

## Introduzione agli Algoritmi

9 giugno 2015

Prof. Emanuela Fachini (canale 1) e Prof. Irene Finocchi (canale 2)

### Parte I

*Le risposte non motivate non saranno prese in considerazione.*

*Negli esercizi di progettazione, prima di passare allo pseudocodice descrivete l'idea algoritmica sottostante. Per gli algoritmi progettati è necessario analizzare tempo di esecuzione e correttezza.*

#### Esercizio 1

Date due funzioni  $f(n) = \Omega(\lg n)$  e  $g(n) = O(n)$ , si considerino le seguenti affermazioni.

- $g(n) = O(f(n))$
- $f(n) = \Omega(\lg g(n))$
- $f(n) + g(n) = O(n)$

Per ciascuna affermazione si stabilisca se è sempre vera, sempre falsa, o se esistono casi in cui è vera e casi in cui è falsa. Per ogni affermazione né sempre vera né sempre falsa, si specifichino due funzioni  $f(n)$  e  $g(n)$  che la rendono vera e due che la rendono falsa.

#### Esercizio 2

Si determini il tempo di esecuzione nel caso peggiore  $T(n)$  della seguente funzione:

```
void f(int A[], int inizio, int fine) {
    int n = fine - inizio + 1
    sia B un array di n interi
    if (n > 0) {
        copia(B, A, inizio, fine)
        f(A, inizio, inizio+n/3)
        f(A, inizio+2n/3+1, fine)
    }
}
```

dove  $\text{copia}(B, A, i, j)$  copia in B il contenuto dell'array A dalla posizione i alla posizione j.

## Introduzione agli Algoritmi

9 giugno 2015

Prof. Emanuela Fachini (canale 1) e Prof. Irene Finocchi (canale 2)

### Esercizio 3

Si consideri un array di  $n$  numeri inizialmente ordinato in ordine crescente, ma nel quale  $k$  valori sono stati diminuiti per cui risultano “fuori posto”.

- Si scriva un algoritmo ***in loco*** che riordina l’array in tempo  $O(kn)$  nel caso peggiore.
- Si scriva un algoritmo alternativo, ***non*** necessariamente ***in loco***, che riordina l’array in tempo  $O(n+k \lg k)$  nel caso peggiore.

## Introduzione agli Algoritmi

9 giugno 2015

Prof. Emanuela Fachini (canale 1) e Prof. Irene Finocchi (canale 2)

### Parte II

*Le risposte non motivate non saranno prese in considerazione.*

*Negli esercizi di progettazione, prima di passare allo pseudocodice descrivete l'idea algoritmica sottostante. Per gli algoritmi progettati è necessario analizzare tempo di esecuzione e correttezza.*

#### Esercizio 4

Si dimostri che nel modello basato su confronti sono necessari  $\Omega(n \lg n)$  confronti per ordinare  $n$  elementi arbitrari.

#### Esercizio 5

1. Si consideri un AVL completo fino al penultimo livello e avente nell'ultimo livello solo due foglie posizionate il più a sinistra possibile. Si dimostri che se si cancella la prima foglia sul penultimo livello (da sinistra) basta una sola rotazione per ribilanciare l'albero.
2. Si disegni un albero AVL con 7 nodi in cui la cancellazione di un nodo comporta una rotazione doppia; si disegni l'albero prima e dopo la cancellazione e la rotazione.

#### Esercizio 6

Si consideri la classe ABRPS di Alberi Binari di Ricerca Pesanti a Sinistra, ovvero alberi binari di ricerca in cui per ogni nodo il numero dei nodi nel suo sottoalbero sinistro è maggiore o uguale a quello dei nodi nel suo sottoalbero destro.

Ogni ABRPS  $T$  è rappresentato in memoria con strutture a puntatori e ogni nodo  $v$  ha un campo aggiuntivo *size* che contiene il numero dei nodi nel sottoalbero radicato in  $v$ .

Si scriva un algoritmo per l'inserimento di un nuovo elemento in un ABRPS, utilizzando come operazione di base il normale inserimento in un ABR e mettendo in evidenza quali operazioni sono necessarie sull'albero per ripristinare la proprietà di essere pesante a sinistra.

Si dimostri la correttezza delle operazioni di ripristino, in modo analogo a come si è fatto nel caso degli AVL.