

# INFORMATICA GENERALE

## Secondo Esonero e Compito Scritto

docenti: TIZIANA CALAMONERI, IVANO SALVO  
Sapienza Università di Roma

11 giugno 2012

**Coloro che intendono partecipare alla prova di esonero devono svolgere gli esercizi 2, 3 e 4; coloro che, invece, intendono partecipare all'intero compito scritto devono svolgere gli esercizi 1, 2 e 3.**

**Esercizio 1 (10punti)** Siano dati  $n$  interi nell'intervallo  $[1, k]$ , per qualche intero  $k$ . Si vuole fare un'analisi preliminare di questi  $n$  interi in modo tale che costi poi  $O(1)$  rispondere ad una qualunque domanda su quanti degli  $n$  interi cadono nell'intervallo  $[a, b]$ , dove  $a$  e  $b$  sono interi;

1. (**2 punti**) si descriva una struttura dati idonea a memorizzare le informazioni necessarie e se ne discuta l'occupazione in memoria;
2. (**4 punti**) si progetti un algoritmo che consenta di riempire opportunamente la precedente struttura dati;
3. (**2 punti**) si scriva lo pseudocodice di tale algoritmo;
4. (**2 punti**) se ne valuti la complessità, che dovrebbe essere  $O(n + k)$ .

**Esercizio 2 (10 punti)** Sia dato un grafo  $G$  mediante la sua matrice di incidenza  $I$ . Si vogliono ottenere le liste di adiacenza dello stesso grafo;

1. (**4 punti**) si descriva a parole l'algoritmo di trasformazione;
2. (**3 punti**) si scriva lo pseudocodice di tale algoritmo;
3. (**3 punti**) se ne valuti la complessità.

**Esercizio 3 (12 punti)** Si consideri il problema di prendere in input una lista di interi e calcolare la lista in cui tutti gli elementi sono sostituiti con la somma dei successivi (elemento incluso). Ad esempio, presa in input la lista  $\langle 1, 7, 11 \rangle$  si vuole ottenere in output la lista  $\langle 19, 18, 11 \rangle$ . Procedere come segue:

1. **(3 punti)** specificare la funzione mediante equazioni ricorsive su sequenze;
2. **(3 punti)** scrivere una funzione `sommaSuccessiviFun` che crea una *nuova* lista;
3. **(3 punti)** scrivere una funzione `sommaSuccessiviRec` ricorsiva che *modifica* la lista di ingresso;
4. **(3 punti)** scrivere una funzione `sommaSuccessiviIter` iterativa che modifica la lista di ingresso.

**Esercizio 4 (11 punti)** L'ordine delle operazioni effettuate da una coda può essere simulato grazie all'uso di due pile. Immaginando di avere un esecutore che può solo fare dei `push(x, i)/pop(i)/test_di_lista_vuota(i)` su due pile ( $i$  può assumere i valori 0 ed 1 ed indica su quale pila le operazioni sono eseguite):

1. **(3+2 punti)** si descrivano a parole le operazioni di *dequeue* ed *enqueue* con le due pile;
2. **(3 punti)** si scriva lo pseudocodice delle due funzioni,
3. **(3 punti)** si valuti la complessità delle due funzioni.