

Algoritmica – Prova di Laboratorio

Corso A e B

Appello del 04/06/2015

Istruzioni

Risolvete il seguente esercizio prestando particolare attenzione alla formattazione dell'input e dell'output. La correzione avverrà in maniera automatica eseguendo dei test e confrontando l'output prodotto dalla vostra soluzione con l'output atteso. Si ricorda che è possibile verificare la correttezza del vostro programma su un sottoinsieme dei input/output utilizzati. I file di input e output per i test sono nominati secondo lo schema: `input0.txt output0.txt input1.txt output1.txt ...`. Per effettuare le vostre prove potete utilizzare il comando del terminale per la redirectione dell'input. Ad esempio

```
./compilato < input0.txt
```

effettua il test del vostro codice sui dati contenuti nel primo file di input, assumendo che `compilato` contenga la compilazione della vostra soluzione e che si trovi nella vostra home directory. Dovete aspettarvi che l'output coincida con quello contenuto nel file `output0.txt`. Per effettuare un controllo automatico sul primo file input `input0.txt` potete eseguire la sequenza di comandi

```
./compilato < input0.txt | diff - output0.txt
```

Questa esegue la vostra soluzione e controlla le differenze fra l'output prodotto e quello corretto.

Una volta consegnata, la vostra soluzione verrà valutata nel server di consegna utilizzando altri file di test non accessibili. Si ricorda di avvisare i docenti una volta che il server ha accettato una soluzione come corretta.

Esercizio

È martedì sera, *winter is coming*: pizza, laptop e Game of Thrones con amici. Tuttavia, l'entusiasmo per la serie TV non è eguagliata dalla vostra memoria e fate fatica a ricordarvi le gesta dei vari personaggi. Il vostro compito è di scrivere un programma che vi aiuti a tener traccia dei vari personaggi mantenendo un database personaggio → lista delle sue vittime.

Il programma deve leggere due sequenze di dati ed una interrogazione.

La prima sequenza consiste in un intero M ed M coppie (*personaggio*, *ID*). I nomi dei personaggi sono stringhe, mentre gli ID sono interi compresi tra 0 e $M - 1$. Entrambi sono univoci, cioè occorrono una volta sola nella sequenza. Le coppie vanno inserite in un albero binario di ricerca **non** bilanciato, in cui i nomi delle vittime sono le chiavi e gli ID i valori associati.

Ad ogni personaggio è associata una lista dei personaggi uccisi. Al termine della lettura della prima sequenza deve essere allocata una tabella T di dimensione M , letto nella sequenza precedente. All'entrata $T[k]$ della tabella deve essere associata la lista dei personaggi uccisi dal personaggio con id k .

La seconda sequenza consiste in un intero N ed N coppie (*personaggio*, *vittima*), con cui riempire la tabella descritta precedentemente. In particolare, dopo aver letto una coppia, bisogna ottenere l'ID x del personaggio tramite una ricerca nell'albero binario di ricerca ed aggiungere la vittima alla lista delle sue vittime inserendola *in testa* alla lista $T[x]$.

Infine, l'interrogazione consiste nel nome di uno dei personaggi, a cui bisogna rispondere stampando la lista delle sue vittime nell'ordine della lista (quindi scorrendo la lista da sinistra verso destra).

L'**input** è formattato nel seguente modo. La prima riga contiene l'intero M . Seguono poi $2M$ righe, due righe per coppia (*personaggio*, *ID*). La prima riga della coppia contiene il nome del personaggio (stringa), mentre la seconda contiene l'ID (intero compreso tra 0 ed $M - 1$). Segue poi un intero N , e a seguire $2N$ righe, due righe per coppia (*personaggio*, *vittima*). La prima riga della coppia contiene il nome del personaggio (stringa), mentre la seconda contiene il nome della vittima (stringa). L'ultima riga dell'input contiene il nome del personaggio X di cui bisogna stampare le vittime (stringa). Tutte le stringhe che fanno parte dell'input sono lunghe al più **19** caratteri, senza spazi (ad esempio "RobertBaratheon").

L'**output** è costituito dai nomi dei personaggi uccisi dal personaggio X , una riga per vittima, nello stesso ordine della lista.

Esempio

Nota 1: il testo in verde è da intendersi come *commento* e dunque non fa parte dell'input.

Nota 2: per evitare *spoiler*, gli accoppiamenti tra personaggi e vittime nell'esempio sono generati in modo casuale.

Input

```
4 # Coppie (personaggio, ID)
EddardStark
2
KhalDrogo
3
TyrionLannister
0
JoffreyBaratheon
1
5 # Coppie (personaggio, vittima)
KhalDrogo
RamsayBolton
JoffreyBaratheon
EddardStark
TyrionLannister
JoffreyBaratheon
KhalDrogo
CerseiLannister
KhalDrogo
PetyrBaelish
KhalDrogo # Interrogazione
```

Albero (Personaggio, ID)

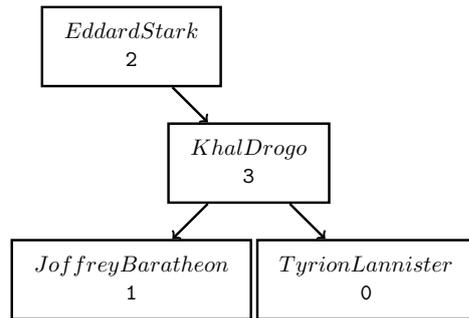


Tabella ID → Vittime

0	→	(JoffreyBaratheon)
1	→	(EddardStark)
2	→	()
3	→	(PetyrBaelish, CerseiLannister, RamsayBolton)

Output

```
PetyrBaelish
CerseiLannister
RamsayBolton
```