Laboratorio
26 Aprile 2012

Albero binario di ricerca: Ricerca

Scrivere un programma che legga da tastiera una sequenza di n interi distinti e li inserisca in un albero binario di ricerca (senza ribilanciamento). Il programma entra poi in un ciclo infinito nel quale legge un intero i da tastiera e lo cerca nell'albero. Se i si trova nell'albero stampa la profondità alla quale l'elemento si trova (contando da 0) altrimenti stampa NO.

L'input è formattato nel seguente modo: nella prima riga si trova la lunghezza della sequenza e nelle successive si trovano gli interi che compongono la sequenza.

Albero binario di ricerca: Visita

Scrivere un programma che legga da tastiera una sequenza di n interi distinti e li inserisca in un albero binario di ricerca (senza ribilanciamento). Il programma deve visitare opportunamente l'albero e restituire la sua altezza.

L'input è formattato nel seguente modo: nella prima riga si trova la lunghezza della sequenza e nelle successive si trovano gli interi che compongono la sequenza.

Albero binario di ricerca: Ordinamento

Scrivere un programma che legga da tastiera una sequenza di n interi distinti e li inserisca in un albero binario di ricerca (senza ribilanciamento). Il programma deve quindi utilizzare un'opportuna visita dell'albero per stampare gli interi della sequenza in ordine non decrescente.

L'input è formattato nel seguente modo: nella prima riga si trova la lunghezza della sequenza e nelle successive si trovano gli interi che compongono la sequenza.

Prova di esame
11 Giugno 2011

Sono date in input due sequenze di N interi positivi dalle quali devono essere costruiti due alberi binari di ricerca NON bilanciati (un albero per sequenza). Al programma viene data una chiave intera K. Si può assumere che l'intero K sia presente in entrambe le sequenze. Il programma deve verificare che le sequenza di chiavi incontrate nel cammino che dalla radice porta al nodo con chiave K nei due alberi coincidano.
L'input è formattato nel seguente modo. Nella prima riga sono contenuti gli interi N e K separatati da uno spazio. Seguono poi $2N$ righe contenenti ognuna un intero. I primi N interi appartengono alla prima sequenza mentre i successivi N interi appartengono alla seconda sequenza.
L'output invece è costituito da una singola riga che contiene il risultato del programma: $1\backslash n$ se le due sequenze di chiavi coincidono, $0\backslash n$ altrimenti.

Esempi

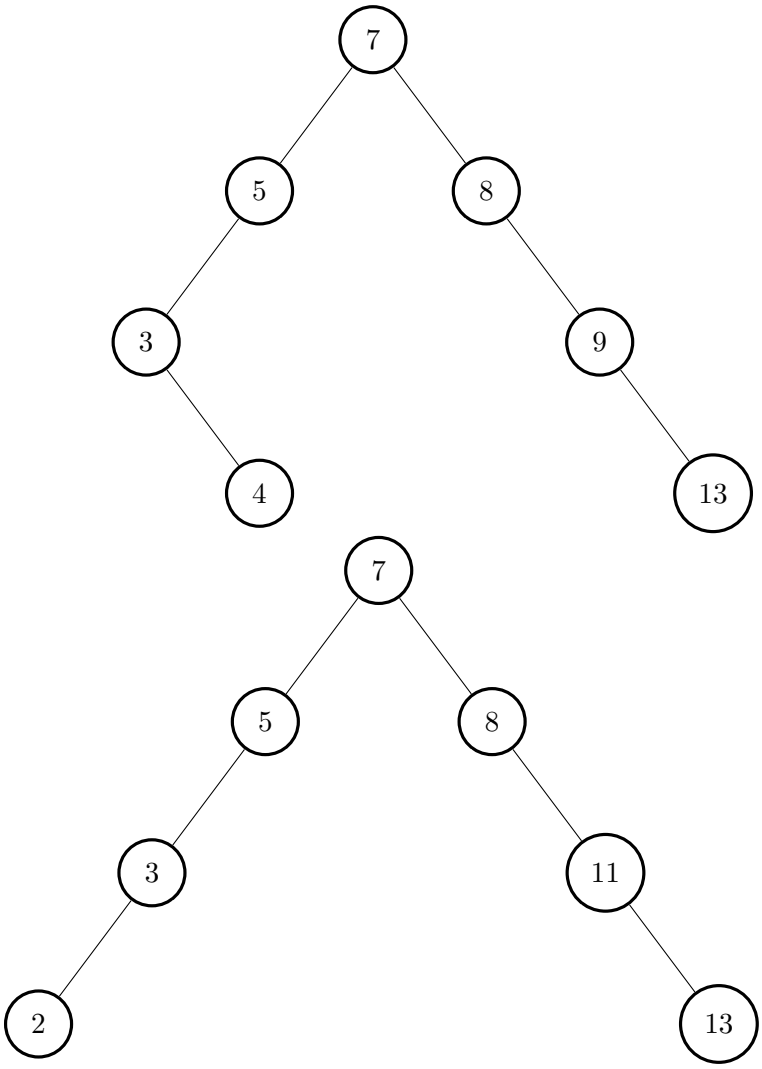
Input	Output
7 3 7 5 3 4 8 9 13 7 8 11 13 5 3 2	1

Input

```
7 13
7
5
3
4
8
9
13
7
8
11
13
5
3
2
```

Output

```
0
```



Sopra sono riportati i due alberi binari di ricerca ottenuti per gli interi dell'esempio. Nel primo esempio entrambe le sequenze di chiavi dalla radice al nodo con chiave 3 sono uguali a $7, 5, 3$. L'output corretto del programma è quindi $1\backslash n$. Nel secondo caso la sequenza di chiavi dalla radice al nodo con chiave 13 è $7, 8, 9, 13$ per il primo albero e $7, 8, 11, 13$ per il secondo. L'output corretto del programma è quindi $0\backslash n$.

