

# Esame di algoritmi e strutture dati

20 gennaio 2023

Modalità d'esame disponibili:

1. esonero (35%) + esercizi 4 e 5 (35%) + progetto (30%, obbligatorio tutti): 1 ora;
2. esercizi 1-4 (70%) + progetto (30%): 2 ore;

Tutti gli esercizi hanno stesso punteggio. Max: 33. Lode: > 30.

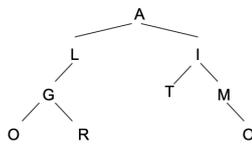
## Esercizio 1

1. Progettare un algoritmo (pseudocodice) con segnatura:

$quanti(\text{AlberoBinario } T, \text{Carattere } x) \rightarrow \text{Intero}$

che, preso in input un albero binario  $T$ , i cui nodi contengono valori di tipo carattere, ed un carattere  $x$ , restituisca il numero di occorrenze del carattere  $x$  in  $T$ .

Ad esempio, con il seguente albero di input,  $quanti(T, 'O')$  deve restituire il valore 2.



2. Illustrare, motivando la risposta, quale sia il costo temporale asintotico dell'algoritmo.

## Esercizio 2

1. Si illustri lo pseudocodice dell'algoritmo di ordinamento *MergeSort*.
2. Se ne dimostri la complessità temporale asintotica nel caso peggiore.
3. Si mostri l'esecuzione dell'algoritmo sul seguente array di input:  $A = (23, 19, 12, 16, 11, 7, 10, 2)$ .

## Esercizio 3

Si consideri un insieme di record con chiavi  $K = \{11, 31, 52, 41, 21, 64, 28, 46, 54, 12\}$  da inserire in una tabella hash di dimensione  $m = 10$ .

1. Assumendo che le chiavi siano tutte equiprobabili, definire una funzione hash  $h : \mathbb{N} \rightarrow [0, m - 1]$  per l'inserimento dei dati nella tabella, che sia semplicemente uniforme.
2. Mostrare i passi d'inserimento nella tabella hash dei record sopra mostrati, secondo l'ordine delle chiavi riportato in  $K$ , adottando uno schema con indirizzamento aperto.
3. Mostrare i passi d'inserimento nella tabella hash dei record sopra mostrati, secondo l'ordine delle chiavi riportato in  $K$ , adottando uno schema con liste di collisione.

## Esercizio 4

1. Si illustri un algoritmo per verificare se un grafo non orientato contenga una sola componente connessa.
2. Si dimostri il costo temporale asintotico dell'algoritmo.
3. Si mostri l'esecuzione dell'algoritmo sul grafo in figura:

