

LES(e, b, e)

if  $e[n, m] = 0$  then return()

if  $e[n, m] = e[n-1, m]$  then

LES(e(n-1), b, e)

else if  $e[n, m] = e[n, m-1]$  then

LES(e, b(m-1), e)

else

LES(e(n-1), b(m-1), e)

↑  
print(e[n])

## ESERCIZIO 11 di PD

Vogliamo realizzare ~~un algoritmo~~ un algoritmo che permetta di nel minor numero di op. di inserimento, cancellazioni e sostituzioni di trasformare una stringa s in t.

Notabilmente è sempre possibile trasformare una data stringa s, in t, cancellando tutti i caratteri di s lunghi m e aggiungendo quelli di t lunghi m, ma noi vogliamo un algoritmo che è ~~realizzare~~ ottenere questo nel minor numero di op.  $dist(s, t) \leq m + m$

EDT-DISTANZA(s, t)

for  $i \leftarrow 0$  to  $m$        $dist[i, 0] \leftarrow i$

for  $j \leftarrow 0$  to  $m$        $dist[0, j] \leftarrow j$

for  $i \leftarrow 1$  to  $m$

for  $j \leftarrow 1$  to  $m$

if  $s[i] = t[j]$

$dist[i, j] \leftarrow \min(dist[i-1, j-1], dist[i-1, j]+1, dist[i, j-1]+1)$

else

$dist[i, j] \leftarrow \min(dist[i-1, j-1]+1, dist[i-1, j]+1, dist[i, j-1]+1)$