Prova di esame
1 Febbraio 2012

Il programma riceve in input una sequenza di N interi positivi e deve costruire un albero **ternario** di ricerca **non** bilanciato. L'ordine di inserimento dei valori nell'albero deve coincidere con quello della sequenza.

Ogni nodo in un albero ternario di ricerca può avere fino a tre figli: figlio sinistro, figlio centrale e figlio destro. L'inserimento di un nuovo valore avviene partendo dalla radice dell'albero e utilizzando la seguente regola.
Il valore da inserire viene confrontato con la chiave del nodo corrente. Ci sono tre possibili casi in base al risultato del confronto.

1. Se il valore è minore della chiave del nodo corrente, esso viene inserito ricorsivamente nel sottoalbero radicato nel figlio sinistro;

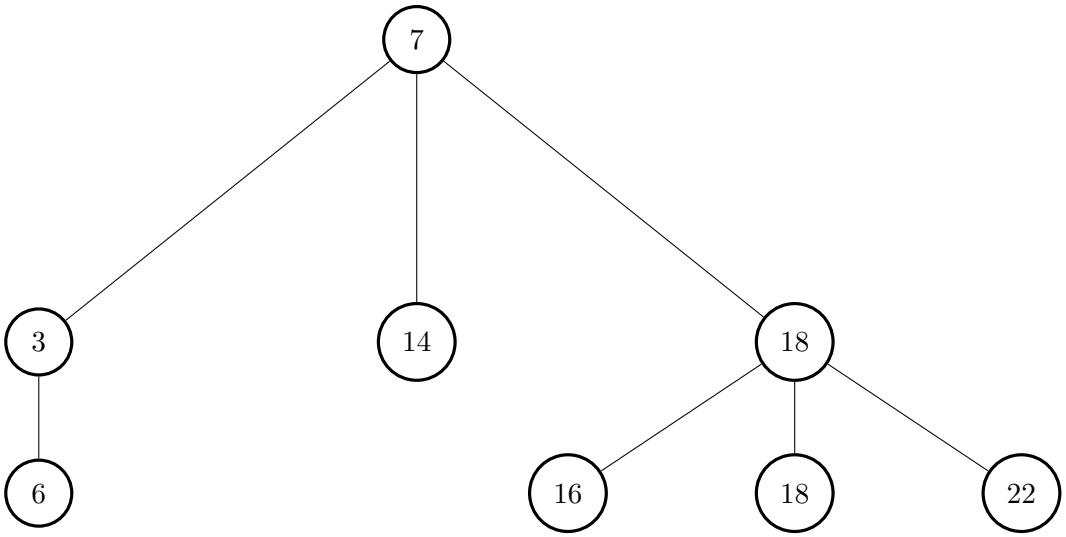
2. Se il valore è **divisibile** per la chiave del nodo corrente, esso viene inserito ricorsivamente nel sottoalbero radicato nel figlio centrale;

3. In ogni altro caso il valore viene inserito ricorsivamente nel sottoalbero radicato nel figlio destro.

Il programma deve riportare il numero di nodi dell'albero che hanno **tre** figli.
L'input è formattato nel seguente modo. Nella prima riga è contenuto l'intero N. Seguono poi N righe contenenti ognuna un intero.
L'output invece è costituito da una singola riga che contiene il risultato del programma.

Esempio

Input	**Output**
8 7 3 14 18 6 18 22 16	2


7
3
14
18
6
16
18
22

Puzzled
Ordinamento

Viene dato un array A di n interi da $\log n$ bit ciascuno (quindi valori da 0 a $2^{\log n} - 1 = n - 1$). Ordinarli *in-place* (cioè usando $O(1)$ spazio aggiuntivo) in $O(n \log n)$.

Puzzled
Orco e hobbit

Un orco ha rapito N hobbit e gli propone: "Domani mattina metterò un cappello a ciascuno di voi. Ogni cappello sarà etichettato con un numero da 0 a $N-1$, doppioni sono possibili. Ogni hobbit potrà vedere il numero sui cappelli degli altri ma non il suo. Quando suonerò la campanella, tutti gli hobbit dovranno simultaneamente dire un numero (potenzialmente diverso per ogni hobbit). Se almeno un hobbit sarà in grado di indovinare il suo numero, tutti gli hobbit saranno liberati." La sera stessa gli hobbit si incontrano e individuano una strategia che è **sempre** vincente, quale?

Per $N = 2$ la strategia è relativamente facile mentre la soluzione per N arbitrario è decisamente più difficile.