

# Algoritmi e Strutture Dati 1

Appello dell' 8/09/2010

## Esercizio 1

- Si presenti un algoritmo di ordinamento a scelta visto a lezione per ordinare in modo non decrescente un array di numeri interi, e si dia una stima della sua complessità computazionale.
- Si consideri ora la seguente classe Java:

```
public class Elem{
    public int blocco;
    public int valore;
}
```

Modificando l'algoritmo presentato, scrivere un metodo **public static void sort (Elem [] A)** in Java che preso in input un array **A** di **n** oggetti di tipo Elem, ordina l'array in modo che gli elementi con lo stesso numero di **blocco** siano tutti consecutivi nell'array, e ordinati in modo **non crescente rispetto a valore**. Inoltre i blocchi devono essere ordinati in modo **non decrescente rispetto a blocco**.

**Suggerimento:** definire un metodo statico che confronta due Elem identificando chi va prima nell'ordinamento (`public static boolean isLessOrEqual (Elem x, Elem y)`).

## Esercizio 2

Dire se, giustificando la risposta applicando la definizione delle notazioni asintotiche,

- a)  $n \log n - 5n = \theta(n^2)$
- b)  $n\sqrt{n} + n = O(n^2)$

## INIZIO SECONDO PARZIALE

## Esercizio 3

- a. Mostrare l'heap di massimo che si ottiene a partire dall'heap vuoto inserendo nell'ordine indicato i seguenti elementi:
  - **13, 2, 60, 22, 67, 5, 25, 12, 3, 24, 28**
- b. Scrivere FUORI dalla classe Heap un metodo **public static int getfourthMax (Heap h)** che, senza modificare l'heap su cui lavora e senza poter accedere direttamente all'albero dell'heap, ma solo potendo utilizzare tutti i metodi già presenti nella classe Heap ed implementati a lezione, restituisce il quarto massimo presente nell'heap, facendo uso di tutti gli altri. Si stimi la complessità computazionale di **getfourthMax** in funzione di **n**, dove **n** è il numero di elementi nell'heap.

## Esercizio 4

Si descriva brevemente come può essere risolto il problema delle collisioni nelle tabelle Hash (sia con lista di trabocco che con indirizzamento aperto).

## Esercizio 5

Si consideri un albero binario di ricerca. Scrivere un metodo ricorsivo **public static int height (Node x)** che preso in input un nodo **x**, restituisce l'altezza del sottoalbero radicato al nodo **x**.

### Attenzione:

- Scrivere **subito** nome, cognome, matricola e numero del compito su OGNI FOGLIO.
- non è ammesso **per nessun motivo** l'uso di telefoni cellulari, calcolatrici, etc...
- **non** è possibile consultare appunti, libri, dispense.