

# INFORMATICA GENERALE

## Secondo Esonero

GIANCARLO BONGIOVANNI, TIZIANA CALAMONERI, IVANO SALVO  
Sapienza Università di Roma

15 giugno 2017

**Esercizio 1 (10 punti)** Si consideri un albero binario di ricerca con  $n$  nodi ed altezza  $h$  memorizzato tramite record e puntatori. Il generico record della struttura contenga le seguenti informazioni:

- un campo chiave, `info`
- due campi puntatori ai figli, `fsin` e `fdes`
- un campo `in-order` in cui sia memorizzato il numero di visita in in-ordine del nodo.

Si progetti un algoritmo che, dato un puntatore `r` alla radice di una tale struttura dati, ed un intero `i`, restituisca il puntatore `punt` al nodo dell'albero la cui chiave, se tutte fossero ordinate in senso non decrescente, occuperebbe la posizione  $i$ .

Dell'algoritmo proposto:

- (**2 punti**) si dia la descrizione a parole;
- (**2+2 punti**) si scrivano due pseudocodici, uno ricorsivo ed uno iterativo, scrivendo per esteso eventuali funzioni studiate;
- (**1+1 punti**) si calcoli il costo computazionale di entrambe le versioni;
- (**2 punti**) si discuta se sia possibile apportare ulteriori modifiche alla struttura dati in modo che il costo computazionale sia  $O(\log n)$ .

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

```
lista s1SommaMax(lista L, int* s, int* l)
```

che data una lista di interi, restituisca in output una *nuova* lista contenente la sotto-lista di somma massima. Contemporaneamente carica i parametri `s` ed `l` con il valore della somma e la lunghezza della sotto-lista di somma massima.

NOTA BENE: È vietato nella soluzione di questo esercizio fare uso di vettori.

FACOLTATIVO: se riuscite, la vostra funzione dovrebbe fare un'unica scansione della lista di input e de-allocare eventuali risultati parziali (ad esempio potenziali liste risultato) non usati nella lista risultato finale (scrivere in questo caso, se necessario, una opportuna funzione `void deallocList(lista L)`).

ESEMPI: Se la lista di ingresso fosse  $L = \langle 31, -41, 59, 26, -53, 58, 97, -93, -23, 84 \rangle$ , la lista risultato dovrà essere  $\langle 59, 26, -53, 58, 97 \rangle$ , `s` caricato con 187 ed `l` con 5.

Nel caso in cui la lista contenga solo numeri negativi, la funzione torna la lista vuota, che ovviamente ha somma 0 e lunghezza 0.

[SUGGERIMENTO: applicare l'algoritmo lineare, in cui una sotto-lista è il potenziale prefisso della sotto-lista risultato di somma massima fino a che la sua somma è positiva, visto nel primo esonero per lo stesso problema sui vettori.]

**Esercizio 3 (10 punti)** Sia dato un grafo  $G = (V, E)$  (non necessariamente connesso) tramite le sue liste di adiacenza, e due nodi  $u, v \in V$ .

Si progetti un algoritmo che restituisca 1 se  $u$  è raggiungibile da  $v$  e 0 altrimenti.

Dell'algoritmo proposto:

- a. (**3 punti**) si dia la spiegazione a parole (evidenziando le differenze con un eventuale algoritmo di visita utilizzato come base);
- b. (**4 punti**) si scriva lo pseudocodice;
- c. (**2 punti**) si calcoli il costo computazionale;
- d. (**1 punto**) se si è usato un algoritmo di visita come base, si discuta la scelta fatta, se non si è usato, se ne spieghi il motivo.