

Laboratorio de la semana 8: Algoritmo de Dijkstra

En este taller, usted debe implementar el algoritmo de Dijkstra para búsqueda de caminos de costo mínimo. Este algoritmo permite encontrar los caminos de costo mínimo desde un vértice fuente hasta todos los vértices alcanzables, en un grafo dirigido sin costos negativos.

Para la implementación, debe crear una clase en Java llamada *Dijkstra*, basada en la clase *Digraph*. El esqueleto de la clase que usted debe realizar se presenta a continuación:

```
import java.util.*;

public class Dijkstra {

    public Dijkstra(Digraph G, int s) {
        caminosMinimo(G, s);
    }

    public void caminosMinimo(Digraph G, int s) {
    }

    public Integer[] costs() {
    }

    public Integer[] arcsVisited() {
    }
}
```

El método *caminosMinimo* aplica el algoritmo de Dijkstra para encontrar los caminos de costo mínimo en el grafo G a partir del vértice s .

El método *costs* devuelve un arreglo cuya i -ésima posición representa el costo del camino mínimo encontrado al vértice i .

El método *arcsVisited* devuelve un arreglo cuya i -ésima posición indica el vértice anterior en el camino de costo mínimo al vértice i . A través de los apuntadores debe ser posible reconstruir el camino de costo mínimo a cada uno de los vértices.

Su implementación debe ser $O(\max(V,E) \log_2 V)$, donde V es el número de vértices y E el número de aristas. Esto implica que debe utilizar una cola de prioridades para la implementación.

Pruebas

Para correr su programa, deberá hacer una clase clienteDijkstra con una rutina main que cree un objeto de clase Dijkstra.

Su programa se podrá llamar con el siguiente comando:

```
java clienteDijkstra <nombre de archivo> <vértice fuente>
```

Donde *nombre de archivo* es el nombre del archivo que contiene el grafo dirigido en el formato utilizado en el laboratorio, y *vértice fuente* es el número del vértice fuente con el que se correrá el algoritmo.

A continuación se muestra un ejemplo de la entrada:

```
8
14
0 1 20
0 3 80
0 6 90
1 5 10
2 5 50
2 7 20
2 3 10
3 2 10
3 6 20
4 1 50
4 6 30
5 2 10
5 3 40
6 0 20
```

Su programa debe imprimir, para cada vértice alcanzable, el costo del camino mínimo hacia él, así como el vértice antecesor en el caminar costo mínimo. Si el vértice no es alcanzable debe imprimir NA. Esta es la salida correspondiente al ejemplo de entrada, comenzado el algoritmo en el vértice 0.

0	0	-
1	20	0
2	40	5
3	50	2
4	NA	
5	30	1
6	70	3
7	60	2

Condiciones de entrega

Debe entregar el día 1 de junio de 2016 antes de las 2:30 am, un archivo comprimido llamado LabSem8-X-Y.tar.gz, con todos los códigos Java y el archivo Makefile. Las letras X y Y del archivo comprimido son los números de carné de los integrantes del equipo.