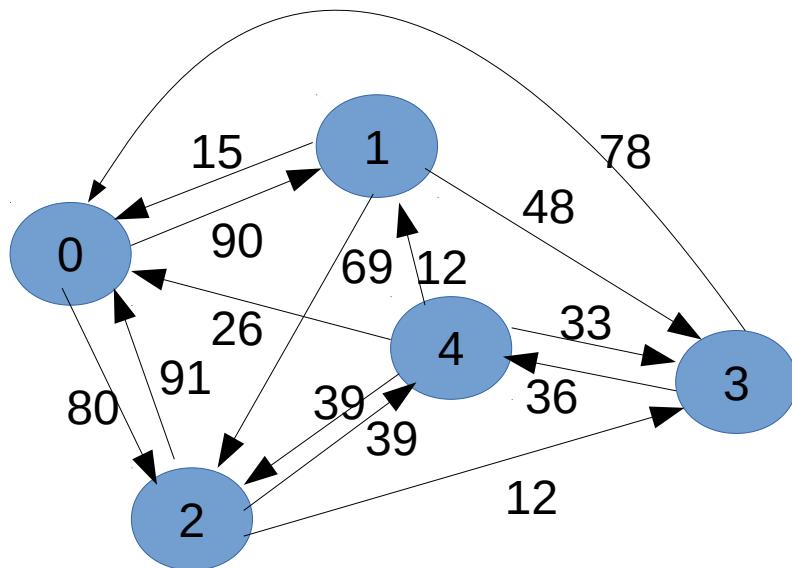


Tarea 6

Parte 1: Caminos de costos mínimos. Implementar los algoritmos de Dijkstra, Bellman Ford y Floyd Warschall para encontrar la matriz de costos mínimos de caminos entre todos los vértices fuente y todos los vértices destino de un grafo dirigido de n vértices con costos en los números naturales. La entrada del programa es un archivo de texto con la matriz de costos para ir de un vértice a otro. El valor -1 en la posición i,j significa que no hay un eje del vertice i al vertice j. Por ejemplo, la matriz:

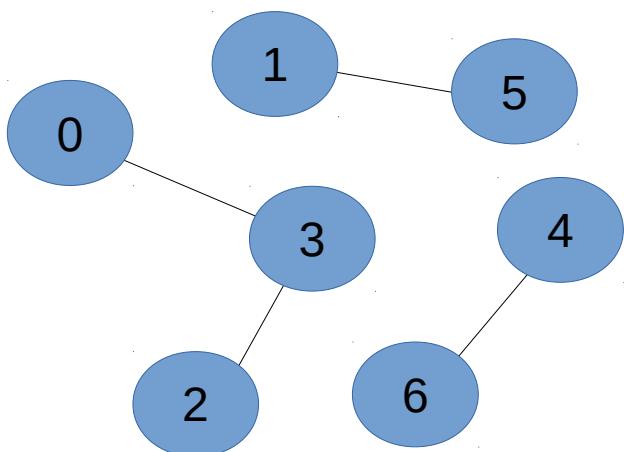
0	90	80	-1	-1
15	0	69	48	-1
91	-1	0	12	39
78	-1	-1	0	36
26	12	39	33	0

Representa al grafo:



Ejecutar cada uno de los algoritmos con las matrices adjuntas y comparar los tiempos de ejecución.

Parte 2: BFS para encontrar componentes conectados. Encontrar los componentes conectados en un grafo no dirigido. Dado un grafo no dirigido representado por una matriz simétrica, encontrar una partición de los vértices tal que para cada par de elementos dentro de un subconjunto de la partición exista un camino en el grafo. Por ejemplo, para el siguiente grafo:



	0	1	2	3	4	5	6	
0	-1	-1	-1	1	-1	-1	-1	-1
1	-1	-1	-1	-1	-1	-1	1	-1
2	-1	-1	-1	1	-1	-1	-1	-1
3	1	-1	1	-1	-1	-1	-1	-1
4	-1	-1	-1	-1	-1	-1	-1	1
5	-1	1	-1	-1	-1	-1	-1	-1
6	-1	-1	-1	-1	1	-1	-1	-1

La respuesta debe ser $\{\{0,2,3\}, \{1,5\}, \{4,6\}\}$

Parte 3: DFS. Determinar si existe al menos un ciclo en un grafo dirigido representado por una matriz como la descrita en la parte 1. Para el grafo del ejemplo, de la parte 1 un ejemplo de ciclo es $<0,2,3,4,0>$. En caso de que el grafo no tenga ciclos, se debe reportar un orden topológico para el grafo.

Entrega: Para las 3 partes, entregar un solo proyecto en Eclipse que incluya al menos 3 programas (clases con main) que resuelvan respectivamente cada uno de los 3 problemas. Incluir un README.txt que indique cómo se debe ejecutar cada programa (formato del archivo de entrada y descripción de la salida).