

01JKE APA I - 01JKF APA II

Appello del 01/02/2010 – Prova di Programmazione (18 punti)

Un “solitario” viene giocato su un grafo orientato pesato, nel quale il giocatore, partendo da un nodo iniziale, deve effettuare un percorso con l’obiettivo di accumulare il massimo numero di punti (numeri interi), secondo le regole sotto elencate.

- I nodi del grafo sono identificati da una stringa univoca (si utilizzeranno, come esempio, nomi di città).
- Il giocatore, con punteggio (*score*) iniziale pari a 0, percorre il grafo, incrementando o decrementando il punteggio nel passaggio su nodi e archi. Il giocatore non può in nessun momento avere un punteggio negativo.
- I nodi possono assegnare dei premi, caratterizzati ognuno da una classe (si utilizzeranno, quale esempio, nomi di colori) e da un punteggio. Lo stesso vertice può contenere nessuno, uno o più premi (comunque, al massimo uno per categoria). Il giocatore, passando da un nodo con premi, ne conquista uno (a scelta), sommando il relativo punteggio al suo score. Il premio assegnato viene quindi rimosso dal nodo.
- È possibile un solo passaggio in un nodo senza premi, mentre sono possibili più passaggi in un nodo con premi (al massimo, un passaggio per ogni premio raccolto, più un ultimo passaggio sul nodo senza premi).
- Ogni arco è caratterizzato da un peso, che ne indica il costo/guadagno in termini di punti. Un giocatore, percorrendo l’arco, ne somma il peso al suo score (un peso negativo diminuisce quindi il punteggio).
- Il giocatore ha a disposizione un numero massimo di mosse pari a MAX (costante predefinita). Il gioco termina, quindi, per una delle condizioni seguenti:
 - il giocatore decide (arbitrariamente) di fermarsi.
 - si raggiunge il numero massimo di mosse.
 - non ci sono mosse disponibili a partire dal nodo raggiunto, in quanto non ci sono archi uscenti, oppure qualunque arco percorribile causerebbe uno score negativo, oppure tutti i nodi raggiungibili dagli archi uscenti non sono più disponibili (già raggiunti una volta senza premio).
- Se il giocatore ha accumulato premi di tutte le categorie disponibili, lo score finale viene raddoppiato.

Si scriva un programma C che legga la descrizione del grafo da un file testo, secondo il formato seguente.

- La prima riga del file contiene tre interi separati da uno spazio: il numero V di vertici e il numero E di archi del grafo, più il numero C di categorie di premi disponibili.
- Le V righe successive riportano i nomi (stringhe prive di spazi e di al più 20 caratteri) dei nodi, uno per riga.
- Le E righe successive riportano l’elenco degli archi (orientati e pesati), uno per riga, caratterizzati dai nomi dei due nodi connessi e dal peso dell’arco (un intero con segno).
- La parte restante del file contiene l’elenco dei premi, uno per riga, ognuno identificato dal nome del nodo, categoria (stringa priva di spazi e di al più 20 caratteri) e ammontare del premio (intero positivo).

Non è lecito assumere alcun ordine particolare per i vertici e gli archi nel file di ingresso.

Letto tale file, il programma deve proporre (mediante scelta a menu) le seguenti funzionalità:

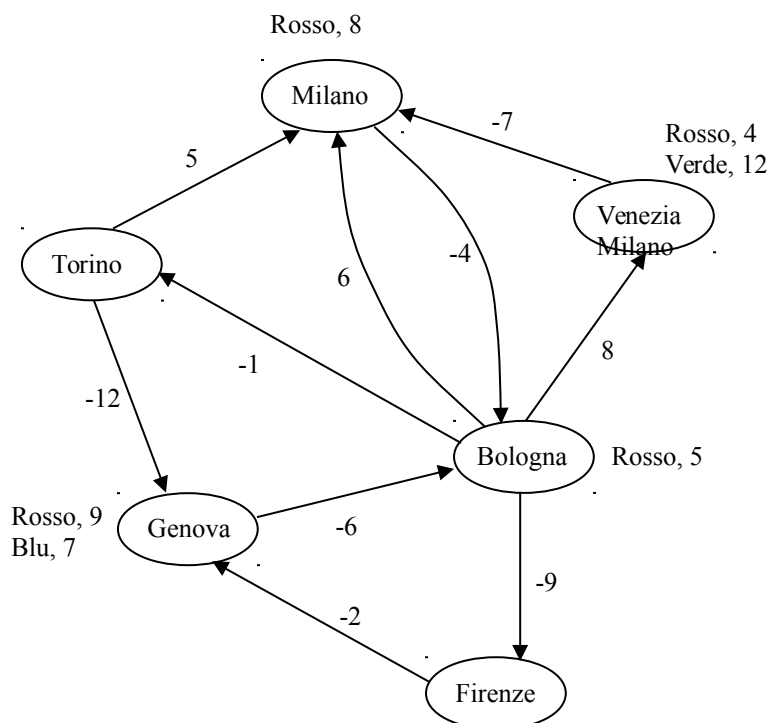
1. Stampare l’elenco degli archi in ordine alfabetico, ovvero seguendo l’ordine alfabetico dei nodi da cui partono e, a parità di questo, l’ordine alfabetico dei nodi su cui arrivano.
2. Partendo da vertice v (inserito dall’utente), verificare l’esistenza di almeno un cammino semplice (ovvero, senza nodi ripetuti) che, SENZA CONSIDERARE i punteggi ed i relativi vincoli, riesca a raggiungere nodi contenenti, complessivamente, premi di tutte le categorie disponibili (come caso limite, andrebbe anche bene raggiungere un solo nodo contenente premi di tutte le categorie). In caso affermativo, occorre visualizzare almeno uno di questi cammini.
3. Cercare il cammino ottimo, cioè il percorso che, partendo da vertice v (inserito dall’utente) e rispettando tutte le regole sopra elencate, permetta di ottenere il più alto score possibile. Si noti che, in questo caso, è sufficiente visualizzare solo il valore del punteggio ottimo, non il cammino corrispondente.

