Lezione 8

8.1 Proprietà di sottostruttura ottima di MATRIX CHAIN MULTIPLICATION

MATRIX-CHAIN MULTIPLICATION

Input: una sequenza $\langle A_1, \dots, A_n \rangle$ di n matrici di dimensioni tali che $\#col(A_i) = \#rows(A_{i+1})$, $\forall i = 1, \dots, n-1$

Goal: calcolare $A_1 \cdots A_n$ minimizzando il numero di moltiplicazioni scalari.

Una Soluzione è una parentesizzazione

costo (P) = # moltiplicazioni scalari della parentesizzazione.

PROP: Il problema MATRIX-CHAIN MULTIPLICATION ha la proprietà di sottostruttura ottima.

Proof: Sia P_{opt} una parentesizzazione ottima, ovvero tale che

$$costo(P_{opt}) \leq costo(P) \quad \forall \text{ parentesizzazione } P \text{ di } A_1 \cdots A_n.$$

$$P_{opt}(A_1 \cdots A_n) = P_1(A_1 \cdots A_{i-1})\, P_2(A_i \cdots A_n), \quad \exists P_1, P_2, \ \exists i \in \{2, \dots, n\}$$

$$costo(P_{opt}) = costo(P_1) + costo(P_2) + h$$

h: # di moltiplicazioni scalari per moltiplicare la matrice risultante da $A_1 \cdots A_{i-1}$ per quella risultante da $A_i \cdots A_n$. Questo valore non dipende da P_1 o da P_2 ma solo da i (che è fissato).

Restringiamo P_{opt} ai sottoproblemi $\begin{cases} \langle A_1, \dots, A_i \rangle \longrightarrow P_1 \\ \langle A_i, \dots, A_n \rangle \longrightarrow P_2 \end{cases}$

Per mostrare che MATRIX CHAIN MULTIPLICATION ha la proprietà di sottostruttura ottima dobbiamo mostrare che

$$\begin{cases} P_1 \text{ è una parentesizzazione ottima di } A_1 \cdots A_{i-1}, \\ P_2 \text{ è una parentesizzazione ottima di } A_i \cdots A_n. \end{cases}$$

Supponiamo per assurdo che P_1 non sia una parentesizzazione ottima di $A_1 \cdots A_{i-1}$, questo vuol dire che $\exists$ una parentesizzazione P_1' di $A_1 \cdots A_{i-1}$ che ha costo inferiore a quello di P_1

$$\text{costo}(P_1') \underset{(*)}{<} \text{costo}(P_1).$$

Definisco allora la parentesizzazione P^*

$$P^*(A_1 \cdots A_n) = P_1'(A_1 \cdots A_{i-1})\, P_2(A_i \cdots A_n)$$

$$\text{costo}(P^*) = \text{costo}(P_1') + \text{costo}(P_2) + h < \underbrace{\text{costo}(P_1) + \text{costo}(P_2) + h}_{\text{costo}(P_{opt})}$$

$$\rightarrow \text{costo}(P^*) < \text{cost}(P_{opt})$$

$\not\ast$ Assurdo

nessuna parentesizzazione di $A_1 \cdots A_n$ può avere costo inferiore di P_{opt}.

$\Rightarrow P_1$ è ottima per $A_1 \cdots A_{i-1}$.

Analogamente si prova che P_2 è ottima per $A_i \cdots A_n$. $\square$

8.2 Quando è possibile applicare la programmazione dinamica.

Consideriamo i seguenti due problemi aventi lo stesso input

INPUT: Un grafo $G=(V,E)$ e due vertici $u,v \in V$.

UNWEIGHTED SHORTEST PATH: trovare un cammino da u a v che abbia lunghezza minima.

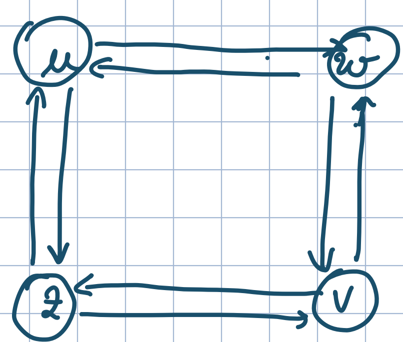
LONGEST SIMPLE PATH: trovare un cammino semplice da u a v che abbia lunghezza massima.

La lunghezza di un cammino $p=\langle a_0, a_1, \dots, a_n \rangle$ è $\ell(p)=n$ il numero di archi che contiene (x il grafo non è pesato).

Ex: UNWEIGHTED SHORTEST PATH ha la proprietà di sottostruttura ottima.

→ possiamo risolvere UNWEIGHTED SHORTEST PATH con la programmazione dinamica.

LONGEST SIMPLE PATH non gode della proprietà di sottostruttura ottima.



Cammino semplice più lungo da u a v

$u \longrightarrow w \longrightarrow v$

Sol ottima

$u \to w$; $w \to v$ sono ottimi per u,w ? w,v ? No, $u \to z \to v \to w$ è semplice ed è più lungo di $u \to w$

↓

non vale la proprietà di sottostruttura ottima

Non posso applicare la programmazione dinamica. ⟵

Longest simple Path NP-hard.

