

Algoritmica – Esame di Laboratorio

06/11/2014

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

Alice e Bob seguono fedelmente la filosofia “*non hai bisogno di più di quindici oggetti diversi per vivere*”, che pone due vincoli al possesso dei beni:

- si devono possedere al più *quindici* oggetti;
- si può possedere al più *un* oggetto per ogni tipologia.

Ad esempio, un individuo non può possedere più di un televisore, ma può possedere un televisore, un frigorifero, un telefono, etc. fino ad un massimo di quindici oggetti.

Dopo qualche anno, Alice e Bob decidono finalmente di sposarsi. La filosofia prevede però che una coppia sposata debba essere vista come un unico soggetto. Dunque Alice e Bob, che complessivamente potevano possedere fino a quindici oggetti a testa (trenta in totale) prima di sposarsi, devono rinunciare ad alcuni oggetti al fine di possedere al più quindici oggetti distinti in totale. Al fine di decidere quali oggetti tenere, Alice e Bob decidono di associare ad ogni oggetto un “valore affettivo” e di scegliere dunque i quindici oggetti (al più) che massimizzano la somma del valore affettivo. Il vostro compito è quello di aiutare Alice e Bob nella scelta, implementando la strategia seguente.

Vi verrà data in input la lista di oggetti posseduti da Alice e da Bob. Ogni oggetto avrà un *nome* (una stringa), che ne contraddistingue la tipologia, ed un *valore affettivo* (un intero). Gli oggetti dovranno essere inseriti in una tabella hash a liste di trabocco, al fine di scoprire i duplicati e di selezionare, a parità di tipologia, l’oggetto col valore affettivo maggiore. In particolare, la tabella dovrà avere $2N$ ingressi, dove N è la somma degli oggetti posseduti complessivamente da Alice e Bob. La funzione hash riceve in input una stringa non vuota (il nome dell’oggetto) e restituisce in output un valore numerico:

- si esegue la somma di tutti i codici decimali associati ai caratteri della stringa (escluso il carattere speciale di fine stringa ‘\0’); ogni carattere **deve** essere trattato come **unsigned int**;
- l’output è pari alla somma precedentemente calcolata modulo $2N$.

Ad esempio, per la stringa “Ciao!” ($N = 5$) si hanno i codici indicati nella seguente tabella:

Carattere	Codice decimale
C	67
i	105
a	97
o	111
!	33

Si ottiene: $67 + 105 + 97 + 111 + 33 = 413$, $2N = 10$ e dunque $\text{hash}(\text{"Ciao!"}) = 413 \% 10 = 3$.

Per ogni oggetto in input, bisognerà aggiornare la tabella hash come segue. Supponiamo che X sia l'oggetto con cui dovrà essere aggiornata la tabella hash, ed indichiamo con $\text{nome}(X)$ il suo nome e con $\text{valore}(X)$ il suo valore affettivo. Ogni oggetto in input dovrà essere cercato nella tabella hash. Vi sono tre casi:

1. la tabella *non* contiene un oggetto Y per cui $\text{nome}(X) = \text{nome}(Y)$.
In questo caso, bisognerà inserire l'oggetto X *in cima* alla lista di trabocco;
2. la tabella contiene un oggetto Y per cui $\text{nome}(X) = \text{nome}(Y)$, e per cui $\text{valore}(X) \leq \text{valore}(Y)$. In questo caso non bisogna far nulla;
3. la tabella contiene un oggetto Y per cui $\text{nome}(X) = \text{nome}(Y)$, e per cui $\text{valore}(X) > \text{valore}(Y)$. In tal caso, bisognerà aggiornare Y impostando $\text{valore}(Y) = \text{valore}(X)$.

Nota come la regola d'aggiornamento garantisca che, ad ogni passo, vi sia nella tabella al più un oggetto per tipologia e che, per ogni tipologia, sia stato selezionato l'elemento col più alto valore affettivo. Una volta riempita la tabella hash, bisognerà ordinarne gli oggetti per valore affettivo decrescente e, a parità di valore affettivo, per ordine lessicografico crescente. Infine, stampare il nome dei primi quindici oggetti ordinati.

Nota: il numero finale degli oggetti potrà essere inferiore a quindici: in tal caso stampare tutti gli oggetti nella tabella hash, sempre in base all'ordinamento richiesto.

L'input è così formato:

- La prima linea contiene il numero N di oggetti da processare;
- Le successive $2N$ linee contengono la descrizione degli N oggetti; la descrizione di ogni oggetto occupa 2 linee consecutive:
 - la prima contiene il nome (una stringa, non vuota, di lunghezza non superiore a 100 caratteri);
 - la seconda contiene il valore affettivo (un intero).

L'output è composto dai nomi degli oggetti selezionati con la strategia esposta precedentemente, uno per linea, ordinati per valore affettivo decrescente e, a parità di valore affettivo, per ordine lessicografico crescente.

Esempio

Input

21
Frigorifero
30
Frigorifero
60
Telefono
50
Asciugamano
10
Libro
5
Libro
100
Tovaglia
70
Computer
120
Scarpe
160
Ventilatore
1
Bilancia
1
Cestino
1
Tazza
1
Cuscino
1
Trolley
1
Appendiabiti
1
Termometro
1
Penna
1
Matita
1
Orologio
1
Abat-jour
1

Output

Scarpe
Computer
Libro
Tovaglia
Frigorifero
Telefono
Asciugamano
Abat-jour
Appendiabiti
Bilancia
Cestino
Cuscino
Matita
Orologio
Penna