

Cognome e Nome:

Numero di Matricola:

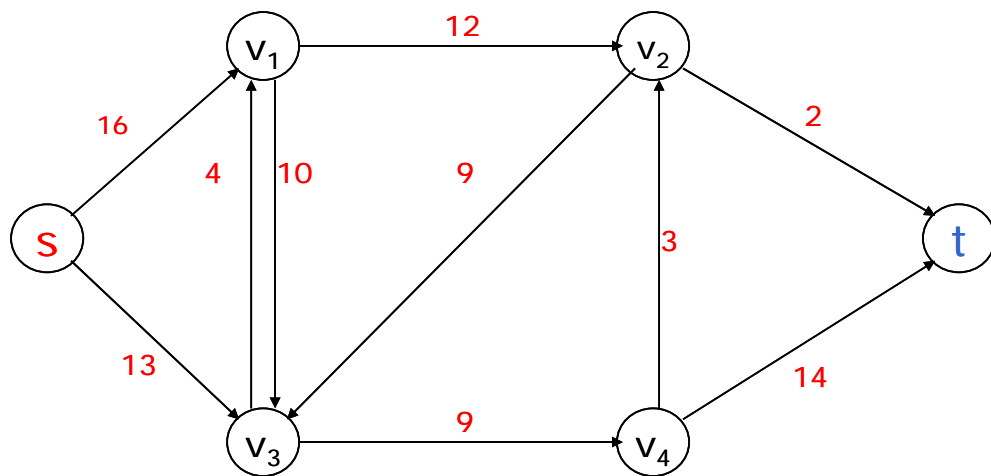
Docente:

**Spazio riservato alla correzione**

1	2	3	4	5	totale
/30	/20	/15	/12	/13	/90

Non usare altri fogli, usare solo lo spazio sottostante. Fogli differenti da questo non saranno presi in considerazione per la correzione.

1. Si consideri il seguente grafo G:



- a. Fornire le liste di adiacenza dei vertici di G. [5 punti]
- b. Fornire lo Shortest Path Tree ottenuto applicando l'algoritmo di Dijkstra a partire dal vertice s. [5 punti]
- c. Codificare l'algoritmo di Dijkstra in Java (scrivere una funzione Java che implementi l'algoritmo). Il codice deve far riferimento alle interfacce illustrate durante il corso. [15 punti]
- d. Analizzare la complessità dell'implementazione proposta dell'algoritmo di Dijkstra. [5 punti]

2. Aggiungere alla classe **LinkedTree** il metodo **distanza(TNode u, TNode v)** che restituisce la distanza (il numero di archi) tra u e v. Quale è la complessità della funzione proposta? Giustificare la risposta.

3. Aggiungere alla classe **VectorCompleteBinaryTree** che implementa l'interfaccia **CompleteBinaryTree** mediante un **Vector**, il metodo **Iterator removeLeaves()** che rimuove tutte le foglie dell'albero e restituisce in output un iteratore delle foglie cancellate. Per l'implementazione non è possibile utilizzare i metodi **elements()** e **positions()** di **CompleteBinaryTree**. Quale è la complessità del metodo proposto? Giustificare la risposta.

4. Scrivere la funzione Java

**Stack merge(Comparator c, Stack S1, Stack S2)**

che riceve come argomenti due stack i cui elementi sono disposti in ordine crescente dal bottom verso il top (es.  $S1 = \langle 2, 5, 6, 9 \rangle$ ,  $S2 = \langle 4, 5, 8 \rangle$ , dove 2 è il bottom di S1 e 4 è il bottom di S2). La funzione restituisce uno stack ordinato contenente gli elementi di entrambi gli stack (es. effettuando il merge degli stack S1 ed S2 si ottiene lo stack  $S = \langle 2, 4, 5, 5, 6, 8, 9 \rangle$ ). I confronti tra gli elementi devono essere effettuati utilizzando il comparatore passato come argomento. Quale è la complessità di tempo della funzione proposta? Giustificare la risposta.

5. Scrivere la funzione **Iterator elenca(Map M, List Keys)** che prende in input una mappa **M** e una lista **Keys** e cerca nella mappa **M** tutte le voci le cui chiavi sono contenute in **Keys** e restituisce un iteratore degli elementi associati alle chiavi trovate.