

Longest Common Subsequence

Deriva dal problema di fare matching tra due sequenze per vedere quanto sono simili

Questo problema gode di una struttura ottimale il che lo rende ideale per un approccio con la programmazione dinamica

Si potrebbe usare un approccio brute-force ma per sequenze grandi sarebbe inusabile in quanto richiede tempo esponenziale

ALGORITMO PROPOSTO CON PROGRAMMAZIONE DINAMICA

LCS-LENGTH (X, Y)

$m \leftarrow X.length$

$n \leftarrow Y.length$

$b \leftarrow \text{NEW MATRIX } [m][n]$

$c \leftarrow \text{NEW MATRIX } [m][n]$

FOR $i \leftarrow 1$ TO m

$c[i][0] \leftarrow 0$

FOR $j \leftarrow 1$ TO n

$c[0][j] \leftarrow 0$

FOR $i \leftarrow 0$ TO m

FOR $j \leftarrow 0$ TO n

IF ($x_i == y_j$)

$c[i][j] \leftarrow c[i-1][j-1] + 1$

$b[i][j] \leftarrow \nwarrow$

```

ELSE IF  $c[i-1][j] \geq c[i][j-1]$ 
     $c[i][j] \leftarrow c[i-1][j]$ 
     $b[i][j] \leftarrow "\uparrow"$ 
ELSE
     $c[i][j] \leftarrow c[i][j-1]$ 
     $b[i][j] \leftarrow "\leftarrow"$ 
RETURN  $b, c$ 

```

La matrice c contiene i valori del matching
mentre la matrice b la tabella per ricostruire
il percorso