

**008AA – ALGORITMICA E LABORATORIO**  
Primo Appello, 12 Luglio 2013

Cognome Nome:

N. Matricola:

Corso: A B

**Esercizio 1.** (9 punti)

Si consideri la seguente relazione di ricorrenza.

$$G(n) = \begin{cases} 1 & \text{per } n = 0 \\ 2 & \text{per } n = 1 \\ G(n-1) + 2G(n-2) & \text{per } n > 1 \end{cases}$$

1. Trovare i valori di  $G(n)$ , per  $n = 2, 3, 4, 5$ .
2. Dal punto precedente, intuire la soluzione esatta di  $G(n)$  e provarla per induzione.
3. Dato un intero  $n$ , descrivere un algoritmo efficiente per il calcolo di  $G(n)$  analizzandone la complessità in tempo.

**Esercizio 2.** (9 punti)

Sia  $T$  un albero binario di ricerca che implementa un dizionario. Sia  $v$  un nodo di  $T$ , e sia  $T_v$  il sottoalbero con radice  $v$ .

- Si progetti un algoritmo efficiente `countLE(v, k)` che, ricevuto in input un nodo  $v \in T$  e una chiave  $k$  restituisca il numero di elementi in  $T_v$  con chiave **minore o uguale a**  $k$ .
- Analizzare la complessità dell'algoritmo.

**Esercizio 3.** (6 punti)

Siano  $G_1 = (V_1, E_1)$  e  $G_2 = (V_2, E_2)$  due grafi e sia  $G = (V, E)$  il grafo dell'unione di  $G_1$  e  $G_2$ , cioè il grafo per cui  $V = V_1 \cup V_2$  e  $E = E_1 \cup E_2$ .

- Si progetti un algoritmo efficiente che costruisca il nuovo grafo  $G$ .
- Analizzare la complessità dell'algoritmo.

**Esercizio 4.** (6 punti)

Analizzare la complessità al caso peggio richiesta per inserire  $n$  elementi in un heap inizialmente vuoto, invocando ripetutamente la funzione `Enqueue`. Identificare, una sequenza di inserimenti che richiede tale complessità.