Prova scritta del 20 gennaio 2020 – Note sulla soluzione dell'esercizio 3

Questa documento NON è la soluzione dell'esercizio, ma contiene solo alcune indicazioni e osservazioni schematiche.

Il problema può essere risolto adattando l'*algoritmo di Dijkstra* per il calcolo delle lunghezze dei cammini minimi (distanze) in un grafo orientato da un vertice s (il centro). L'algoritmo può essere strutturato in 3 parti:

1. Calcolo delle distanza di ogni vertice da s mediante l'algoritmo di Dijkstra.
2. Calcolo del valore di r, come massimo valore delle distanze ottenute al punto precedente.
3. Identificazione dei vertici richiesti in output.

Il punto 2 può essere effettuato insieme punto 1, man mano si ottengono le distanze definitive.

Lo pseudocodice può essere presentato modificando quanto visto a lezione. In particolare si può utilizzare come punto di partenza il Codice 13.2 o il Codice 13.3, presentati negli appunti forniti sul sito web del corso, con opportune modifiche e integrazioni per i punti 2 e 3 (il Codice 13.2 è più semplice e schematico, il Codice 13.3 rende più facile presentare la stima del tempo di calcolo).

Il punto (c) richiede di indicare una possible rappresentazione del grafo che permetta di implementare efficientemente l'algoritmo. Come discusso a lezione, per implementare efficientemente l'algoritmo di Dijkstra è opportuno utilizzare liste di adiacenza o di incidenza. Queste risultano utili anche nella parte di identificazione dei vertici richiesti in output per controllare, da ogni vertice corrispondente alla fascia centrale o semicentrale, se tra gli archi uscenti ve ne è uno che raggiunge un vertice della fascia periferica. Con altre rappresentazioni, come le matrici di adiacenza o di incidenza, non si ottengono, per questo algoritmo, le stesse prestazioni.

Le code con priorità sono una struttura utile per implementare l'algoritmo, in particolare per realizzare la strategia greedy utilizzata dall'algoritmo di Dijkstra, ed è corretto menzionarle nella stima di complessità. Tuttavia, si tenga presente che le code con priorità *non sono una rappresentazione del grafo.*

In alternativa, ma con un inutile spreco di spazio e tempo, si poteva risolvere il problema adattando l'algoritmo di Floyd e Warshall. In questo caso è opportuno avere una rappresentazione del grafo mediante matrice dei pesi.