

## ESERCIZI (da temi d'esame)

**Esercizio 1.** Si consideri una tabella hash  $T$  di  $m$  posizioni, in cui le liste di trabocco per gli elementi con uguale indirizzo hash (valore hash associato alla chiave) sono organizzate come alberi AVL. Si faccia l'ipotesi che la tabella contenga  $n$  chiavi.

- Si dia il codice di un algoritmo che esegua la ricerca e l'eventuale inserzione nella struttura di una chiave  $k$ .
- Se ne determini la complessità nel caso pessimo.
- Se ne determini la complessità nel caso medio, tenendo conto che l'altezza di un albero AVL  $O(\log n)$  sia nel caso medio che nel caso pessimo.

**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.**

Progettare un algoritmo che verifichi se un albero binario è 1-bilanciato.

**Esercizio 4.**

Dato un albero binario, progettare un algoritmo efficiente per determinare il minimo valore di  $\Delta$  per cui l'albero risulti  $\Delta$ -bilanciato. Analizzare la complessità dell'algoritmo proposto.