

# Programmazione dinamica

- problemi di ottimizzazione  $\left\{ \begin{array}{l} \text{max} \\ \text{min} \end{array} \right.$  LCS

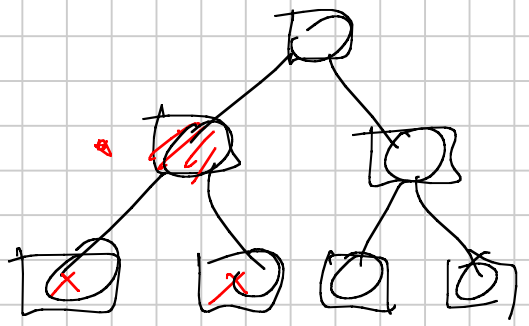
1) Caratterizzazione delle strutture delle soluzioni ottime.

-  $LCS(X, Y)$   $|X| = n$   $|Y| = m$

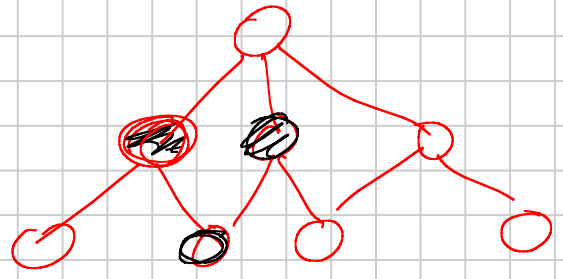
-  $LCS(X_i, Y_j)$   $X_i = x_1 \dots x_i$   $Y_j = y_1 \dots y_j$

2) Definizione delle regole ricorsive  
condizioni iniziali  
regole generali

3) Memorizzazione  
delle sottosoluzioni



D. I.



P. D.

4) Dimostrazioni dell'ottimalità dei sottoproblemi.

# EDIT DISTANCE

## Elaborazione di testi

### Biologie computazionale

$$X = x_1 \dots x_m$$

$$Y = y_1 \dots y_n$$

distanza = numero di errori  
nel confronto di X e Y

### errori

1) mismatch : caratteri diversi

2) spazio che corrisponde a un carattere di X inserzione

3) spazio che " " " " di Y cancellazione

X ALBERO  
Y LABBRO  
• • •

} 3 errori

ALBERO  
LABBRO

2 ← • 3 • ← 1

distanza = 3

# Alg. Programmazione Dinamica per l'EDIT DISTANCE

$ED(X_m, Y_m) = la$  EDIT DISTANCE TRA  $X$  e  $Y$

$ED(X_i, Y_j)$  = la EDIT DISTANCE DEI PREFISSI

$X_i$  e  $Y_j$  di  $X$  e  $Y$

Regola ricorsiva

$$ED(\phi, Y_j) = j$$

$$ED(X_i, \phi) = i$$

	$\phi$	L	A	B	B	R	O
$\phi$	$\phi$	1	2	3	4	5	6
A	1						
L	2						
B	3						
.							
.							

$ED(i, j)$

$ED(i-1, j-1)$

$ED(i-1, j)$



Soluzione dell'ED(x,y)

Soluzione in E(n+1, m+1)

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

Spazio aggiuntivo  $\Theta(n \cdot m)$

$M [0..n, 0..m]$

		$\phi$	L	A	B	B	R	O
$\phi$	$\phi$	0	1	2	3	4	5	6
A	1	1	2	3	4	4	5	6
L	2	1	2	2	3	3	4	5
B	3	2	2	3	3	4	4	5
E	4	3	3	3	3	3	3	4
R	5	4	4	4	4	4	4	4
O	6	5	5	5	5	4	4	3

A L B E R O  
L A B B R O  
• • •

→ A L B E R O  
L A B B R O

A L L I B E R O  
L L A B B R O

ALX  $n + m$   
ALY

$O(m+n)$

$k = m+n$ ;  $h = k$ ;  $i = m$ ;  $j = n$

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

if  $((i > 0) \&\& (j > 0) \&\& (M[i,j] == M[i-1,j-1] \&\& x[i] == y[j])$   
 OR  $(M[i,j] == M[i-1,j-1] + 1) \&\& x[i] != y[j])$  {

$ALX[h] = x[i]$ ;  $ALY[h] = y[j]$ ;  $i--$ ;  $j--$ ;

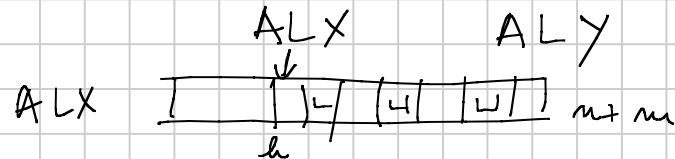
else {

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

$ALX[h] = -$ ;  $ALY[h] = y[j]$ ;  $j = j - 1$ ;

else  $ALX[h] = x[i]$ ;  $ALY[h] = -$ ;  $i = i - 1$ ;

$h = h - 1$ ; } return  $h$ ;



allinea  $x[i]$  con  $y[j]$   
 (diagonale)

allinea  $y[j]$  con  $h$   
 (colonne)

allinea  $x[i]$  con  $h$   
 (righe).

# APPARIZIONI APPROSSIMATE

APPROXIMATE

STRING MATCHING

pattern

$$X = m$$

regola ricorsiva =

testo

$$Y = n$$

$$m \gg n$$

Edit Distance



$Y = \text{SER} \overset{1}{\boxed{\text{RATUR}}} \text{A}$   
 $X = \text{RAT} \underset{0}{\boxed{\text{A}}}$

Si vogliono trovare tutte le apparizioni di  $X$  in  $Y$  con al più  $k$  errori.

$$AP(\phi, Y_j) = \phi \quad \forall j$$

$$AP(X_i, \phi) = i$$

Risultati in  $AP(m, j) \quad \forall j$

