Analisi e progettazione di algoritmi

(III anno Laurea Triennale - a.a. 2023/24)

Prova scritta 4 settembre 2024

Esercizio 1 Si considerino due sequenze X e Y di lunghezza $n = 3$ e $m = 4$ rispettivamente. L'algoritmo per il calcolo della LCS visto a lezione ha complessità $\Theta(n \cdot m)$, quindi in questo caso otteniamo, per qualunque coppia di sequenze, $T(3,4) = 12$; infatti, trascurando la riga e colonna iniziali di zeri, viene costruita una matrice $3 \cdot 4$.

1. Quale complessità $T_{best}(3,4)$ del caso migliore e $T_{worst}(3,4)$ del caso peggiore otteniamo per l'algoritmo che ricostruisce la sottosequenza a partire dalla matrice? Fate un esempio di caso migliore e uno di caso peggiore.
2. Quale complessità $T_{best}(3,4)$ del caso migliore e $T_{worst}(3,4)$ del caso peggiore otteniamo per l'algoritmo brute-force? Fate un esempio di caso migliore e uno di caso peggiore.
3. Quale complessità $T_{best}(7) = T_{best}(3,4)$ del caso migliore e $T_{worst}(7) = T_{worst}(3,4)$ del caso peggiore otteniamo per l'algoritmo ricorsivo divide-et-impera? Fate un esempio di caso migliore e uno di caso peggiore.

Soluzione

1. L'algoritmo che ricostruisce la sottosequenza a partire dalla matrice ha $T_{best}(3,4) = 3$, e questo si ha quando X è la parte finale di Y, quindi si percorre la diagonale, ossia si esaminano gli elementi LCS(3,4), LCS(2,3), ed LCS(1,2). Invece $T_{worst}(3,4) = 6$, e questo si ha, scegliendo arbitrariamente se andare a sinistra o in alto[1], quando X ed Y non hanno elementi in comune, quindi, si esaminano, per esempio, gli elementi LCS(3,4), LCS(3,3), LCS(3,2), LCS(3,1), LCS(2,1), LCS(1,1).
2. L'algoritmo brute-force genera tutte le sottosequenze di X, che sono $2^3 = 8$, e per ognuna di queste controlla se sia una sottosequenza di Y; questo controllo richiede di percorrere Y, quindi costa al più 4. Supponiamo di considerare le sottosequenze di X in ordine di lunghezza decrescente; allora, il caso migliore si ha quando X è una sottosequenza di Y, e costa 4 (più precisamente 3 se è la parte iniziale), il caso peggiore quando X ed Y non hanno elementi in comune e costa $8 \cdot 4 = 32$ (più precisamente $7 \cdot 4 = 28$ perchè per la sequenza vuota non serve controllare)[2].
3. L'algoritmo ricorsivo divide-et-impera ha complessità del caso migliore 3, e si ha quando X è la parte finale di Y; infatti in questo caso si ha

[1] Invece se si sceglie sempre una direzione il caso peggiore è quando solo il primo elemento è uguale.

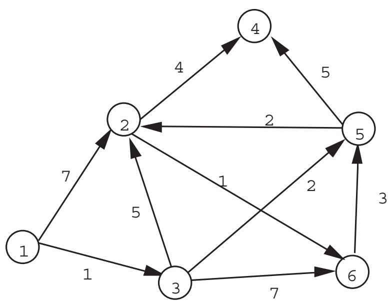
[2] Viceversa, considerando le sottosequenze in ordine di lunghezza crescente, il caso migliore si ha quando tutte le sottosequenze lunghe 1 di X non sono sottosequenze di Y, e costa $3 \cdot 4$, il caso peggiore quando X è una sottosequenza di Y.

$$
\begin{aligned}
T(7) &= T(5) + 1 = 3 \\
T(5) &= T(3) + 1 = 2 \\
T(3) &= T(1) + 1 = 1 \\
T(1) &= 0
\end{aligned}
$$

La complessità del caso peggiore è 63, e si ha quando X ed Y non hanno elementi in comune; infatti in questo caso si ha:

$$
\begin{aligned}
T(7) &= 2\,T(6) + 1 = 63 \\
T(6) &= 2\,T(5) + 1 = 31 \\
T(5) &= 2\,T(4) + 1 = 15 \\
T(4) &= 2\,T(3) + 1 = 7 \\
T(3) &= 2\,T(2) + 1 = 3 \\
T(2) &= 2\,T(1) + 1 = 1 \\
T(1) &= 0
\end{aligned}
$$

Esercizio 2 Si consideri il seguente grafo orientato e pesato.



1. Applicando l'algoritmo di Dijkstra, si determinino i pesi dei cammini minimi che collegano il vertice 1 con tutti gli altri vertici. Più precisamente, si completi la seguente tabella:

1	2	3	4	5	6
0	∞	∞	∞	∞	∞
...					

dove ogni riga corrisponde a un'iterazione, e ogni casella contiene: per i nodi per i quali è già stata trovata la distanza definitiva, un simbolo speciale (per esempio `-`), per gli altri la distanza provvisoria corrente.

2. Si dica quale è il cammino minimo da 1 a 4, e come l'avete ricavato a partire dalla tabella.

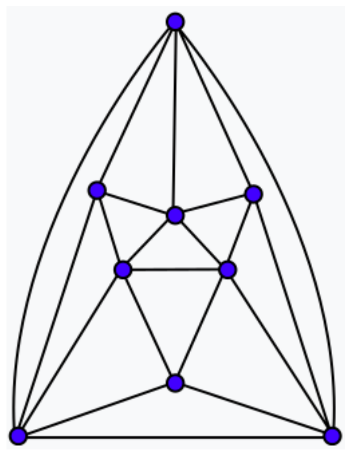
Soluzione

1. Nella prima colonna indichiamo il nodo estratto.

	1	2	3	4	5	6
	0	∞	∞	∞	∞	∞
1	-	7	1	∞	∞	∞
3	-	6	-	∞	3	8
5	-	5	-	8	-	8
2	-	-	-	8	-	6
6	-	-	-	8	-	-
4	-	-	-	-	-	-

2. Il cammino minimo da 1 a 4 è 1,3,5,4, ottenuto partendo da 4 e prendendo a ogni passo il nodo parent, ossia:
   - il parent di 4 è 5 perché la distanza definitiva 8 è stata assegnata con l'estrazione di 5
   - il parent di 5 è 3 perché la distanza definitiva 3 è stata assegnata con l'estrazione di 3
   - il parent di 3 è 1 perché la distanza definitiva 1 è stata assegnata con l'estrazione di 1

Esercizio 3 Supponi di applicare l'algoritmo *MCMinCut* al grafo in figura. Quante volte devi ripetere *MCMinCut* per ottenere il taglio minimo con probabilità almeno del 99%?



Guida alla correzione

1.1 almeno 5 a chi ha capito cosa si doveva fare; -2 per caso migliore guardando lo 0

1.2 almeno 5 a chi ha capito cosa si doveva fare; almeno 7 a chi dice il caso peggiore giusto (32)

1.3 almeno 5 a chi ha capito cosa si doveva fare

2.1 3 a chi ha fatto Prim

2.2 almeno 5 a chi ha capito cosa si doveva fare