

Prova intermedia di algoritmi e strutture dati

17 novembre 2025
Tempo a disposizione: 1h 30m

Esercizio 1

1. Progettare un algoritmo **ricorsivo** con segnatura:

$$\text{positivo}(\text{Albero } T) \rightarrow \text{Boolean}$$

che, preso in input un albero T con chiavi intere, restituisce il valore *true* se e solo se tutti i nodi dell'albero T hanno chiave con valore strettamente positivo. Se T è l'albero vuoto, l'algoritmo deve restituire il valore *true*.

2. Illustrare il costo temporale dell'algoritmo proposto.

Esercizio 2

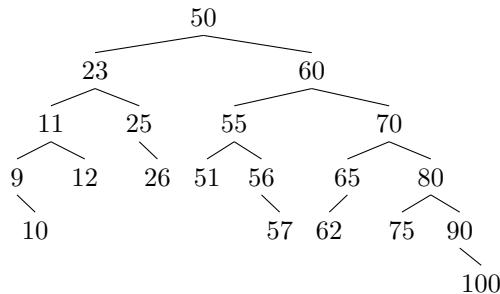
1. Risolvere la seguente equazione di ricorrenza: $T(n) = \begin{cases} 1, & \text{se } n = 1 \\ 2T\left(\frac{n}{4}\right) + \sqrt{n}, & \text{altrimenti} \end{cases}$

2. Risolvere la seguente equazione di ricorrenza usando il **metodo dell'iterazione**:

$$T(n) = \begin{cases} 1, & \text{se } n = 1 \\ 2T\left(\frac{n}{2}\right) + 2, & \text{altrimenti} \end{cases}$$

Esercizio 3

Dato l'albero AVL in figura:



1. Illustrare i passi per l'inserimento di un nodo con chiave 27, in modo da ottenere nuovamente un albero AVL.
2. Illustrare i passi per l'inserimento di un nodo con chiave 9.5, in modo da ottenere nuovamente un albero AVL.
3. Illustrare i passi per la cancellazione del nodo con chiave 11, in modo da ottenere nuovamente un albero AVL.

ATTENZIONE: tutte le operazioni dei punti precedenti sono tra loro indipendenti e devono essere eseguite partendo dall'albero in figura.

Esercizio 4

1. Mostrare lo psedocodice dell'algoritmo *MergeSort*.
2. Dimostrarne il costo temporale nel caso peggiore.
3. Mostrare l'esecuzione dell'algoritmo sul seguente array di input:

17	12	6	7	13	14	11	9
----	----	---	---	----	----	----	---