

# INFORMATICA GENERALE

## Sessione Invernale

### Esame Scritto

docenti: GIANCARO BONGIOVANNI, TIZIANA CALAMONERI, IVANO SALVO  
Sapienza Università di Roma

19 Gennaio 2018

**Esercizio 1 (10 punti)** Si consideri la seguente equazione di ricorrenza:

$$T(n) = 9T(n/3) + \Theta(n)$$
$$T(1) = \Theta(1)$$

e la si risolva 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 giustifichino le proprie risposte. Si discuta poi **(1 punto)** brevemente come cambierebbe la soluzione se il termine additivo  $\Theta(n)$  fosse invece  $\Theta(n^2)$  e se il fattore 9 fosse invece 3.

**Esercizio 2 (10 punti)** Considerare le liste di interi definite a lezione. Scrivere una funzione `int prodottoScalare(list L, list M)` tra liste procedendo nel seguente modo:

- definire il tipo di dato `pairList` delle liste che contengono *coppie* di interi.
- scrivere una funzione `int sommaLista(list L)`; che calcola la somma di tutti gli elementi di una lista di interi;
- scrivere una funzione `pairList zip(list L, list M)` che date due liste di interi della stessa lunghezza costruisce la lista di coppie in cui l'*i*-esimo elemento di *L* è accoppiato con l'*i*-esimo elemento di *M*. Ad esempio, `zip(<1, 7, 70>, <17, 11, 75>) = <<1, 17>, <7, 11>, <70, 75>>` (Dire cosa calcola la vostra funzione (o cosa accade) se le liste di ingresso non soddisfano le precondizioni, cioè quando non hanno la stessa lunghezza);

4. scrivere una funzione `list prodotti(pairList L)` che ricevendo una lista di coppie di interi, ritorna la lista di interi contenente i prodotti di ciascuna coppia in  $L$ . Ad esempio, `prodotti(((1, 17), (7, 11), (70, 75))) = <17, 77, 5250>`
5. scrivere infine la funzione `int prodottoScalare(list L, list M)` in una sola riga semplicemente invocando opportunamente le funzioni richieste ai punti precedenti.

Osservate che le funzioni `zip` e `prodotti` *devono* allocare nuove liste.

**Esercizio 3 (10 punti)** Sia dato un albero binario di ricerca  $T$  ed il puntatore ad un suo nodo  $x$ .

1. **(3 punti)** Si illustri a parole l'algoritmo per trovare il nodo successore di  $x$  (cioè la cui chiave seguirebbe quella di  $x$  se le chiavi di  $T$  fossero ordinate in senso crescente), dettagliando i vari casi.
2. **(3 punti)** Si dettagli lo pseudocodice, considerando che  $T$  è memorizzato tramite record e puntatori ed è dato tramite puntatore alla sua radice.
3. **(2 punti)** Si dimostri il costo computazionale dell'algoritmo descritto.
4. **(2 punti)** Si dia un esempio per ciascun caso considerato.