

Hoja de trabajo No. 9

Realizar: Programa de implementación de grafos para algoritmo de Floyd.

Realizarse: en parejas.

Objetivos:

- Implementación de un grafo en Java.
- Implementación del algoritmo de Floyd (camino más corto entre cualquier par de vértices del grafo) en Java.
- Usar los resultados del algoritmo de Floyd para calcular el centro del grafo, en Java.
- Opcional: usar el módulo de Python NetworkX para el algoritmo de Floyd.

Programas a realizar:

- Interfaz genérica para grafos. Deberá tener por lo menos los métodos para agregar nodos (o vértices) y arcos. Basta con implementar grafos con dirección.
- Se recomienda (pero no es obligatorio) usar la implementación de un grafo por medio de una Matriz de Adyacencia. (Solo para grafos con dirección).
- Implementar el algoritmo de Floyd para encontrar la ruta entre cualquier par de nodos del grafo. El resultado del camino más corto debe indicar el peso de la ruta y los nodos por los que pasa la ruta más corta entre cualquier par de nodos.
- El programa permite leer el grafo desde un archivo y posteriormente permitir agregar o eliminar arcos desde la pantalla.
- OPCIONAL:** hacer el programa equivalente en Python, usando el módulo NetworkX, para calcular la distancia más corta con Floyd y el centro del grafo.

Problema:

Usted es el encargado de logística de uno de los partidos políticos de la República de Guatemala. Debe planificar los viajes de proselitismo para la campaña electoral. Para apoyo de sus labores usted mantiene un grafo (con dirección), cuyos nodos son las ciudades principales de Guatemala y los arcos son la distancia, expresada en KM, que existe entre un par de ciudades.

Debido a los desastres naturales y bloqueos en carreteras, que afectan nuestra red vial, usted se ve en la necesidad de actualizar constantemente su grafo, para indicar que no existe conexión directa entre algún par de ciudades. Al efectuar esa modificación, usted necesita que el programa le indique nuevamente la distancia más corta y la ruta a tomar para viajar entre cualquier par de ciudades. Además el programa le indica cual ciudad es el centro del grafo para que usted considere colocar allí las oficinas centrales de la campaña, ya que no desea que estas oficinas estén solamente en la Ciudad Capital.

Su programa lee un archivo txt que representa el grafo. Cada línea tiene, el nombre de la ciudad1, nombre de la ciudad2 y la distancia en KM que hay desde la ciudad1 a la ciudad2 (recuerde que es un grafo con direcciones).

Ciudad1 Ciudad2 KM

Luego de leer el archivo, se construye el grafo y se aplica el algoritmo de Floyd para calcular la distancia más corta entre cualquier par de ciudades y cuál es la ciudad que queda en el centro del grafo. Se debe mostrar la matriz de adyacencia que representa el grafo.

Las opciones de su programa, en una repetición, son:

- Preguntar el nombre de la ciudad origen y ciudad destino. El programa mostrará el valor de la ruta más corta y los nombres de las ciudades intermedias por donde pasa esa ruta.
- Indicar el nombre de la ciudad que queda en el centro del grafo.
- Modificar el grafo indicando algunos de los siguientes datos: a) hay interrupción de tráfico entre un par de ciudades o b) se establece una conexión entre ciudad1 y ciudad2 con valor de x KM de distancia. Al fin de estas modificaciones se recalculan las rutas más cortas y el nuevo centro del grafo.
- Finalizar el programa.

Tareas:

- Construir las interfaces y clases necesarias para implementar un grafo.
- Construir el programa que utilice la implementación del grafo y realice el algoritmo de Floyd. Con base a los resultados del algoritmo de Floyd se calcula el centro del grafo.
- Usar JUnit para probar por lo menos los métodos para agregar nodos, agregar y eliminar arcos y el algoritmo de Floyd.
- Usar un sistema de control de versiones, como Git para registrar el trabajo realizado. Ambos miembros de la pareja deben reflejar su contribución en el Git.
- Hacer el diagrama UML de clases que se emplean en su programa.
- Colocar el programa Java y el programa Python en la plataforma del curso. Incluir el diagrama UML de clases, las pruebas unitarias y el enlace al control de versiones del programa.
- Opcionalmente: también puede elaborar el mismo programa empleando el módulo de Python: NetworkX. Ese módulo incluye el algoritmo de Floyd-Warshall. Este también debe estar en el sistema de control de versiones.

Calificación: su programa debe funcionar para ser calificado.

Aspecto	Puntos
Implementación del Grafo en Java	20
Implementación del algoritmo de Floyd en Java	15
Implementación del cálculo del centro del grafo.	15
Programa principal que usa el grafo y el algoritmo de Floyd	20
Diagrama UML de clases	10
Control de versiones	10
Pruebas unitarias de los métodos del grafo con JUnit	10
(Opcionalmente) Implementación del programa usando Networkx	50
TOTAL:	150

Referencias:

- El algoritmo de Floyd lo puede encontrar en la plataforma del curso.
- Adjunto a esta hoja se encuentra el algoritmo para calcular el centro de un grafo.
- El módulo Python, NetworkX está en el sitio: <http://networkx.github.io/> (el cual incluye tutoriales y la referencia a los algoritmos que proporciona el módulo).