

# INFORMATICA GENERALE

## Secondo Esonero

GIANCARLO BONGIOVANNI, TIZIANA CALAMONERI, IVANO SALVO  
Sapienza Università di Roma  
18 Giugno 2019

**Esercizio 1 (12 punti)** Un albero binario di ricerca  $T$  sia memorizzato tramite record e puntatori e sia dato tramite il puntatore  $r$  alla sua radice. Il record che rappresenta ciascun nodo  $v$  contenga, oltre i campi con la chiave ed i puntatori al padre e ai figli sinistro e destro, un campo *num* che memorizza il numero di nodi contenuti nel sottoalbero radicato in  $v$ .

Dato un puntatore  $pv$  ad un nodo  $v$  di  $T$ , si consideri il problema di individuare quale sia la posizione  $i$  della chiave di  $v$  nella sequenza ordinata delle chiavi, ovvero tale che esistano esattamente  $i - 1$  chiavi minori di essa memorizzate nell'albero.

Si scriva un algoritmo efficiente che, sfruttando l'informazione aggiuntiva contenuta nei nodi di  $T$ , restituisca il valore di  $i$ .

Dell'algoritmo proposto, si dia:

1. **(5 punti)** la descrizione a parole;
2. **(3 punti)** lo pseudocodice;
3. **(2 punti)** il costo computazionale;
4. **(2 punti)** si valuti, infine, come varia il costo computazionale nel caso in cui  $T$  sia un albero rosso-nero.

### Esercizio 2 (10 punti)

1. **(2 punti)** Definire in linguaggio C il tipo *lista di coppie* `pairList` in cui ogni nodo della lista memorizza una coppia di numeri interi;
2. **(5 punti)** Scrivere una funzione C di prototipo `pairList zip(list L, list M)` che date due liste di interi  $L$  ed  $M$  crea una nuova lista di coppie

in cui gli elementi di  $L$  ed  $M$  vengono ordinatamente accoppiati. La lista risultato dovrà essere lunga come la più corta tra  $L$  ed  $M$ .

ESEMPIO: se  $L = \langle 1, 7, 4 \rangle$  ed  $M = \langle 2, 6, 11, 15, 2 \rangle$  il risultato sarà la lista di coppie  $\langle (1, 2), (7, 6), (4, 11) \rangle$ .

3. **(3 punti)** Scrivere una funzione `C` di prototipo `pairList pairs(list L)` che costruisce una lista con tutte le possibili coppie di elementi di  $L$ . Evitare, nel risultato, le coppie simmetriche.

ESEMPIO: se  $L = \langle 1, 7, 4 \rangle$  il risultato sarà la lista di coppie:

$\langle (1, 1), (7, 7), (4, 4), (1, 7), (7, 4), (1, 4) \rangle$ .

SUGG: una possibile soluzione potrebbe fare uso opportuno della funzione `zip` scritta al punto precedente (osservate anche l'esempio).

Potete assumere in questo esercizio di avere funzioni standard sul tipo `pairList`, analoghe alle corrispondenti funzioni sulle liste, come `pairList cons(int, int, pairList)`, `pairList append(pairList, pairList)` etc.

**Esercizio 3 (10 punti)** Dato un grafo  $G = (V, E)$  ed un nodo  $s \in V$ , si vogliono contare i nodi raggiungibili da  $s$ .

Si progetti un algoritmo che, ricevendo in input le liste di adiacenza di  $G$  ed un nodo  $s$ , restituisca il numero di tali nodi.

Di tale algoritmo:

1. **(3 punti)** si dia la descrizione a parole;
2. **(3 punti)** si scriva lo pseudo-codice (modificando eventuali funzioni studiate a lezione);
3. **(2 punti)** si calcoli il costo computazionale;

Infine, si testi il proprio algoritmo sul seguente input **(2 punti)**:

1: 2  $\rightarrow$  3  $\rightarrow$  4

2: 1  $\rightarrow$  3

3: 1  $\rightarrow$  2  $\rightarrow$  4  $\rightarrow$  6

4: 1  $\rightarrow$  3  $\rightarrow$  6

5: 7  $\rightarrow$  9

6: 3  $\rightarrow$  4

7: 5  $\rightarrow$  8  $\rightarrow$  9

8: 7  $\rightarrow$  9

9: 5  $\rightarrow$  7  $\rightarrow$  8

ed  $s = 1$ .