

Treni

Un agente segreto deve fuggire dopo aver sottratto alcuni documenti da un laboratorio segreto. Si reca quindi in incognito alla stazione ferroviaria della città più vicina e consulta la mappa con i collegamenti ferroviari. Tale mappa riporta, per ogni città del paese l'elenco delle città raggiungibili. Si supponga che, per ogni tratta del collegamento ferroviario tra due città, siano indicati sia il numero di Km che separano le due città che il costo della tratta.

L'agente ha in tasca W Euro, e cerca sulla mappa qual è l'itinerario che gli permette di compiere più strada con tale cifra in modo da seminare i suoi inseguitori.

Si definisca la struttura dati che meglio si presta a memorizzare le informazioni relative alla mappa dei collegamenti ferroviari, e si codifichi in linguaggio C una procedura che permetta di calcolare qual è l'itinerario di viaggio che permetta di compiere il maggior numero di Km con la cifra di W Euro che l'agente ha a disposizione.

Si supponga che le informazioni relative alla mappa dei collegamenti ferroviari sia memorizzata in un file che abbia il seguente formato:

```
Numero di città
Nome città 1
Nome città 2
.
.
.
*Città 1
Numero di città raggiunte
Città raggiunta tramite la tratta 1      costo tratta      Km tratta
Città raggiunta tramite la tratta 2      costo tratta      Km tratta
*Città 2
Numero di città raggiunte
Città raggiunta tramite la tratta 1      costo tratta      Km tratta
Città raggiunta tramite la tratta 2      costo tratta      Km tratta
.
.
.
```

Non è richiesta la scrittura della procedura per effettuare la lettura della mappa da file.

SOLUZIONE

La struttura dati che meglio si presta a memorizzare le informazioni date è un grafo non orientato e pesato.

I nodi del grafo sono le città del paese considerato e descritte nel file dati. Gli archi del grafo sono invece le tratte ferroviarie che connettono le città. Infine, il peso associato agli archi è un insieme composto da due informazioni: il costo della tratta e la lunghezza della tratta espressa come numero di Km

Il grafo è non orientato in quanto si suppone che i collegamenti ferroviari tra due città possano essere percorsi in entrambe i sensi di percorrenza.

Il grafo è rappresentato con una matrice di adiacenza il cui tipo è il seguente:

```
typedef struct T {  
    int  costo;  
    int  Km;  
} TRAINTABLE;
```

La matrice si suppone allocata dinamicamente (per i dettagli si veda il codice C).

I nomi delle città per cui sono noti i collegamenti ferroviari sono memorizzate in una vettore di stringhe costruito durante la lettura del file dati. Tale vettore è utilizzato sia durante la lettura da file della matrice di adiacenza per passare dal nome della città alla sua posizione nella matrice di adiacenza, che durante la stampa della soluzione ottima, in modo da elencare i nomi delle città visitate.

Come soluzione alternativa per la memorizzazione dei nomi delle città potrebbe essere utilizzata una tabella di Hash. In tal modo si eviterebbe una ricerca lineare ogniqualvolta fosse necessario passare dal nome di una città alla sua posizione all'interno della matrice di adiacenza.

La ricerca della soluzione ottima avviene attraverso una procedura ricorsiva che genera tutte le possibili combinazioni per cui esiste almeno una tratta ferroviaria che connetta gli elementi della soluzione, ed per cui il costo complessivo del viaggio sia minore o al più uguale alla cifra disponibile.

La procedura ricorsiva termina non appena si è esaurita la somma disponibile, oppure non è più possibile aggiungere nuove destinazioni alla soluzione corrente.

Per ogni soluzioni trovata dalla procedura, si controlla se la nuova soluzione migliora la soluzione ottima, ed in caso affermativo si aggiorna la soluzione ottima.