9/04/2024 | LEZ 25 | PROGRAMMAZIONE DINAMICA

VEDREMO ALTRI 2 PROBLEMI → OGGI VEDIAMO IL 1° + SUA OTTIMIZZAZIONE

1 SEQUENCE ALLIGMENT

INTROD. INFORMAL.: RICONOSCERE QUANDO 2 STRINGHE SONO SIMILI.

- PROBLEMA MOLTO COMUNE E APPLICATO: ? GOOGLE
- MA x STUDIARLO, INTRO PARAMETRI x GIUDICARE "SIMILARITÀ" TRA WORD.

STRUMENTO DI CONFRONTO

EDIT DISTANCE ⟹ costo per 'transformare' la prima parola nella seconda (riallineare).

DOVE:

$$COSTO = \delta + \alpha_{pq}$$

δ = COSTO INSERIMENTO GAP

α_{pq} = COSTO MISMATCHED TRA P e Q → DISALLINEAMENTO

ES:

PALETTE
PALATE

ASSUMENDO $\delta = 2$, $\alpha = 1$

∀ OP., AVREMO COSTO FINALE 3

N.B. IL COSTO DIPENDE DAL TIPO DI PARAMETRO DI MISURA ADOTTATO, ESEMPIO:

- x una ricera di google, può essere più grave (costo alto) due parole che sono lontane dalla tastiera, o che hanno un suono fonetico molto diverso

NEXT PAGE: DEF. FORMALE OF PROBLEMA

PROBLEMA: 2 STRINGHE $x_1, \dots, x_n$, $y_1, \dots, y_m$.

GOAL: MIN COSTO-ALLINEAMENTO. → MIN. EDIT. DISTANCE

N.B. ALLINEAMENTO: SET M DI COPPIE $X_i - Y_j$ (MATCHING), t.c. NOT INCROCIATI (2 COPPIE $X_{i'} - Y_{j'}$, $X_i - Y_j \Rightarrow i < i'$ e $j < j'$)

COSTO DI M

$$\text{COST}(M) = \overbrace{\sum_{(x_i, y_i) \in M} \alpha_{x_i, y_i}}^{\text{MISMATCHED}} + \overbrace{\sum_{\substack{i : y_i \ \emptyset \text{ MATCH} \\ \downarrow \\ \text{VUOTO}}} \delta + \sum_{x_i : i \ \emptyset \text{ MATCH}} \delta}^{\text{GAP}}$$

COSTO PICCOLO
⇓
PAROLE SIMILI

ALGORITMO

CONSIDERO 2 STRINGHE:

$x_1, x_2, \dots, x_n$
$y_1, y_2, \dots, y_j$

QUALI SONO I SOTTOPROBLEMI?

SOTTOPROBLEMA

- $OPT(i, J)$ = MIN. COST × ALLIGN $x_1, x_2, \dots, x_i$, $y_1, y_2, \dots, y_j$

GOAL
$OPT(n, m)$ → AL PASSO $OPT(i, J)$ CHE AVRÀ FATTO ALGO?

3 CASI

1. OPT CHOOSE $x_i - y_j$

$$OPT(i,j) \; \underline{\alpha_{ij} + OPT(i-1, j-1)}$$

2. OPT CHOOSE $(x_i, j) \rightarrow$ GAP

$$OPT(i,j) = \underline{\delta + OPT(x_{i-1}, y_j)}$$

3. " " (i, y_j) "

$$OPT(i,j) = \underline{\delta + OPT(x_i, y_{j-1})}.$$

$j = 1 \downarrow$ ESSE GAP

GAP $\uparrow$ $i = 0$

EQ. BELLMAN:

$$\begin{cases} j\delta & i = 0 \\ i\delta & j = 0 \\ \min \begin{cases} \alpha_{x_i y_i} + OPT(i-1, y-1) \;]\; \text{MISMATCH} \\ \delta + OPT(i-1, j) \\ \delta + OPT(i, j-1). \end{cases} \Bigg] \text{GAP} \end{cases} \rightsquigarrow$$

CODE

SEQUENCE-ALIGNMENT(m, n, x_1, …, x_m, y_1, …, y_n, δ, α)

```
FOR i = 0 TO m
   M[i, 0] ← i δ.
FOR j = 0 TO n
   M[0, j] ← j δ.

FOR i = 1 TO m
   FOR j = 1 TO n
      M[i, j] ← min { α_{x_i y_j} + M[i–1, j–1],
                      δ + M[i–1, j],
                      δ + M[i, j–1] }.

RETURN M[m, n].
```

already computed

1 2 EQ. BELL.

