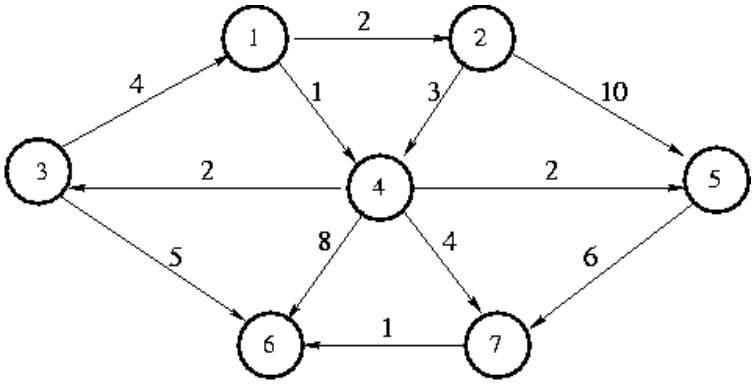
TRABAJO PRACTICO N° 8

ALGORITMOS DE GRAFOS

Nota1: para cada clase hacer un programa que pruebe cada uno de los métodos que provee.

Nota2: documentar cada clase y utilizar Javadoc para generar la misma.

1. Utilizar el TAD de grafos para crear el grafo que se muestra en la figura:



a) Listar todos los arcos que salen del vértice 1.
b) Listar todos los arcos que llegan al vértice 4.
c) Mostrar el vértice opuesto al vértice 5 con el arco que tiene un peso de 10.
d) Mostrar el arco que hay entre el vértice 6 y el vértice 7.
e) Mostrar los vértices del arco que tiene peso 8.

Responder a las mismas consultas considerando el mismo grafo no dirigido.

2. Utilizar un grafo para representar el plan de estudio de la carrera que está cursando. Los vértices representan las materias y los arcos sus correlativas. Mostrar las materias ordenadas de mayor a menor por la cantidad de correlativas que tienen.

3. Realizar una aplicación que dado un grafo cree una copia del mismo.

4. Agregar a la clase **GraphAlgorithms** el método ***sources*** (fuentes) que retorna una lista de vértices que solo tienen arcos que salen de él y el método ***sinks*** (sumideros) que retorna una lista de vértices que solo tienen arcos dirigidos hacia él.

5. Utilizar el método ***shortestPathLengths*** de la clase **GraphAlgorithms** (que implementa el algoritmo de *Dijkstra*) para calcular el camino más corto de un vértice a todos los vértices del grafo.

6. Utilizar el método ***spTree*** de la clase **GraphAlgorithms** para obtener el árbol de caminos más corto desde un vértice dado (raíz).

TRABAJO PRACTICO N° 8

ALGORITMOS DE GRAFOS

7. Dado un árbol de caminos más corto y un vértice cualquiera del árbol, indique todos los vértices intermedios para llegar a la raíz del mismo.