

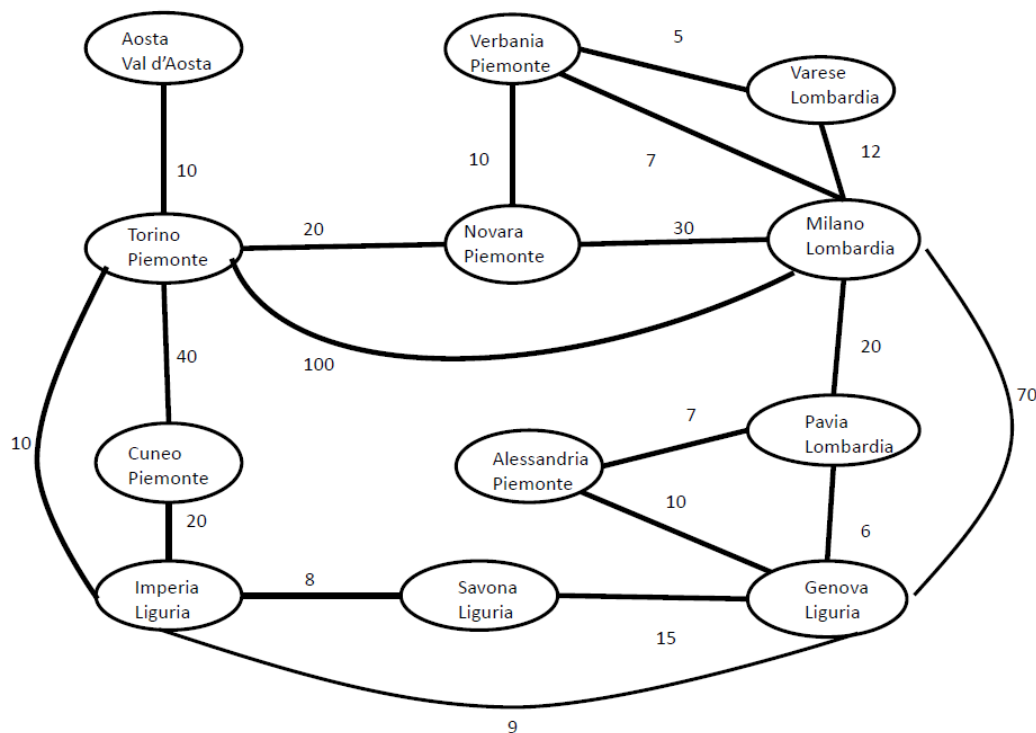


Esercitazione di laboratorio n. 12

(Caricamento nella Sezione Elaborati del Portale entro e non oltre il 13/01/2015)

Esercizio n. 1: Grafi - memorizzazione e gestione

Un grafo non orientato e pesato rappresenta una rete di città appartenenti ciascuna ad una regione. Il peso associato ad ogni arco rappresenta un flusso tra 2 città della stessa regione o di regioni diverse, come nell'esempio seguente:



Il grafo è contenuto in un file, il cui nome è passato come argomento sulla linea di comando di un programma C. Il file ha il seguente formato:

- sulla prima riga un unico intero N rappresenta il numero di vertici del grafo
- seguono N righe ciascuna delle quali contiene una coppia di stringhe di al massimo 10 caratteri, la prima per il nome della città, la seconda per quello della regione
- sulle righe successive, in numero indefinito, si trovano coppie di città ciascuna rappresentata da una stringa di lunghezza massima pari a 10 caratteri. Ogni coppia è un arco del grafo. Vi è inoltre, sempre sulla stessa riga, un intero che rappresenta il flusso tra le 2 città, cioè il peso dell'arco.

Si facciano anche le seguenti assunzioni:

- i nomi dei singoli nodi sono univoci all'interno del grafo
- il numero di nomi distinti di vertici incontrati leggendo gli archi è coerente col numero N
- non sono ammessi cappi
- tra due nodi c'è al massimo un arco (non è un multigrafo).



Si consideri come esempio (parziale) del grafo di figura il seguente file di testo:

```
12
Aosta ValdAosta
Torino Piemonte
---
Genova Liguria
Aosta Torino 10
Torino Milano 100
Genova Imperia 9
---
Verbania Varese 5
```

Si scriva un programma in C in grado di caricare in memoria il grafo (non orientato e pesato), leggendone i contenuti da file e di potervi effettuare alcune semplici operazioni.

La rappresentazione della struttura dati in memoria deve essere fatta tenendo conto dei seguenti vincoli:

- il grafo sia implementato come ADT di I categoria, predisposto in modo tale da poter contenere sia la matrice che le liste di adiacenza (nella fase di caricamento dei dati da file si generino le sole liste di adiacenza, su comando esplicito va generata anche la matrice di adiacenza)
- le informazioni relative alle città e alle regioni siano memorizzate in un vettore di Item, dove ciascun item contiene la stringa del nome della città e del nome della regione (max 10 caratteri)
- si utilizzi una tabella di simboli (ADT di I categoria) per associare gli interi che rappresentano i nomi dei vertici del grafo ai nomi delle città: tale tabella di simboli viene utilizzata per la sola conversione efficiente nome->intero, la conversione intero->nome viene fatta mediante accesso diretto al vettore di Item.

Sul grafo, una volta acquisito da file, sia possibile:

- calcolare e visualizzare il grado di ogni nodo
- stampare il numero totale di vertici ed elencarli esplicitamente per nome della città
- stampare il numero di archi incidenti su un nodo e l'elenco di vertici ad esso connessi
- generare la matrice di adiacenza, **SENZA** leggere nuovamente il file, a partire dalle liste di adiacenza
- determinare l'ammontare complessivo dei flussi intraregionali ed interregionali globali di tutte le regioni. Nell'esempio il flusso intraregionale è pari a 134 e quello interregionale 275. Si noti che per effettuare questo calcolo è sufficiente percorrere iterativamente tutti gli archi del grafo.

Esercizio n. 2: Code a priorità - implementazione a Heap

Un ufficio postale ha deciso di adottare un sistema a priorità per la gestione dei clienti in coda. Tutti i clienti sono inseriti in un'unica coda e ad ognuno di essi è associata una priorità sulla base dell'operazione richiesta.

Si scriva un programma in C in grado di simulare la situazione appena descritta sfruttando un ADT di I categoria "coda a priorità" implementata ad Heap.

I clienti siano descritti a loro volta da un ADT di I categoria "cliente" caratterizzato da un codice alfanumerico di 5 caratteri e un valore intero non negativo a descriverne la priorità.

Si organizzi il programma su tre moduli distinti:

- ADT di I categoria Coda a priorità
- ADT di I categoria Cliente



- main di gestione dell'ufficio postale.

Dal main, si rendano possibili le seguenti operazioni:

- aggiunta di un nuovo cliente in coda leggendone i dati da tastiera
- segnalazione del completamento delle operazioni associate a un cliente e conseguente estrazione di un nuovo cliente dalla coda.
- lettura e inserimento in coda di clienti i cui dati sono memorizzati su un file di testo, il cui nome sia richiesto all'utente. Nel file non è indicato il numero di clienti presenti, che sono riportati in ragione di uno per riga, da coppie del tipo <codice> <priorità>.

Si consideri come esempio il seguente file di testo:

```
CLI01 12  
CLI02 6  
CLI03 23  
CLI04 1
```