ESERCIZI (da temi d'esame)

Esercizio 1.
Si consideri un array S di n chiavi intere.

- Si dia il codice di un algoritmo che con un'unica scansione di S conti il numero r di chiavi distinte in S. Usare un dizionario D inizialmente vuoto (non interessa l'implementazione di D).
- Facendo l'assunzione che D sia implementato come un array ordinato, si analizzi la complessità in funzione di n e del numero r di chiavi distinte.

Esercizio 2.
Sia dato un array S di n interi di valore non limitato, ma che possono assumere solo $\lfloor \log n \rfloor$ valori distinti:

Esempio: $S = \langle 349; 12; 12; 102; 349; 12; 102; 102 \rangle$

Progettare un algoritmo di ordinamento che operi in tempo minore di $O(n \log n)$. Spiegare dettagliatamente l'analisi della complessità.

Esercizio 3.
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 4.
Un albero binario *proprio* è un albero in cui ogni nodo interno ha esattamente due figli. Sia T un albero binario proprio. Dato un nodo $v \in T$ si definisca `imbalance(`v`)` la differenza in valore assoluto tra il numero di foglie nei sottoalberi sinistro e destro di v (se v è una foglia `imbalance(`v`)=0`). Si definisca anche `imbalance(`T`)` $= max_{v \in T}$ `imbalance(`v`)`.

- Dimostrare un limite superiore all'imbalance di un albero binario proprio con n nodi, e descrivere un albero il cui imbalance raggiunge tale limite.
- Disegnare un albero binario proprio T in cui `imbalance(`T`)` = `imbalance(`v`)` e v non è la radice dellalbero.
- Si progetti un algoritmo efficiente per determinare `imbalance(`T`)` e analizzarne la complessità in tempo.

Esercizio 5.
Dato un albero binario T, una catena sinistra di T è una sequenza di r nodi ($r \geq 1$) legati uno all'altro dal puntatore sinistro. Una catena massimale sinistra è una catena che non è contenuta in nessun'altra catena sinistra. Detto L_T il numero di catene massimali sinistre

1. Indicare le catene massimali su un albero binario completamente bilanciato di altezza 4.
2. Dimostrare che se T è completamente bilanciato e ha $n = 2^k - 1$ nodi, $L_T = 2^{k-1}$.
3. Si definisca un algoritmo efficiente che calcoli il numero di catene massimali sinistre L_T per un qualsiasi albero binario T.