

# INFORMATICA GENERALE

## Sessione Autunnale

### Esame Scritto

docenti:

GIANCARLO BONGIOVANNI, TIZIANA CALAMONERI, IVANO SALVO  
Sapienza Università di Roma  
24 Settembre 2015

**Esercizio 1 (10 punti)** Si consideri la seguente funzione in pseudocodice, che prende come parametri un vettore  $V$  di interi e due indici *primo* e *ultimo* che alla prima chiamata valgono 1 ed  $n$  rispettivamente:

```
funzione MagicFunction (V: vettore; primo, ultimo: intero)
  if (primo >= ultimo) return
  for i = primo to ultimo
    V[i] <- V[i] - V[ultimo - i + 1]
  medio <- FLOOR((primo + ultimo)/2)
  for i = 1 to 2
    MagicFunction(V, primo, medio)
  for i = 1 to 2
    MagicFunction(V, medio+1, ultimo)
return
```

Da essa si ricavi l'equazione di ricorrenza che ne esprime il costo computazionale, specificando i contributi delle varie istruzioni (**1 punto**). Inoltre, si risolva l'equazione di ricorrenza trovata utilizzando:

- (**3 punti**) il metodo iterativo;
- (**2 punti**) il metodo dell'albero;
- (**1 punto**) il metodo principale;
- (**3 punti**) il metodo di sostituzione.

In tutti i casi, si dettagli il procedimento usato e si giustificino le proprie risposte.

**Esercizio 2 (10 punti)** Scrivere una funzione C di prototipo:

```
int shuffle(lista l, lista m, lista n);
```

che restituisce 1 se la lista puntata da  $n$  contiene l'unione degli elementi delle liste puntate da  $l$  ed  $m$ , collocati in modo tale che, nella lista puntata da  $n$ , la sequenza di elementi di ciascuna delle altre due liste mantenga lo stesso ordine.

Ad esempio, se  $l$  ed  $m$  puntano alle liste  $1 \rightarrow 2 \rightarrow 3$  e  $4 \rightarrow 5$ , la funzione `shuffle` restituisce 1 se  $n$  punta alla lista  $1 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 3$  mentre restituisce 0 se  $n$  punta alla lista  $1 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 2$ .

**Nota:** Si assuma che le liste  $l$  ed  $m$  non abbiano elementi in comune.

**Esercizio 3 (10 punti)** Sia dato un heap massimo memorizzato *tramite record e puntatori* (e non, come al solito, tramite notazione posizionale); si progetti una funzione `Cerca-Min-In-Heap` che, preso in input il puntatore alla radice dell'heap, restituisca un puntatore al record con chiave minima.

Dell'algoritmo progettato:

- a. (4 punti) Si scriva l'idea a parole, evidenziando le differenze apportate rispetto all'algoritmo che si usa quando l'heap é memorizzato tramite vettore;
- b. (4 punti) Si dia lo pseudocodice (dettagliando anche eventuali funzioni viste a lezione);
- c. (2 punti) Si calcoli il costo computazionale.