



Esercitazione di laboratorio n. 7

(Caricamento sul portale entro le 23.59 del 29/12/2017 di entrambi gli esercizi)

Esercizio n. 1: Via Emilia

(Rivisitazione del) Tema d'esame del 26/01/2016 – Traccia da 18 punti

N città sorgono lungo la via Emilia. Ogni città è caratterizzata da un nome, dal numero di abitanti e dalla sua distanza rispetto all'inizio della via. La distanza tra due città è data dalla differenza, in valore assoluto, tra le relative distanze dall'inizio della via. Le informazioni sulle città sono lette da un file, il cui nome è passato come parametro sulla riga di comando. Il file ha il seguente formato:

- la prima riga contiene N (numero intero)
- in ognuna delle N righe successive compaiono una stringa di al massimo 20 caratteri per il nome della città, un intero per la popolazione (in migliaia) e un intero per la distanza rispetto all'inizio della strada.

Esempio: contenuto del file

```
11
Piacenza 102 0
Fidenza 27 35
Parma 190 57
ReggioEmilia 171 83
Modena 185 115
Bologna 386 144
Imola 69 177
Faenza 58 193
Forlì 118 207
Cesena 97 225
Rimini 147 253
```

Si vuole pianificare in quali città localizzare K < N Autorità di Ambito Territoriale Ottimale (ATO). K è passato come parametro sulla riga di comando. Il sottoinsieme di K città va scelto in modo tale da minimizzare la distanza media, per ogni abitante, dalla sede di Autorità ATO più vicina. Questo risultato si può ottenere minimizzando la sommatoria *SomDist* delle distanze di ogni abitante dall'Autorità ATO più vicina:

$$\text{SomDist} = \sum_{i=0}^{\text{num.città}-1} \text{popolazione}_i * \text{distMinDaATO}_i$$

Nella formula precedente *popolazione_i* indica la popolazione dell'i-esima città, mentre *distMinDaATO_i* ne indica la distanza dalla sede di Autorità ATO più vicina.

Si scriva in C un programma che, ricevuti come argomenti al *main* in nome del file e K:

1. legga il file e generi una opportuna struttura dati
2. calcoli la mutua distanza tra ogni città e tutte le altre, memorizzando tali informazioni in una struttura dati appropriata. La complessità dell'algoritmo deve essere $O(N^2)$
3. calcoli, mediante un algoritmo ricorsivo, la soluzione ottima richiesta (un sottoinsieme di K città in cui collocare le sedi ATO, in modo da minimizzare *SomDist*).

Per determinare *SomDist* si realizzi una opportuna funzione *SommaDistanze*, la quale, a partire dal vettore delle popolazioni e da un insieme di città selezionate come sedi dell'Autorità ATO, calcoli



il relativo valore `SomDist`. Si noti che è possibile realizzare `SommaDistanze` con un algoritmo $O(N)$.

Suggerimento: si identifichi prima il modello di Calcolo Combinatorio utilizzato, poi si realizzi la funzione di controllo di ottimalità.

Esercizio n. 2: Atleti (piano di allenamenti)

Si consideri lo scenario introdotto nell'esercizio n.3 del laboratorio 6 (Atleti a liste).

Si estenda il codice già scritto per supportare la memorizzazione del piano di allenamenti settimanale, composto da un dato insieme di esercizi, assegnato ad ogni atleta.

Le informazioni relative agli esercizi disponibili sono acquisite da un file (`esercizi.txt`) nella cui prima riga si trova il numero `E` di esercizi presenti. Sulle `E` righe seguenti è riportata una terna `<nome_es>` `<categoria_es>` `<tipologia_es>` in ragione di una per riga. Tutti gli elementi sono stringhe senza spazi, di massimo 50 caratteri.

Il piano di allenamenti per ogni atleta viene acquisito da un file (uno per ogni atleta) il cui nome coincide col codice dell'atleta (ad esempio `A0001.txt` per l'atleta `A0001`), contenente un numero non noto di terne, una per riga, riportanti `<nome_es>` `<num_set>` `<num_ripetizioni>`, dove `<num_set>` e `<num_ripetizioni>` rappresentano il numero di set e di ripetizioni per set dell'esercizio di nome `<nome_es>` (ad esempio di un dato esercizio si potrebbero fare 4 set ciascuno di 10 ripetizioni). Il nome dell'esercizio è la chiave che permette di associare un esercizio di un piano a uno degli esercizi in `esercizi.txt`.

Si scriva un programma che permetta di:

- caricare in un vettore di strutture, allocato dinamicamente, l'elenco di esercizi possibili/disponibili
- chiedere iterativamente all'utente di quale atleta voglia caricare il piano di allenamenti, oppure salvare su file il piano di allenamenti (se disponibile)
- modificare il numero di set e/o ripetizioni di un dato esercizio nel piano di allenamenti di un dato atleta (vanno acquisiti da tastiera il nome dell'atleta, l'esercizio e i nuovi numeri di set/ripetizioni).

Ai fini dell'esercizio, si imposti la struttura dati in modo che sia coerente con la rappresentazione grafica proposta in figura (i nomi dei tipi e dei campi sono solo a titolo di esempio).

