

Algoritmi e Strutture Dati – Prova di Laboratorio

29/06/2017

Istruzioni

Risolvere il seguente esercizio prestando particolare attenzione alla formattazione dell'input e dell'output. La lettura dell'input e la scrittura dell'output **DEVONO** essere effettuate tramite gli stream **cin** e **cout** rispettivamente. 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.

Entro la **prima ora** è possibile richiedere **una sola volta** ai docenti la valutazione della complessità della propria soluzione. Tale richiesta può essere effettuata solo qualora la valutazione sul server sia stata **positiva**.

Esercizio

Si consideri un sistema di memorizzazione che legga una sequenza di N interi non negativi e li inserisca dentro una particolare *tabella hash*, in cui a ciascun indirizzo è associato un albero binario di ricerca (ABR). Ciascun ABR mantiene una sola entrata per ogni valore, tenendo traccia del numero di duplicati per ciascuno di essi. Dato un ABR, si definisce D come l'etichetta del nodo con più duplicati, e a parità di questi ultimi si considera il valore di etichetta più alto. Nel caso l'albero sia vuoto, $D = -1$.

Scrivere un programma che legga da tastiera una sequenza di N interi x e per ciascuno di essi

- individui l'indirizzo corrispondente utilizzando la seguente funzione hash:

$$h(x) = \{[(a \times x) + b] \% p\} \% S$$

dove $p=999149$, $a=1000$ e $b=2000$;

- lo inserisca nell'ABR associato all'indirizzo calcolato al punto precedente.

Il programma dovrà poi stampare al più i primi K valori di D non negativi, ordinati in maniera decrescente.

L'**input** è formattato nel seguente modo: la prima riga contiene gli interi N , K e S separati da uno spazio. Seguono N righe contenenti un intero ciascuna.

L'**output** contiene gli elementi della soluzione, uno per riga.

Esempio

Input

```
6 2 3
9
17
16
7
6
7
```

Output

```
17
9
```