

# Esame di algoritmi e strutture dati

13 giugno 2018

Tempo a disposizione: 2 ore

## Esercizio 1

(6 punti)

1. Progettare un algoritmo (pseudocodice) con segnatura

$assente(\text{Albero } T, \text{Intero } v) \rightarrow \text{Boolean},$

che, preso in input un albero binario  $T$  i cui nodi contengono valori interi, restituisce  $true$  se e solo se nessun nodo dell'albero contiene l'intero  $v$ .

2. Determinare, fornendo una dimostrazione, il costo temporale dell'algoritmo.

## Esercizio 2

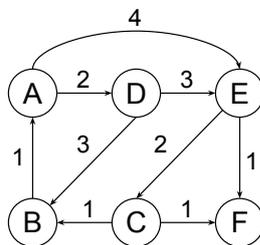
(6 punti)

- Mostrare lo pseudocodice dell'algoritmo *BubbleSort*.
- Determinare, fornendo una dimostrazione, la complessità temporale dell'algoritmo.

## Esercizio 3

(6 punti)

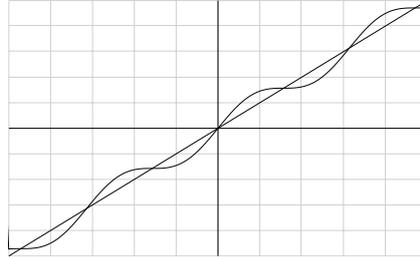
1. Mostrare lo pseudocodice dell'algoritmo di Dijkstra.
2. Mostrare i passi eseguiti dall'algoritmo sul seguente input, prendendo il nodo  $D$  come sorgente:



## Esercizio 4

(6 punti)

1. Si considerino le due funzioni  $f(x) = x$  e  $g(x) = \sin(x) + x$ , i cui grafici sono riportati in figura:



2. Dimostrare o confutare la validità di ciascuna delle seguenti affermazioni:

- $f(x) = \mathcal{O}(g(x))$
- $f(x) = \Omega(g(x))$
- $f(x) = \Theta(g(x))$

## Esercizio 5

(6 punti)

Determinare, fornendo una dimostrazione, il costo temporale  $T(n)$  del seguente algoritmo:

```
A(Array x) → Boolean{
  n = |x|;
  if(n == 0) return true;
  r1 = A(x[0, n/4 - 1]);
  r2 = A(x[(n/4), n/2 - 1]);
  r3 = A(x[(n/2), 3n/4 - 1]);
  r4 = A(x[(3n/4), n - 1]);
  return(r1 ∨ r2 ∨ r3 ∨ r4);
}
```

dove  $a[i, j]$  rappresenta il sotto-array di  $a$  contenente le componenti  $i, i + 1, \dots, j$ .

## Domanda per la lode

Descrivere, in al più due frasi, la differenza tra un albero binario di ricerca ed un albero AVL. Discutere, in al più due frasi se, e per quale motivo, i secondi siano preferibili ai primi.