

# Esame di algoritmi e strutture dati

18 luglio 2024

Modalità d'esame:

1. prova intermedia (35%) + esercizi 3 e 4 (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.

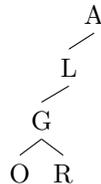
## Esercizio 1

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

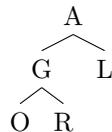
$$full(\text{AlberoBinario } T) \rightarrow \text{Boolean}$$

che, preso in input un Albero Binario  $T$ , restituisca *true* se e solo se l'albero è *pieno*. Un albero si dice *pieno* se ogni suo nodo ha esattamente zero o due figli.

Ad esempio, con il seguente albero di input,  $full(T)$  deve restituire *false*, in quanto i nodi con chiave 'A' ed 'L' hanno un solo figlio.



Con il seguente albero di input,  $full(T)$  deve restituire *true*.



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

## Esercizio 2

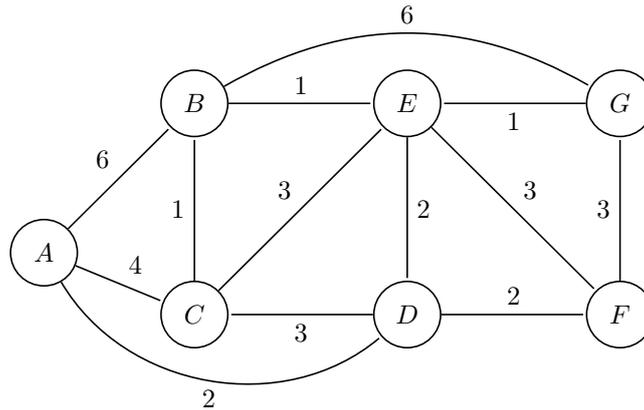
1. Si illustri lo pseudocodice dell'algoritmo di ordinamento *QuickSort*.
2. Se ne dimostri la complessità temporale nel caso peggiore (non caso medio).
3. Si mostri l'esecuzione dell'algoritmo sul seguente array di input:  $A = (53, 66, 41, 5, 29, 26, 16, 64)$ .

### Esercizio 3

1. Si mostrino i passi di costruzione dell'albero AVL ottenuto inserendo, nell'ordine riportato, le seguenti chiavi: 22, 11, 34, 68, 66, 29, 71, 52, 35, 45.
2. Si illustrino i passi da eseguire per cancellare, nell'ordine riportato, i nodi con chiave 34 e 29 ed ottenere, dopo ciascuna eliminazione, un albero AVL.

### Esercizio 4

Si consideri il grafo in figura:



1. Illustrare un possibile algoritmo per la costruzione di un minimo albero ricoprente del grafo.
2. Mostrare i passi d'esecuzione dell'algoritmo sul grafo.
3. Determinare il costo temporale dell'algoritmo presentato.