Esercitazione di laboratorio n. 12

La corretta risoluzione dell'esercizio 1 sarà oggetto di valutazione ai fini dell'attribuzione del punteggio per i laboratori.

Esercizio n. 1: rete di calcolatori

Una rete di calcolatori può essere rappresentata, in modo semplificato, mediante un grafo non orientato pesato con le seguenti caratteristiche:

- ogni nodo modella un elaboratore
- gli archi rappresentano le connessioni di rete punto-punto (cioè tra 2 nodi) e sono caratterizzati da una velocità massima di comunicazione (misurata in kbit/s).

La rete di calcolatori viene descritta su un file di testo, il cui nome compare come argomento alla riga di comando, dove le righe descrivono le connessioni e la loro velocità con il seguente formato:

<nodo1> <nodo2> <velocità>

<nodo1> e <nodo2> sono stringhe di al più 10 caratteri che identificano univocamente i nodi e <velocità> è un intero che rappresenta la velocità di comunicazione su quella connessione. Si noti che non è dato un elenco esplicito dei nodi, che vanno quindi ricavati dall'elenco delle connessioni. Il numero massimo di nodi è un argomento della riga di comando.

Dopo aver acquisito la descrizione della rete dal file in un'opportuna struttura dati interna, l'applicazione deve essere in grado di offrire ripetutamente all'utente un menù di possibili operazioni, tra cui:

1. verifica di connettività: occorre verificare che da qualsiasi elaboratore appartenente alla rete sia possibile raggiungere ogni altro elaboratore della rete stessa
2. *connectivity enforcement:* nel caso la rete non risulti connessa, l'utente può decidere di renderla tale, aggiungendo un insieme minimo di archi secondo i seguenti passi:
    - si ordinino in base alla loro cardinalità le componenti connesse
    - poi si proceda in catena a connettere quella di cardinalità massima con quella di cardinalità immediatamente inferiore, sino all'ottenimento di una sola componente connessa. L'arco aggiunto insista sui 2 nodi di grado massimo di ciascuna componente connessa. Il peso di detto arco sia specificato da tastiera dall'utente
3. rimozione di 1 elaboratore e di tutte le connessioni che lo coinvolgono
4. inserzione di nuove connessioni o rimozione di connessioni esistenti. Nel caso la nuova connessione riguardi elaboratori nuovi, essa sia preceduta dalla creazione dei nodi corrispondenti
5. in una rete connessa, identificazione delle connessioni critiche, cioè quelle che, se eliminate, ne comportano la disconnessione (bridge). Per ogni bridge, nel caso entrambe le componenti connesse che si otterrebbero rimuovendolo abbiano cardinalità > 1, si aggiunga un opportuno arco ridondante diverso dal bridge che le connetta con un cammino alternativo. Se una o entrambe le componenti hanno cardinalità pari a 1 non si faccia nulla. Per esempi di grafi con bridge e per l'algoritmo si vedano i lucidi 16 Le applicazioni degli algoritmi di visita pagg. 7-38
6. visualizzazione su video o stampa su file della rete in un formato a scelta.

Vincolo:
non è lecito assumere alcun ordinamento per i nodi e le connessioni riportate nel file.

Osservazione:
si noti che il grafo è dinamico, nel senso che, pur nel rispetto del numero massimo di nodi possibili, ad ogni istante la cardinalità degli insiemi vertici e archi può variare per effetto delle inserzioni e delle cancellazioni

Suggerimenti:

- per rappresentare il grafo come lista o come matrice delle adiacenze si allochino rispettivamente o un vettore o una matrice sovradimensionati in base al numero massimo di nodi possibili passato come argomento alla riga di comando
- all'interno dell'ADT grafo si mantenga un intero che rappresenta il numero di nodi attivi in quell'istante
- poiché il grafo è dinamico (sono possibili cancellazioni e inserzioni di nodi e di archi), per poter mantenere una struttura con vettore o matrice di vertici si suggerisce di non procedere alla rimozione dei vertici quando ne è richiesta la cancellazione, bensì di etichettarli come non attivi, seguendo la filosofia adottata nelle tabelle di hash con open addressing, dove la chiave cancellata non è rimossa, ma sostituita da un valore sentinella (cfr soluzione 1, pag. 38, lucidi 12 Le tabelle di hash)
- quando invece è richiesta la cancellazione di un arco, si suggerisce di rimuoverlo effettivamente dalla matrice delle adiacenze o dalla lista delle adiacenze, che dovrà essere implementata come lista linkata con l'operazione di cancellazione.

