

Algoritmi e Strutture Dati – Test d’Esame

27/05/2022

Istruzioni

Risolvere il seguente esercizio implementando un programma in un singolo file .cpp, completo di funzione *main*. Si presti particolare attenzione alla formattazione dell’input e dell’output, e alla complessità target indicata per ciascuna funzionalità. Nel caso la complessità target non sia specificata, si richiede che sia la migliore possibile. La lettura dell’input e la scrittura dell’output **DEVONO** essere effettuate tramite gli stream **cin** e **cout** rispettivamente. La correzione avverrà prima in maniera automatica inviando il file .cpp al server indicato in aula. Quest’ultimo esegue dei test confrontando l’output prodotto dalla vostra soluzione con l’output atteso. In caso la verifica abbia esito positivo sarà possibile consegnare il compito, il quale verrà valutato dai docenti in termini di complessità. Si ricorda che è possibile testare la correttezza del vostro programma in locale su un sottoinsieme dei input/output utilizzati nella seguente maniera. 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.

Esercizio

Scrivere un programma che legga da tastiera una sequenza di N interi positivi e li inserisca in un albero binario di ricerca (**senza** ribilanciamento) nello stesso ordine con il quale vengono forniti in input.

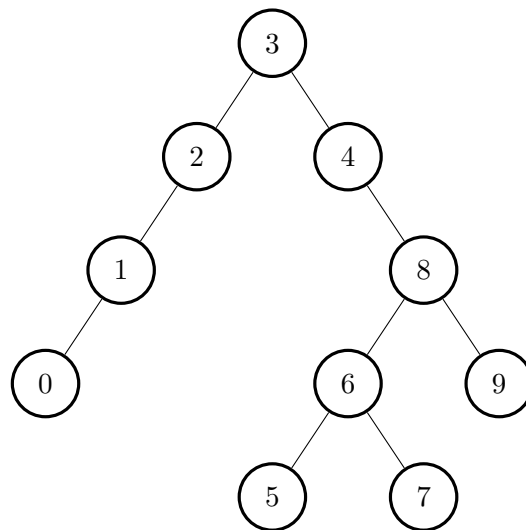
Il programma deve verificare che l'albero soddisfi la seguente proprietà: *per ogni nodo* dell'albero, le altezze dei suoi sottoalberi sinistro e destro devono differire al massimo di uno.

L'input è formattato nel seguente modo: la prima riga contiene l'intero N . Seguono N righe contenenti un intero ciascuna. L'output è formato da una sola riga contenente la stringa *ok* qualora la proprietà sopra descritta sia verificata, la stringa *no* altrimenti.

Esempio 1

Input

10
3
2
1
4
0
8
6
7
5
9



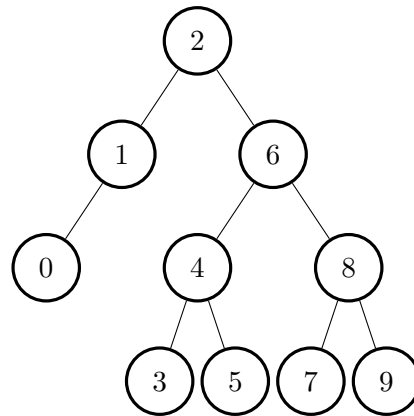
Output

no

Esempio 2

Input

10
2
1
0
6
4
3
5
8
7
9



Output

ok