- Indicare nell'elaborato e nella relazione (oltre a nome, cognome e numero di matricola) anche il nome del corso per cui si sta sostenendo l'esame (ad esempio, APA 5 crediti, APA 10 crediti, PAC, EDP, ...).
- È consentito utilizzare chiamate a funzioni standard, quali ordinamento per vettori, inserzione/estrazione/ricerca relative a FIFO, LIFO, liste, BST, tabelle di hash e altre strutture dati, considerate come librerie esterne. Gli header file delle librerie utilizzate devono essere allegati all'elaborato. Le funzioni richiamate, inoltre, dovranno essere incluse nella versione del programma allegata alla relazione.
- Consegna delle relazioni (per tutti i corsi): entro giovedì 04/02/2010, alle ore 14:00, via e-mail all'indirizzo: sergio.nocco@polito.it, usando come subject (oggetto) la stringa APA#<m>, essendo <m> il proprio numero di matricola. L'allegato alla mail deve essere costituito da un unico file: un archivio compresso, contenente sia il codice corretto, sia la relazione (**NO eseguibili**). **QUALORA IL CODICE SPEDITO CON LA RELAZIONE NON COMPILI CORRETTAMENTE, VERRÀ APPLICATA UNA PENALIZZAZIONE.** Si ricorda che la valutazione del compito viene fatta senza discussione o esame orale, sulla base dell'elaborato svolto in aula.

Esempio

Sia il seguente il contenuto del file di ingresso, corrispondente al grafo visualizzato sulla destra.

```
6 10 3
Torino
Milano
Genova
Bologna
Venezia
Firenze
Torino Milano 5
Bologna Torino -1
Bologna Milano 6
Milano Bologna -4
Venezia Milano -7
Bologna Venezia 8
Genova Bologna -6
Firenze Genova -2
Bologna Firenze -9
Torino Genova -12
Genova Rosso 9
Venezia Verde 12
Milano Rosso 8
Genova Blu 7
Venezia Rosso 4
Bologna Rosso 5
```



Allora, il programma deve:

- visualizzare l'elenco degli archi nel seguente ordine:

```
Bologna Firenze
Bologna Milano
Bologna Torino
Bologna Venezia
Firenze Genova
Genova Bologna
Milano Bologna
Torino Genova
Torino Milano
Venezia Milano
```

- supponendo di partire dal vertice Torino, indicare che un possibile cammino semplice che permette di raggiungere nodi contenenti tutti e tre i tipi di premi è: Torino - Genova - Bologna - Venezia

- ipotizzando che il valore della costante MAX sia pari a 5 (e sempre supponendo di partire dal vertice Torino), riportare che il punteggio massimo ottenibile è pari a 36 (ottenuto seguendo il cammino Torino - Milano - Bologna - Milano - Bologna - Venezia, da cui: $0+5+8-4+5+6+0-4+0+8+12=36$). Se invece la costante MAX fosse uguale a 6, allora il punteggio ottimo sarebbe pari a 48 (dal cammino Torino - Milano - Bologna - Firenze - Genova - Bologna - Venezia, da cui: $0+5+8-4+5-9+0-2+7-6+0+8+12=24$, che però va raddoppiato avendo preso premi di tutte le categorie).

Si noti che un eventuale cammino Torino - Genova - ... non è lecito, in quanto farebbe diventare negativo lo score del giocatore. Analogamente, anche un cammino tipo Torino - Milano - Bologna - Torino - ... non è lecito, in quanto, non assegnando premi, è possibile toccare il vertice Torino solo una volta.