = 0

"MISMATCH" (caratteri \neq tra loro, e \neq dello spazio) \leadsto distanza = 1.

"SPACE" (spazio - carattere o viceversa) \leadsto distanza = 1

Problema trovare l'allineamento ottimo, che minimizza la distanza tra le due sequenze

Allineamento → sequenza di operazioni che permettono di nascondere
X in Y

- sostituzione di un carattere con un altro (costo 1)
- cancellazione di un carattere (costo 1)
- inserimento di un carattere (costo 1)

X: A L - B E R O
 Y: - L A B B R O

~~X~~ L^A B^B E R O

- cancella A +1
 - mantieni L
 - inserisci A +1
 - mantieni B
 - sostituisci E con B +1
 - mantieni R
 - mantieni O
-
- costo 3

Algoritmo di PD (x EDIT DISTANCE / ALLINEAMENTO OTTIMO)

① Definizione dei sottoproblemi

$$X = x_1 x_2 x_3 \dots x_n$$

$$Y = y_1 y_2 y_3 \dots y_m$$

Sottoproblema $i-j$: trovare la edit-distance tra i prefissi

$$X_i \text{ e } Y_j$$

$$\underline{0 \leq i \leq n}, \underline{0 \leq j \leq m}$$

$$X_i = x_1 x_2 \dots x_i$$

$$Y_j = y_1 y_2 \dots y_j$$

tabella $(n+1) \times (m+1)$

$$M(i, j) = \underbrace{\text{edit-distanza}}_{\substack{\downarrow \\ \text{distanza} \\ \text{minima}}} \text{ tra } X_i \text{ e } Y_j \quad (n+1) \times (m+1)$$

② Sottoproblemi elementari

$$i=0, \text{ o } j=0$$

X_i

Y_0

$$\begin{array}{cccccc} x_1 & x_2 & x_3 & \dots & x_i \\ \hline + & + & + & \dots & + \end{array}$$

$$M(i, 0) = i \quad 0 \leq i \leq n$$

X_0

Y_j

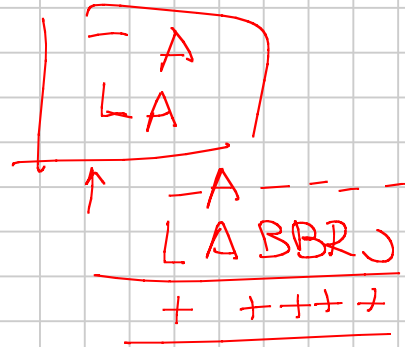
$$\begin{array}{cccc} y_1 & y_2 & \dots & y_j \\ \hline + & + & & + \end{array}$$

$$\text{costo } j \Rightarrow M(0, j) = j \quad 0 \leq j \leq m$$

X

	∅	L	A	B	B	R	O
∅	0	1	2	3	4	5	6
A	1	1	1	2	3	4	5
L	2	1	2	2	3	4	5
B	3	2	2	2	2	3	4
R	4	3	3	3	3	3	4
O	5	4	4	4	4	3	4
O	6	5	5	5	5	4	3

