

Cognome e Nome:
Numero di Matricola:

Docente:

Spazio riservato alla correzione

1	2	3	4	5	6	totale
/12	/6	/16	/16	/12	/28	/90

1. [12 punti] Si scriva la classe **MyPriorityQueue** che implementa l'interfaccia **PriorityQueue** dove le chiavi appartengono all'insieme {1,2,3}. I metodi di inserimento nella coda, di restituzione e cancellazione del minimo devono avere complessità di tempo $O(1)$.
2. [8 punti] Implementare il metodo **removeAtRank** della classe **ArrayVector** che implementa l'interfaccia **Vector** con un array.
3. [16 punti] Scrivere la funzione

UndirectedGraph removeDirection(DirectedGraph G)

che ricevuto in input un grafo direzionato $G(V,E)$, restituisce come output un grafo non direzionato $G'(V',E')$ tale $V'=V$ ed $E' = \{(u,v) : (u,v) \in E \text{ oppure } (v,u) \in E\}$.

4. [16 punti] Aggiungere alla classe **LinkeTree** che implementa l'interfaccia **Tree** il metodo

boolean equals(Tree T);

che restituisce true se e solo se l'albero su cui il metodo è invocato è identico a T. Non possono essere utilizzati i metodi **elements()** e **positons()**.

5. [12 punti] Scrivere la classe **DMap** che Implementa l'interfaccia **Map** con un **Dictionary**. La classe **DMap** ha un'unica variabile di istanza di tipo **Dictionary**.

6. [28 punti]

- [4 punti] Descrivere le quattro versioni del problema dei cammini minimi in un grafo orientato (SSSP, SDSP, SPSP, APSP).
- [9 punti] Mostrare, come un algoritmo utilizzato per risolvere il problema SSSP possa essere utilizzato per risolvere i problemi SDSP, SPSP, APSP.
- [15 punti] Mostrare i passi eseguiti dall'algoritmo di Dijkstra sul seguente grafo. Per ogni passo indicare l'insieme S, la frontiera ed il valore di ogni vertice.