TAB = MATRIX DI POSS. RISULT. (= A KNAPSACK)

- CREA MATRIX = $\Theta(n \cdot m)$
- ACC. CASELLA = $O(1)$
- COSTO VISITA MATRIX = $\Theta(n \cdot m)$

COSTO

$\Theta(n \cdot m)$ TIME $\wedge$ SPACE

ALGO. HIRSCHBERG

ALGO X S. ALLINMENT U :

TIME = $\Theta(n \cdot m)$

SPACE = $\Theta(n+m)$

MIX TRA
P.D.
+
DIVIDE ET IMPERA

COME FARE?

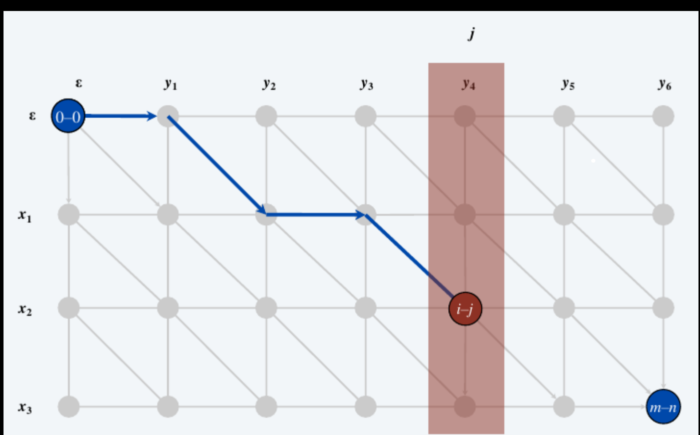
1° APPROCCIO ($\emptyset$ BUONO)

- $\forall$ OGNI CALCOLO DI (i,J), MANTENGO IN MEMORIA SOLO LA PRECEDENTE RIGA E COLONNA
- COSTO SPACE = $\Theta(n+m)$ → n = RIGA
  m = COLONNA

- CALCOLI EDIT DISTANCE, MA $\emptyset$ FIND LA SEQUENZA.

2° APPROCCIO: GRAFO OF EDIT DISTANCE

MATRICE = GRAFO.



- CON $F(i,J)$ = LUNGHEZZA OF SHORTEST PATH $(0,0) \rightsquigarrow (i,J)$

LEMMA

$F(i,J) = OPT(i,J) \quad \forall\, i, J$

DIM

Pf of Lemma. [by strong induction on $i + j$] FORTE INDUZIONE $i+J$

- Base case: $f(0, 0) = OPT(0, 0) = 0$.
- Inductive hypothesis: assume true for all (i', j') with $i' + j' < i + j$.
- Last edge on shortest path to (i, j) is from $(i-1, j-1)$, $(i-1, j)$, or $(i, j-1)$.
- Thus,

$$
\begin{aligned}
f(i,j) &= \min\{\alpha_{x_i y_j} + f(i-1, j-1),\ \delta + f(i-1, j),\ \delta + f(i, j-1)\} \\
&= \min\{\alpha_{x_i y_j} + OPT(i-1, j-1),\ \delta + OPT(i-1, j),\ \delta + OPT(i, j-1)\} \\
&= OPT(i,j) \, .
\end{aligned}
$$

IPOT. INDUT.

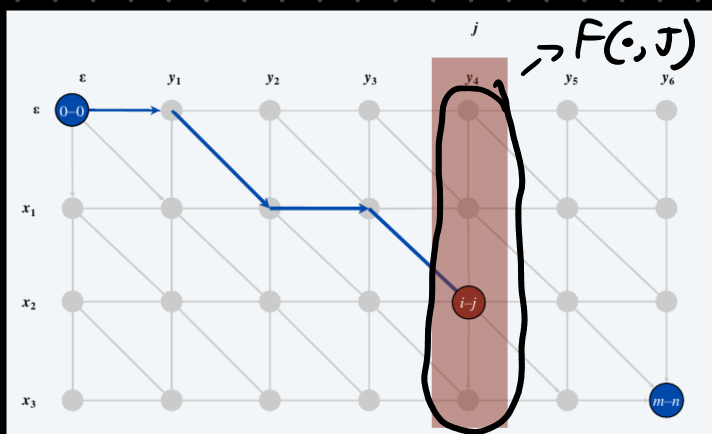
CON F, AGGIUNGIAMO ....