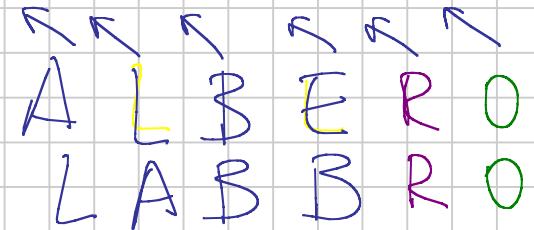
← L
 ← A L B E R O
 ← A B B - R O



costo 5

$M(n, m) = 3$

edit-distance 3



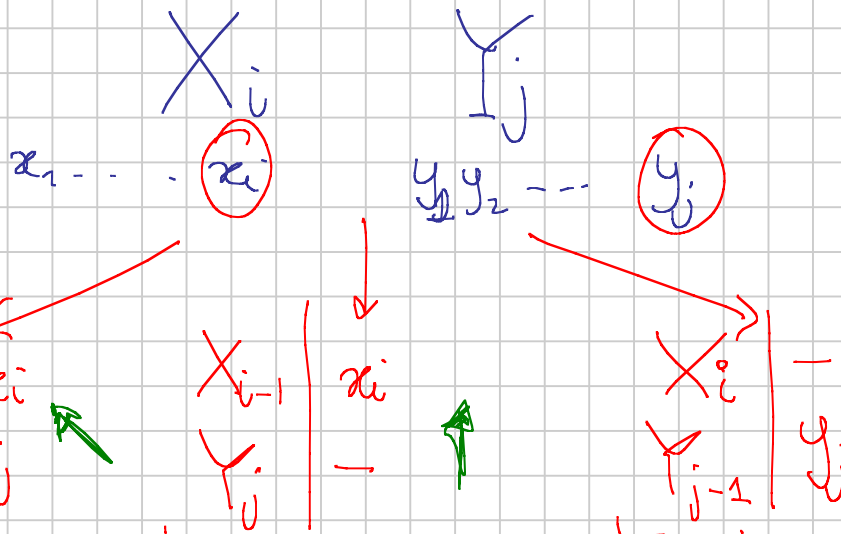
+ + +

(3)

③ Regola ricorsiva

$$P = \begin{cases} 0 \\ 1 \end{cases}$$

$$\begin{aligned} x_i &= y_j \\ x_i &\neq y_j \end{aligned}$$



allineamento ottimo
ha i prefissi X_{i-1}, Y_{j-1}

$$M[i-1, j-2] + P$$

allineamento ottimo

$$M[i-1, j] + 1$$

all. ottimo

$$M[i, j-1] + 1$$

SCELGO IL MINIMO

$$M(i, j) = \begin{cases} 0 & i=0, j=0 \\ i & i \neq 0, j=0 \\ j & i=0, j \neq 0 \\ \min \left\{ \begin{array}{l} M(i-1, j-1) + P \\ M(i-1, j) + 1 \\ M(i, j-1) + 1 \end{array} \right\} & \begin{array}{l} i > 0 \\ j > 0 \end{array} \end{cases}$$

h) Risultato $M(n, m) = \text{edit distance } X_n \text{ e } Y_m (X \text{ e } Y)$

ED(X, p) // X e Y array di dim n e m X[1...n] Y[1...m]

① M = nuova matrice (n+1) x (m+1) // M[0...n, 0...m]

② for i = 0 to n { M[i, 0] = i }
 for j = 1 to m { M[0, j] = j }

③ for i = 1 to n {
 for j = 1 to m {
 if (X[i] == Y[j]) p = 0;
 else p = 1;
 M[i, j] = min { M[i-1, j-1] + p, M[i-1, j] + 1, M[i, j-1] + 1 }
 }
 }

④ return M[n, m]

$$T(n, m) = \Theta(n \cdot m)$$

ALLINEA(X, Y, M) // restituisce ALLX e ALLY

ALLX = nuovo array di dimensione $m+n$

ALLY = " " " " " " $m+n$

$h = m+n$;

$i = n$

$j = m$

while ($i > 0 \parallel j > 0$)

if ($i > 0 \ \&\& \ j > 0 \ \&\&$
 $(M[i,j] == M[i-1,j-1]) \ \&\& \ X[i] == Y[j])$

← $(M[i,j] == M[i-1,j-1] + 1 \ \&\& \ \underline{X[i] \neq Y[j]})$

$ALLX[h] = X[i];$

$ALLY[h] = Y[j];$

$i--; j--;$

x_i
 y_j

else if ($j > 0$ && $M[i,j] == M[i,j-1] + 1$) {

ALLX[h] = -
 ALLY[h] = Y[j]
 j--;

Y
j

$$T(n, m) = O(n + m)$$

↑ else { ALLX[h] = ~~X~~ X[i];
 ALLY[h] = -;
 i--;

↳ h--;

↳ return h, ALLX, ALLY

M → Elenco operazioni per trasformare X in Y

ELENCO (X, Y, M, i, j)

if ($i > 0 \parallel j > 0$) {

if ($i > 0 \ \&\& \ j > 0 \ \&\& \ M(i, j) == M(i-1, j-1) \ \&\& \ X(i) == Y(j)$) {

ELENCO (X, Y, M, i-1, j-1);

PRINT "MANTIENI X(i)"

}

else if ($i > 0 \ \&\& \ j > 0 \ \&\& \ M(i, j) == M(i-1, j-1) + 1$) {

ELENCO (X, Y, M, i-1, j-1);

PRINT "SOSTITUISCI X(i) CON Y(j)"

}

else if ($j > 0 \ \&\& \ M(i, j) == M(i, j-1) + 1$) {

ELENCO(X, Y, M, i, j-1);

PRINT "INSERISCI Y[j]";

else {
ELENCO(X, Y, M, i-1, j);

PRINT "CANCELLA X(i)";

}

}