

# INFORMATICA GENERALE

## Esame Scritto

docenti: G. BONGIOVANNI, T. CALAMONERI, I. SALVO  
Sapienza Università di Roma

7 settembre 2017

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

$$T(n) = 4T(n/2) + \Theta(n^2)$$
$$T(1) = \Theta(1).$$

Si risolva tale equazione di ricorrenza utilizzando:

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

**Esercizio 2 (10 punti)** Considerare il problema di verificare se una lista di caratteri è palindroma, cioè se letta da sinistra a destra o da destra a sinistra dà la stessa sequenza di caratteri (come le parole 'non', 'otto', 'radar', 'ingegni', 'onorarono', 'ossesso', ...).

- (3 punti) Considerate il seguente programma (dove la funzione `eqList` verifica se due liste sono uguali e `reverse` restituisce il puntatore alla testa di una lista che contiene gli elementi di `L` in ordine rovesciato):

```
int palindroma(charList L){
    if (eqList(L, reverse(L)) return 1;
    else return 0;
}
```

Sotto quali ipotesi sulla funzione `reverse` la funzione dà risultati corretti?

2. **(6 punti)** Scrivere una funzione C *ricorsiva* `int palindroma(charList L)` che restituisce 1 solo se la lista  $L$  in ingresso è palindroma *senza* usare strutture dati ausiliarie.
3. **(1 punto)** Quale struttura dati ausiliaria viene in realtà allocata *implicitamente* durante l'esecuzione della vostra funzione?

**Esercizio 3 (10 punti)** Sia dato un albero binario **qualunque**  $T$  memorizzato tramite liste di adiacenza  $L$  (quindi come un più generale grafo) ed un suo nodo  $r$  che ne sia la radice. Si progetti un algoritmo che, presi in input  $L$  ed  $r$ , stampi le chiavi di tutte le foglie di  $T$ .

Dell'algoritmo proposto:

1. **(2 punti)** Si dia la descrizione a parole;
2. **(2 punti)** Si scriva lo pseudocodice;
3. **(2 punti)** Si calcoli il costo computazionale, dettagliando comunque il costo delle funzioni studiate eventualmente usate;
4. **(4 punti)** Si consideri ora lo stesso albero memorizzato tramite la notazione posizionale.  
Qual è, nel caso peggiore, la dimensione della struttura dati rispetto alla dimensione dell'input?  
È possibile adattare l'algoritmo proposto a questa struttura dati? Perché?  
Si discuta brevemente come cambiano eventualmente l'algoritmo ed il costo computazionale.