

Algoritmi e Strutture Dati – Prova di Laboratorio

20/07/2016

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 per la gestione di alberi binari di ricerca (ABR) in grado di memorizzare nodi ad etichette intere.

Dato un nodo x :

- si definisce d_x come la distanza tra x e la radice;
- si definisce l_x come la massima distanza tra x e le foglie contenute nel sottoalbero radicato in esso;
- x si dice **mediano** se i valori di l_x e d_x differiscono al più di uno.

Si scriva un programma che

- legga da tastiera N etichette e le inserisca all'interno dell'ABR. I valori devono essere inseriti nello stesso ordine con cui vengono letti (le etichette \leq vanno inserite a sinistra);
- verifichi per ogni nodo che sia mediano o meno; (complessità al più $\mathcal{O}(n)$)
- stampi le etichette dei primi K nodi mediani ordinati per etichetta in maniera non decrescente. Nel caso i valori fossero meno di K , stampare quelli disponibili. (complessità al più $\mathcal{O}(n)$)

NON è permesso utilizzare strutture dati (array/vector) di appoggio.

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

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

Esempio

Input

6 2
10
5
20
15
30
25

Output

5
20

