

# Algoritmi e Strutture Dati – Prova di Laboratorio

29/06/2016

## 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

[leggere il testo prestando particolare attenzione alle definizioni]

Si consideri un sistema per la gestione di alberi binari di ricerca (ABR) in grado di memorizzare nodi ad etichette intere. Siano date le seguenti definizioni:

- una foglia si dice *sinistra* (*destra*) se è figlio sinistro (destro) di un nodo padre;
- per ogni nodo  $x$  si definisce  $lsx$  ( $ldx$ ) il numero delle foglie sinistre (destre) che fanno parte del sottoalbero radicato in  $x$ ;
- per ogni nodo si definisce  $d$  la distanza di detto nodo dalla radice dell'albero;
- per ogni nodo si definisce  $v = (d \times lsx) + (K \times ldx)$ , con  $K$  valore intero.

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);
- calcoli  $v$  per ogni nodo dell'albero; (complessità al più  $\mathcal{O}(n)$ )
- stampi i valori  $v$  ordinati in maniera non decrescente. (complessità al più  $\mathcal{O}(n \log n)$ )

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

5 2  
9  
7  
6  
8  
10

### Output

0  
0  
0  
3  
4