$y(i,J)$ = LENGHT OF S.P. $(i,J) \rightsquigarrow (m,n)$

COME CALCOLO F e y ?

- F

SIA $F(\cdot, J)$.



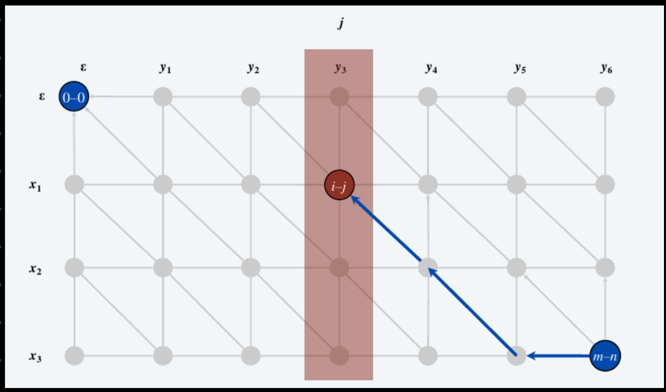
$\to F(\cdot, J)$

- POSSO CALCOLARE $F(\cdot, J)$ $\times \forall J$ IN:

$\rightsquigarrow$ USANDO ALGO. PRIMA

- TIME $= O(n\, m)$
- SPACE $= O(n+m)$ $\hookrightarrow$ CALCOLO MANTENENDO 2 COLONNE $\times$ VOLTA

• $y \to$ COME F, MA "INVERTO" VERSO OF ARCHI E CALCOLO



ORA, CONSIDERANDO UNA COLONNA J, IL PATH XFORZA PASSA BY J, QUINDI:

CON (q, J) = PUNTO IN J ∈ PATH,

$$S.P. = F(q, J) + y(q, J)$$

ALGO

• SIA $J = n/2$

• ∄ q t.c. $\rightsquigarrow \underbrace{F(q, n/2) + y(q, n/2)}$ MINIMO

↳ x DISCORSO PRIMA ∃ PATH OPT CHE PASSA x $(q, n/2)$

• MEMORIZZO $(q, n/2)$

• 2 CHIAMATE RICORS.

↳ · UNO CON PARAMETRO PATH (0,0) ⇝ (q, n/2)

· // // // (q, n/2) ⇝ (m, n)

GOAL ⇝ CREO RICORSIVAMENTE PATH ALLINEAMENTO

COSTO

MEMORY = $\Theta(n+m)$

Pf.
- Each recursive call uses $\Theta(m)$ space to compute $f(\cdot, n/2)$ and $g(\cdot, n/2)$.
- Only $\Theta(1)$ space needs to be maintained per recursive call.
- Number of recursive calls $\leq n$. ▪

→ 2 COLONNE X VOLTA
↳ QUANDO TROVATO

TIME = $\Theta(n \cdot m)$

Theorem. Let $T(m, n)$ = max running time of Hirschberg's algorithm on strings of lengths at most m and n. Then, $T(m, n) = O(mn)$.

Pf. [by strong induction on $m + n$]
- $O(mn)$ time to compute $f(\cdot, n/2)$ and $g(\cdot, n/2)$ and find index q.
- $T(q, n/2) + T(m - q, n/2)$ time for two recursive calls.
- Choose constant c so that: $T(m, 2) \leq cm$
  $T(2, n) \leq cn$
  $T(m, n) \leq cmn + T(q, n/2) + T(m - q, n/2)$
- Claim. $T(m, n) \leq 2cmn$.
- Base cases: $m = 2$ and $n = 2$.
- Inductive hypothesis: $T(m', n') \leq 2cm'n'$ for all (m', n') with $m' + n' < m + n$.

$$\begin{aligned} T(m, n) &\leq T(q, n/2) + T(m-q, n/2) + cmn \\ &\leq 2cqn/2 + 2c(m-q)n/2 + cmn \\ &= cqn + cmn - cqn + cmn \\ &= 2cmn \; ▪ \end{aligned}$$

inductive hypothesis