

ESAME DI ALGORITMI E STRUTTURE DATI
A.A. 2015/2016 – APPELLO DEL 25/10/2016

TEMPO A DISPOSIZIONE: 120 MINUTI

ESERCIZIO 1:

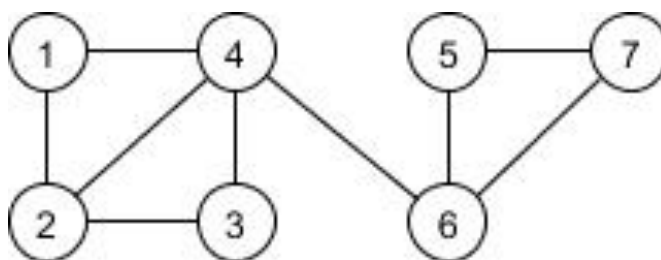
- i) Si scriva lo pseudocodice dell'algoritmo `BubbleSort` in grado di ordinare gli elementi contenuti in un array **A** di **n** interi preso in ingresso.
- ii) Descrivere i passi eseguiti dall'algoritmo `BubbleSort` sul seguente array in input.

0	1	2	3	4	5	6	7
9	7	6	3	5	8	2	4

- iii) Indicare la complessità dell'algoritmo `BubbleSort` nel caso peggiore.

ESERCIZIO 2:

- i) Si scriva lo pseudocodice dell'algoritmo `visitaDFS`(vertice v) \rightarrow albero che esegue la visita in profondità di un grafo non orientato.
- ii) Descrivere i passi dell'algoritmo di visita quando applicato al grafo orientato rappresentato in figura partendo dal nodo **1** e presentare l'albero DFS prodotto dalla visita.



ESERCIZIO 3:

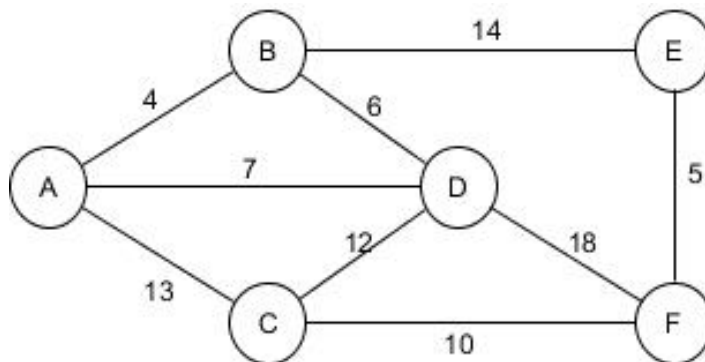
- i) Dare la definizione di albero AVL.
- ii) Costruire un albero AVL contenente le seguenti chiavi:

9 - 4 - 1 - 6 - 5 - 7 - 8 - 3

- iii) Indicare il costo dell'algoritmo di ricerca su alberi AVL `search(chiave k)` nel caso peggiore.

ESERCIZIO 4:

- i) Si scriva lo pseudocodice dell'algoritmo di Prim per il calcolo di un minimo albero ricoprente di un grafo non orientato, connesso e pesato sugli archi.
- ii) Calcolare il minimo albero ricoprente del grafo rappresentato in figura adottando l'algoritmo di Prim. Descrivere dettagliatamente i vari passi eseguiti dall'algoritmo.



ESERCIZIO 5:

Nel caso in cui non si possano evitare le collisioni in una tabella di hash, è necessario escogitare un metodo per risolverle. Indicare e descrivere due metodi per la risoluzione delle collisioni nelle tabelle di hash.

ESERCIZIO 6 (PER LA LODE):

La successione di Fibonacci segue la seguente regola di ricorrenza:

$$F_n = \begin{cases} 1 & \text{per } 0 < n < 3 \\ F_{n-1} + F_{n-2} & \text{per } n \geq 3 \end{cases}$$

Fornire un algoritmo ricorsivo "fibonacci(intero n) → intero" per il calcolo dell'*n*-esimo numero di Fibonacci F_n