Esercitazione 7: Dynamic Programming

Giacomo Paesani

May 8, 2024

Esercizio 1 (24.4-2, [1])**.** Sia $G = (V, E)$ un grafo orientato con pesi sugli archi, che possono essere anche negativi ma in cui non sono presenti cicli di peso negativo. Dimostrare che l'algoritmo di Dijkstra applicato su grafi di questo tipo nono calcola necessariamente i cammini di costo minimo tra la sorgente e gli altri vertici del grafo.

Esercizio 2 (24.3-6, [1])**.** Sia $G = (V, E)$ un grafo non diretto che rappresenta un network di comunicazioni. Ad ogni arco (u, v) viene associato un valore $r(u, v)$, che è un numero reale con $0 \leq r(u, v) \leq 1$ che rappresenta l'affidabilità dell'arco nel eseguire una comunicazione tra i vertici u e v. Interpretiamo $r(u, v)$ come la probabilità che la trasmissione tra u e v non fallisce e supponiamo che tali probabilità sono indipendenti tra loro. Fornire uno pseudo-codice che dato G e un vertice s restituisce l'albero dei cammini più sicuri da s.

Esercizio 3 (24.2-3, [1])**.** Dato un grafo orientato $G = (V, E)$ e con archi pesati da una funzione w, senza cicli orientati di peso negativo, e sia m il massimo del numero minimo di archi di un cammino minimo da un vertice s a v, per ogni vertice $v \in V$. Fornire uno pseudo-codice modificando l'algoritmo di *Bellman-Ford* in modo che vengono fatte al più $m + 1$ iterazioni, anche se m non è noto a priori.

Esercizio 4 (15.4-5:6, [1])**.** Fornire in pseudo-codice un algoritmo che data una sequenza finita di numeri interi X restituisce la lunghezza della più lunga sotto-sequenza strettamente crescente Y. Se, ad esempio, $X = (1, 3, 8, 5, 4, 2, 6, 0, 1, 2, 8, 9, 5)$ allora si ottiene $Y = (1, 3, 4, 6, 8, 9)$. Implementare questo algoritmo in modo che il tempo di esecuzione sia al più $\mathcal{O}(n^2)$ (ma si può fare anche in $\mathcal{O}(nlog(n))$). Come deve essere modificato l'algoritmo per far si che restituisca una sotto-sequenza strettamente crescente di lunghezza massima?

References

[1] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. Introduction to algorithms. 2022.