

Introduzione agli Algoritmi (canale EO, A.A. 2008-2009)
Prova di esame del 14/7/2009
Prof.ssa Irene Finocchi

GLI STUDENTI ESONERATI DEVONO SVOLGERE SOLO GLI ESERCIZI DELLA PARTE 2 (TEMPO CONCESSO: 1H 30'). GLI ALTRI STUDENTI DEVONO SVOLGERE GLI ESERCIZI DI ENTRAMBE LE PARTI (TEMPO CONCESSO: 3H).

Parte 1

Esercizio 1.1 (punti 12/30)

Un array A di n valori interi è *unimodale perfetto* se consiste di una metà crescente seguita da una metà decrescente. Più precisamente, deve risultare:

- $A[i] \leq A[i+1]$ per ogni intero $i \in [1, n/2)$;
- $A[i] \geq A[i+1]$ per ogni intero $i \in [n/2, n)$.

Progettare un algoritmo che, dato un array A , lo trasformi in un array unimodale perfetto. Dimostrare la correttezza dell'algoritmo proposto ed analizzarne il tempo di esecuzione nel caso peggiore.

Esercizio 1.2 (punti 18/30)

Considerare la seguente funzione:

```
void analizzami(array T, int inizio, int fine) {
    n = fine - inizio + 1
    if (n>0){
        alloca un array S di dimensione n
        copia(S, T, inizio, fine)
        sort(S,n)
        analizzami(T, inizio, inizio+n/3)
        analizzami(T, inizio+n/3+1, inizio+2*n/3)
        analizzami(T, inizio+2*n/3+1, fine)
    }
}
```

dove $\text{copia}(S, T, i, j)$ copia in S il contenuto dell'array T dalla posizione i alla posizione j e $\text{sort}(T, n)$ ordina il vettore S di dimensione n .

Studiare il tempo di esecuzione della funzione analizzami nel caso peggiore e nel caso migliore, nel caso in cui sort sia:

- a. il mergeSort ;
- b. l' insertionSort .

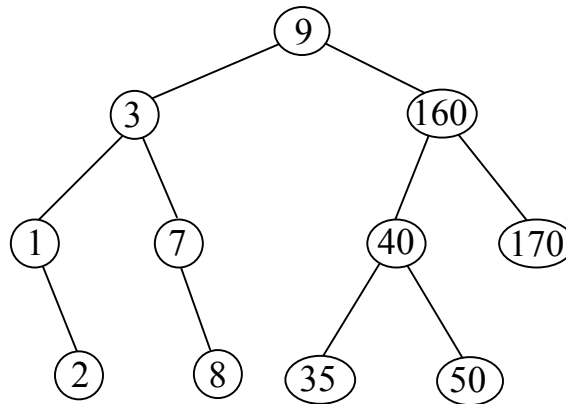
Introduzione agli Algoritmi (canale EO, A.A. 2008-2009)
Prova di esame del 14/7/2009
Prof.ssa Irene Finocchi

Parte 2

Esercizio 2.1 (punti 17/30)

Preso un albero binario di ricerca T con chiavi intere, descrivere un algoritmo $\text{printM}(T)$ che, per ogni nodo, stampi la chiave contenuta nel nodo se essa è maggiore della somma delle chiavi nei nodi precedenti. L'algoritmo dovrebbe avere tempo di esecuzione lineare. Dimostrare la correttezza dell'algoritmo proposto ed analizzarne il tempo di esecuzione.

Esempio: dato l'albero sottodisegnato



l'algoritmo dovrà stampare: 1, 2, 7, 35, 160

Esercizio 2.2. (punti 13/30)

A partire dall'albero AVL binario completo contenente le chiavi 10, 20, 30, 40, 50, 60, 70, eseguire nell'ordine la seguente sequenza di operazioni:

```
insert(1)
insert(2)
insert(15)
delete(60)
delete(50)
insert(18)
delete(10)
insert(16)
delete(1)
delete(20)
```

Per ogni operazione, mostrare i fattori di bilanciamento dei nodi ed eventualmente indicare il sottoalbero che crea lo sbilanciamento e le rotazioni effettuate.