

PRIMA PARTE

1. **[3 punti]** Definire come problema computazionale il problema di trovare, dato un intero $n > 1$, un intero k con $1 < k < n$ tale che n è divisibile per k , se un tale k esiste.
2. **[3 punti]** Sia T un albero binario proprio con $n > 1$ nodi e radice r . Sia u il figlio sinistro di r . Qual è il nodo successore di u nella visita in postorder di T .
3. **[3 punti]** Si consideri una coda con priorità P , implementata tramite *heap*, inizialmente vuota. Si supponga di eseguire le seguenti operazioni nell'ordine dato: `P.insert(5,.)`, `P.insert(3,.)`, `P.insert(6,.)`, `P.insert(4,.)`, `P.insert(7,.)`, `P.insert(2,.)`, `P.removeMin()`. Disegnare lo heap risultante (come albero, riportando solo le chiavi) alla fine della sequenza di operazioni.
4. **[4 punti]** Dimostrare che un Multi-Way Search Tree contenente n entry ha $n + 1$ foglie.
5. **[3 punti]** È vero che applicando l'algoritmo probabilistico **Randomized QuickSort** a una sequenza S di n chiavi distinte ordinata in senso decrescente, vengono eseguite $\Theta(n^2)$ operazioni? Motivare la risposta.

SECONDA PARTE

1. [8 punti] Dato un albero binario proprio T , definiamo il *level numbering* dei nodi di T come segue:

- se v è radice di T , allora l'indice $\ell(v)$ di v nel level numbering è 1, cioè $\ell(v) = 1$;
- se v non è radice di T , sia u il padre di v : allora $\ell(v) = 2\ell(u)$ se v è figlio sinistro di u ; $\ell(v) = 2\ell(u) + 1$ se v è figlio destro di u .

Si vuole progettare un algoritmo ricorsivo che salvi in un campo $v.\text{indexLN}$ l'indice $\ell(v)$ di v nel level numbering dei nodi di T , per ogni nodo $v \in T$.

- (a) Si fornisca lo pseudocodice per l'algoritmo richiesto.
- (b) Si analizzi la complessità dell'algoritmo.

Per ricevere il massimo dei punti l'algoritmo deve avere complessità in tempo $O(n)$, dove n è il numero di nodi di T .

2. [8 punti] Sia $G = (V, E)$ un grafo con n vertici ed m archi.

- (a) Progettare un algoritmo che dato G ed un intero $k > 0$, per ogni nodo $v \in V$ salva in una variabile $v.\text{numNeighbors}$ il numero di nodi a distanza $\leq k$ da v .
- (b) Analizzare la complessità dell'algoritmo.