

# Algoritmi e Strutture Dati 1

Appello del 6/07/2011

## Esercizio 1 (9 punti)

Scrivere un metodo `public static int secondMax (int [] A)` in Java che preso in input un array A di numeri interi restituisce in output il secondo elemento più grande presente nell'array A.

La procedura **NON può modificare** l'array A. Si assuma che l'array A contiene almeno due elementi.

Ad esempio, se  $A=[1,5,2,5,-1,3]$ , il metodo deve restituire 5. Se invece  $A=[1,5,2,-1,3]$  il metodo deve restituire 3.

Il metodo deve avere tempo di esecuzione nel caso peggiore  $O(n)$ , e NON può sfruttare nessuno dei metodi visti a lezione.

Si stimi la complessità dell'algoritmo proposto, e si dica, giustificando la risposta, se possono esistere algoritmi aventi tempo di esecuzione asintoticamente migliore.

## Esercizio 2 (9 punti)

$$\text{Sia } T(n) = \begin{cases} T(n/5) + c_1 & \text{se } n > 1 \\ c_2 & \text{se } n = 1 \end{cases}$$

dove  $c_1$  e  $c_2$  sono due costanti.

Si dia una stima esplicita (non ricorsiva) di  $T(n)$  facendo uso del **metodo di sostituzione ed induzione**.

## Esercizio 3 (9 punti)

Si consideri la classe SimpleList

```
public class SimpleList{
    int key;
    SimpleList next;
    ...
}
```

che rappresenta un elemento di una lista e possiede una chiave intera ed il puntatore all'elemento successivo. Si noti come una lista è identificata in modo naturale dal suo elemento di testa.

Scrivere un metodo **iterativo** `public static SimpleList ricerca (SimpleList L, int k)` in Java che data una SimpleList L ed una chiave k restituisce **il puntatore ad un elemento che contiene la chiave k** se tale chiave k è presente nella lista L, **null** altrimenti.

## Esercizio 4 (9 punti)

- Mostrare l'albero binario di ricerca che si ottiene a partire dall'albero vuoto inserendo nell'ordine indicato i seguenti elementi:
  - **24, 2, 6, 15, 8, 90, 5, 22**
- Si consideri un albero binario. 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 [se x è un foglia l'albero radicato in x ha altezza 0].