

Algoritmi e Strutture Dati – Prova di Laboratorio

10/06/2014

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

Scrivere un programma che legga da tastiera un intero N , una sequenza A di N coppie **[intero,stringa]** ed un intero D . Ogni coppia rappresenta il nodo di un albero binario di ricerca. Il programma deve:

- Inserire uno alla volta, nell'ordine dato, le coppie **[intero,stringa]** di A in un albero binario di ricerca **senza ribilanciamento**. L'inserimento deve essere tale per cui, per un qualsiasi nodo, il sottoalbero sinistro contenga le coppie il cui valore intero è **minore o uguale** del valore intero del nodo, mentre il sottoalbero destro contiene le coppie il cui valore intero è **maggiore**.
- Stampare **in ordine lessicografico** tutte le stringhe che si trovano nell'albero in nodi a profondità D . Si ricorda che la profondità di un nodo è uguale alla sua **distanza dalla radice e che la profondità della radice è 0**

L'input è formattato nel seguente modo. Le prime due righe contengono i due interi N e D . Seguono $2N$ righe, due righe per coppia. Per ogni coppia abbiamo la prima riga che contiene il valore intero e la seconda che contiene la stringa associata. Si assume che N è sempre maggiore di zero.

L'output, se esistono nodi a profondità D , deve stampare nella prima riga il numero di nodi trovati e successivamente una riga per ogni stringa trovata. **Si ricorda che le stringhe vanno restituite in ordine lessicografico.** Se non esistono nodi a profondità D , il programma stampa solo 0.

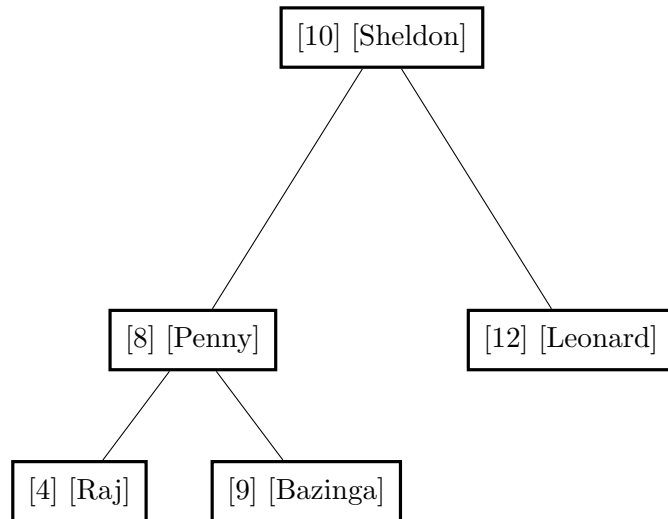
Esempi

Input

5
2
10
Sheldon
12
Leonard
8
Penny
4
Raj
9
Bazinga

Output

2
Bazinga
Raj



Input

5
10
10
Sheldon
12
Leonard
8
Penny
4
Raj
9
Bazinga

Output

0