Esercizio ulteriore (non oggetto di valutazione, ma utile ai fini della preparazione all'esame)

Esercizio 2: calcolo dei cammini minimi di un grafo
Questo esercizio deve essere risolto sulla base di quanto già realizzato per il laboratorio n. 11.

È dato un file contenente un elenco di voli aerei, secondo il formato seguente:

- la prima riga contiene il numero totale `NC` delle città (aeroporti di partenza o arrivo dei voli)
- le righe successive contengono le città in ordine alfabetico
- segue l'elenco dei voli, in ragione di uno per riga, secondo il formato:

```
<partenza> <arrivo> <costo> <durata>
```

essendo `<partenza>`, `<arrivo>`, `<costo>` e `<durata>` la città di partenza, la città di arrivo, il costo in euro (intero) e la durata in ore (reale) del volo, rispettivamente.

I voli sono ordinati secondo città di partenza (in ordine alfabetico crescente). Inoltre, prima di ogni gruppo di voli in partenza dalla stessa città, il file contiene una riga indicante il numero totale di questi voli. Si supponga, infine, che i nomi delle città siano delle stringhe di lunghezza massima pari a 20 caratteri e non contengano spazi.

Un possibile esempio di contenuto del file è il seguente:

```
7
Francoforte
Londra
Milano
NewYork
Parigi
Roma
Torino
2
Londra Francoforte 220 1.5
Londra Milano 280 2.5
2
Milano Londra 250 2.2
Milano NewYork 950 12.4
```

```
2
NewYork Londra 820 8.2
NewYork Parigi 900 9.0
1
Parigi Milano 230 2.1
3
Roma Francoforte 300 2.8
Roma Londra 190 3.5
Roma Parigi 350 2.6
2
Torino Milano 150 0.5
Torino Roma 200 1.5
```

Si scriva un programma in linguaggio C che legga il file in ingresso e memorizzi le informazioni in esso contenute come grafo (si rappresenti il grafo tramite matrice di adiacenza, oppure tramite liste di adiacenza). Una volta memorizzato il grafo, il programma acquisisca da tastiera il nome di una città A di partenza e quello di una città B di arrivo e calcoli i percorsi consigliati tra esse. Nel dettaglio, il programma deve riportare a video le informazioni su:

- il percorso che minimizza il tempo di volo
- il percorso che minimizza il costo
- il percorso che minimizza il numero di scali.

Il programma deve fornire come output i dettagli di ogni tratta e le statistiche totali. Ad esempio:

```
Citta' di partenza: Torino
Citta' di arrivo: Londra

Cammino minimo per durata
-------------------------
Da Torino a Milano (prezzo: 150, durata: 0,5)
Da Milano a Londra (prezzo: 250, durata: 2,2)
*** TOTALE
Prezzo: 400
Durata: 2.7
Cambi: 1

Cammino minimo per prezzo
-------------------------
Da Torino a Roma (prezzo: 200, durata: 1,5)
Da Roma a Londra (prezzo: 190, durata: 3,5)
*** TOTALE
Prezzo: 390
Durata: 5.0
Cambi: 1

Cammino minimo per cambi
-------------------------
Da Torino a Milano (prezzo: 150, durata: 0,5)
Da Milano a Londra (prezzo: 250, durata: 2,2)
*** TOTALE
Prezzo: 400
Durata: 2.7
Cambi: 1
```

Suggerimento
Si gestiscano i nomi delle città mediante tabella di simboli (integrata o modularizzata) in modo che i vertici siano identificabili mediante interi.

Possibile variante
Si provi lo stesso esercizio nel caso in cui il file non riporti né il numero totale delle città, né l'elenco delle città, né il numero di voli in partenza da una stessa città e non sia previsto alcun ordinamento per i voli riportati.