

## PRIMA PARTE

1. **[6 punti]** Nell'inserimento di una nuova entry in un (2,4)-Tree un nodo  $v$  può andare in *overflow*
  - (a) Cosa si intende per overflow di  $v$ ?
  - (b) Descrivere brevemente la procedura  $\text{Split}(v)$  usata per sanare la condizione di overflow analizzandone la complessità.
2. **[4 punti]** Spiegare cosa significa che Randomized QuickSort ha complessità  $O(n \log n)$  con alta probabilità.
3. **[6 punti]** Sapendo che un free tree con  $n$  vertici ha  $m = n - 1$  archi, dimostrare che un grafo connesso con  $n$  vertici ha  $m \geq n - 1$  archi.

## SECONDA PARTE

1. **[9 punti]** Sia  $T$  un  $(2,4)$ -Tree dove ogni nodo  $v \in T$  memorizza in una variabile  $v.size$  il numero di entry in  $T_v$  (incluse quelle in  $v$ ). Progettare un algoritmo non ricorsivo che conti quante entry in  $T$  hanno chiave  $\leq k$ , e analizzarne la complessità.
2. **[7 punti]** Sia  $G = (V, E)$  un grafo non diretto e connesso in cui ciascun vertice ha grado esattamente  $c$ , con  $c > 2$  costante intera. Si consideri l'esecuzione di  $BFS(G, s)$  a partire da un arbitrario vertice  $s \in V$ . Dimostrare per induzione su  $i$  che il livello  $L_i$  generato da  $BFS(G, s)$  contiene  $\leq c \cdot (c - 1)^{i-1}$  vertici, per ogni  $i \geq 0$ .

**TEMPO COMPLESSIVO A DISPOSIZIONE: 1.5 ore**