

Algoritmi e Strutture Dati – Prova di Laboratorio

08/06/2016

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. Si ricorda di avvisare i docenti una volta che il server ha accettato una soluzione come corretta.

Esercizio

Si consideri un sistema per la gestione di alberi binari di ricerca (ABR) in grado di memorizzare nodi con le seguenti caratteristiche.

Ogni nodo è caratterizzato da un intero ID , da un intero P che ne rappresenta il **peso**, e da un intero C_{max} che ne rappresenta il **carico massimo**. Il **carico** di un nodo n è definito come la somma dei pesi di tutti i nodi facenti parte del sottoalbero radicato in n , escluso quest'ultimo. Il carico di un nodo è **tollerabile** se \leq rispetto al suo carico massimo. Un ABR si dice *integro* se tutti i suoi nodi hanno un carico tollerabile.

Il sistema dovrà inserire i nodi all'interno dell'ABR usando l' ID come etichetta. I valori devono essere inseriti nello stesso ordine con cui vengono letti (per convenzione, le etichette \leq vanno inserite a sinistra).

Scrivere un programma che:

- legga da tastiera una sequenza di N triple $[ID, P, C_{max}]$ ciascuna rappresentante un nodo, e le inserisca in un ABR;
- stampi la stringa 'ok' se l'ABR è integro e 'no' altrimenti;
- in caso l'ABR **non** sia integro, stampi l' ID dei nodi con carico **non** tollerabile in ordine di ID non decrescente.

L'**input** è formattato nel seguente modo: la prima riga contiene l'intero N . Seguono N righe contenenti una tripla $[ID, P, C_{max}]$ ciascuna, con gli elementi separati da uno spazio.

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

Esempio

Input

```
6
4 2 5
8 2 1
6 1 4
9 1 6
3 1 6
2 1 1
```

Output

```
no
4
8
```