8.3 LONGEST COMMON SUBSEQUENCE (LCS)

INPUT: due sequenze $X = \langle x_1, \dots, x_m \rangle$ e $Y = \langle y_1, \dots, y_n \rangle$.

GOAL: trovare una sottosequenza Z comune a X e Y che abbia lunghezza massima.

Esempio:
$X = \langle A\ B\ C\ B\ D\ A\ B \rangle \quad m = 7$

$Y = \langle B\ D\ C\ A\ B\ A \rangle \quad n = 6$

$Z = LCS(X,Y) = \langle B\ C\ B\ A \rangle$, $length(Z) = 4$

e non esistono sequenze comuni di lunghezza 5.

Data una sequenza $X = \langle x_1, x_2, \dots, x_m \rangle$ e una sequenza $Z = \langle z_1, \dots, z_k \rangle$, diciamo che Z è una sottosequenza di X se $\exists$ una sequenza crescente di indici $\{i_1, \dots, i_k\}$ di X, $x_{i_j} = z_j$.

Nell'esempio precedente $Z = \langle B\ C\ B\ A \rangle$ la sequenza di indici di X è $\langle 2, 3, 4, 6 \rangle$

perché
$x_2 = z_1 \rightarrow i_1 = 2$, $x_6 = z_4 \rightarrow i_4 = 6$

$x_3 = z_2 \rightarrow i_2 = 3$

$x_4 = z_3 \rightarrow i_3 = 4$

$x_{i_1} x_{i_2} x_{i_3} x_{i_4} = z_1\ z_2\ z_3\ z_4$

Approccio brute-force: enumerare tutte le sottosequenze di X e controllare per ognuna di esse se è una sottosequenza di Y, tenendo traccia della sottosequenza comune più lunga.

Ogni sottosequenza di X corrisponde a un sottoinsieme di indici di $\{1, 2, \dots, m\}$.

→ Ci sono 2^m sottosequenze da controllare.

⇒ richiede tempo esponenziale.

8.3.1 PROPRIETÀ DI SOTTOSTRUTTURA OTTIMA DI LCS

L'i-esimo prefisso di $X = \langle x_1, x_2, \dots, x_m \rangle$ è $X_i = \langle x_1, \dots, x_i \rangle$, per $i = 0, 1, \dots, m$.

<u>THM (SOTTOSTRUTTURA OTTIMA DI LCS)</u>: Siano $X = \langle x_1, x_2, \dots, x_m \rangle$ e $Y = \langle y_1, y_2, \dots, y_n \rangle$ sequenze e sia $Z = \langle z_1, z_2, \dots, z_k \rangle$ una LCS di X e Y.

1. Se $x_m = y_n \Rightarrow z_k = x_m = y_n$ e Z_{k-1} è LCS di X_{m-1} e Y_{n-1}.
2. Se $x_m \neq y_n$ e $z_k \neq x_m$ e Z è LCS di X_{m-1} e Y.
3. Se $x_m \neq y_n$ e $z_k \neq y_n$ e Z è LCS di X e Y_{n-1}.

<u>PROOF</u>:

1. • Se $z_k \neq x_m$ allora potremmo appendere/aggiungere $x_m = y_n$ a Z ottenendo una sottosequenza comune di lunghezza $k+1$ $\divideontimes$ Z è la più lunga → $z_k = x_m = y_n$

- Z_{k-1} è comune a X_{m-1} e Y_{n-1} e ha lunghezza $k-1$. Se $\exists$ una sequenza W comune a X_{m-1} e Y_{n-1} più lunga di Z_{k-1}, allora $W \cup \{x_m\}$ avrebbe lunghezza maggiore di k, sarebbe più lunga di Z $\#$.

2. Se $z_k \neq x_m \Rightarrow Z$ è comune a X_{m-1} e Y. Se $\exists$ W comune a X_{m-1} e Y di lunghezza maggiore a Z allora W sarebbe comune a X e Y e più lunga di Z $\#$ contraddicendo l'ipotesi.

3. Simmetrico al punto 2. $\square$

Il Teorema suggerisce un approccio ricorsivo (Funzione ricorsiva di LCS)

$$c[i,j] = \begin{cases} 0 & i=0, j=0 \\ c[i-1,j-1]+1 & i,j>0 \text{ e } x_i = y_j \\ \max(c[i,j-1], c[i-1,j]) & i,j>0 \text{ e } x_i \neq y_j \end{cases}$$

$\downarrow$

lunghezza di $LCS(X_i, Y_j)$

$X = \langle x_1, \ldots, x_{i-1}, \cancel{x_i} \rangle$

$Y = \langle y_1 \ldots, y_{j-1}, \cancel{y_j} \rangle$

$X_{i-1} = \langle x_1, \ldots, x_{i-1} \rangle$

$Y_j = \langle y_1 \ldots, y_j \rangle$

$$\left.\begin{aligned} X_i &= \langle x_1, \ldots, x_i \rangle \\ Y_{j-1} &= \langle y_1, \ldots, y_{j-1} \rangle \end{aligned}\right\}$$