Estructura de Datos 20/21

Práctica 6

Algoritmo de Dijkstra

Objetivos

- Implementar el algoritmo de Dijkstra utilizando una tabla de dispersión

Introducción

En esta práctica implementaremos el algoritmo de Dijkstra utilizando una tabla de dispersión.

El algoritmo de Dijkstra nos permite calcular el camino mínimo entre dos nodos de un grafo ponderado (coste positivo). Por ejemplo, ponemos pensar en una red de carreteras como un grafo ponderado. Los nodos del grafo son las ciudades. Y las aristas, con sus costes, son las carreteras entre las ciudades (y los kilómetros entre ellas).¿Cuál es el camino más corto entre la ciudad A y B? El algoritmo de Dijkstra nos permite contestar a esta pregunta.

El algoritmo de Dijkstra lo tenéis documentado en el PDF disponible en el campus virtual. Os recomiendo que antes de hacer la práctica, os familiaricéis con el algoritmo para que entendáis su funcionamiento. En esta práctica no nos centraremos en explicar el algoritmo. Nos centraremos en los aspectos de programación y de estructuras de datos para implementarlo.

Implementación del algoritmo y funcionamiento

1. El grafo de entrada al algoritmo estará definido en un fichero TXT como el que se adjunta en la práctica. Cada línea tiene 3 elementos, separados por un tabulador:

nodo_origen nodo_destino peso

2. El programa, en su método main, lee el fichero y crea el GRAFO en el que se aplicará Dijkstra. El código para leer el fichero y crear el grafo lo tenéis en el campus virtual (fichero **Dijkstra.java**).

3. La implementación del algoritmo se realizará en el método ***dijkstra*** del fichero **Graph.java**. El resto de métodos no hace falta que los modifiquéis. Para la implementación, es un requisito de la práctica utilizar una tabla de dispersión – ver apuntes de teoría.

4. Sobre el grafo, fijaros que se crea añadiendo Aristas (en Inglés, Edge). Los nodos del grafo, y de las aristas, están guardados en una tabla de dispersión. Dependiendo de si ya existe el nodo o no en la tabla, se añade. Esta funcionalidad la tenéis que implementar en el método ***getVertex*** del fichero **Graph.java**

5. La clase **Edge** ya la tenéis implementada y está disponible en el campus virtual.

6. La clase **Vertex** la tenéis que implementar. Diremos que un Vertex (vértice o nodo) tiene un nombre, una lista de aristas a los nodos adyacentes, la distancia o peso, y un vértice destino – esto lo necesitamos para crear el camino mientras se ejecuta Dijkstra.

Tareas a realizar

1. Implementar la clase Vertex.java

2. Implementar el método dijkstra y getVertex de la clase Graph.java
3. Para comprobar el funcionamiento de vuestra implementación del algoritmo, podéis utilizar el PDF que se adjunta, y el grafo ejemplo que se proporciona en el campus virtual.

Entrega

Fecha: ver tarea correspondiente en el campus virtual

Material: Un fichero ZIP con el proyecto IntelliJ IDEA y un documento RTF con la estrategia / algoritmo implementado.

Criterios (generales) de evaluación

- Práctica no entregada o entregada fuera de plazo = 0
- Entrega parcial, <= 3
- Entrega completa, entre 4 y 10. Se valorará el diseño y claridad del código, comentarios, etc.