UNIVERSIDAD NACIONAL DE SAN AGUSTIN

FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

GUÍA DE LABORATORIO

ESTRUCTURA DE DATOS Y ALGORITMOS

2021A

EDITH PAMELA RIVERO TUPAC Ing. Informática y de Sistemas Mgt. Seguridad Informática

Informe de Actividades:

1. Crear un repositorio en GitHub, donde incluyan la resolucion de los ejercicios propuestos y el informe.

   - https://github.com/Braulio20182313/lab09-EDA-Grafos.git

2. Implementar el cogido de Grafo cuya representacion sea realizada mediante LISTA DE ADYACENCIA. (3 puntos)

   - **Lista de Adyacencia:**

```
Graph g = new Graph(10);
g.addEdge(1, 3, 5);
g.addEdge(1, 2, 14);
g.addEdge(0, 2, 10);
g.addEdge(0, 5, 15);
g.addEdge(2, 2, 10);
g.addEdge(9, 3, 16);
```

```
0=>[(5,15), (2,10)]
1=>[(2,14), (3,5)]
2=>[(2,10)]
3=>[]
4=>[]
5=>[]
6=>[]
7=>[]
8=>[]
9=>[(3,16)]

true
```

3. Implementar BSF, DFS y Dijkstra con sus respectivos casos de prueba. (5 puntos)

   - **BSF y DFS:**

     Datos: 5 nodos, 4 enlaces -> los enlaces se leen Ejem. de nodo 0 a nodo 1, separados por pares

```
ingresar numero de vertices y aristas
5
4
ingresar 4 aristas
0
1
1
2
2
3
3
4
Ingresa origen y destino:
0
3
```

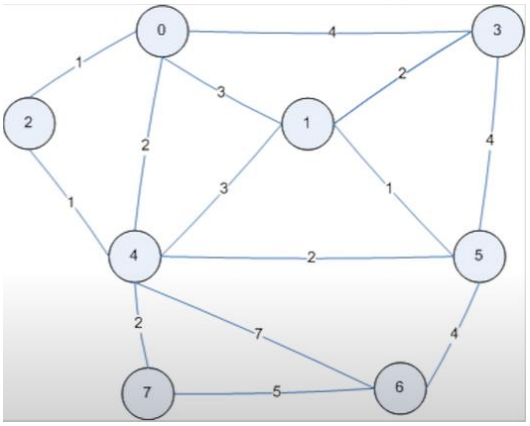
Resultado:

```
BSF--------------------
3->2->1->Distancia minima es: 3
DSF--------------------
Posible ->true
```

- Algoritmo Dijkstra:
Datos:

```
Edge[] edges = {
        new Edge(0,2,1),
        new Edge(0,3,4),
        new Edge(0,4,2),
        new Edge(0,1,3),
        new Edge(1,3,2),
        new Edge(1,4,3),
        new Edge(1,5,1),
        new Edge(2,4,1),
        new Edge(3,5,4),
        new Edge(4,5,3),
        new Edge(4,6,7),
        new Edge(4,7,2),
        new Edge(5,6,4),
        new Edge(6,7,5)
};
```

Grafo:

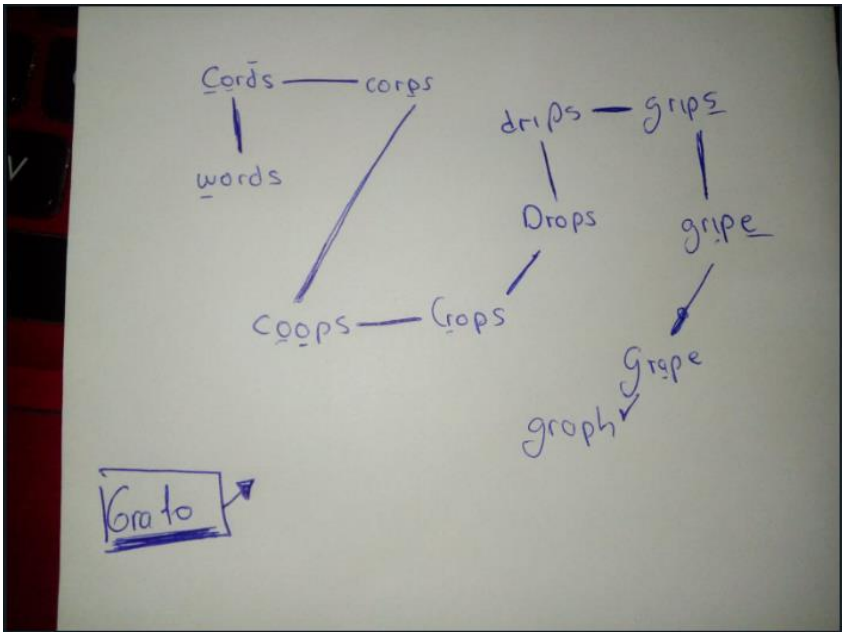


Resultado:

```
Numero de nodos = 8
 Numero de aristas = 14
La distancia mascorta del nodo 0 al nodo 0 es: 0
La distancia mascorta del nodo 0 al nodo 1 es: 3
La distancia mascorta del nodo 0 al nodo 2 es: 1
La distancia mascorta del nodo 0 al nodo 3 es: 4
La distancia mascorta del nodo 0 al nodo 4 es: 2
La distancia mascorta del nodo 0 al nodo 5 es: 4
La distancia mascorta del nodo 0 al nodo 6 es: 8
La distancia mascorta del nodo 0 al nodo 7 es: 4
```

4. Solucionar el siguiente ejercicio: (5 puntos)
   El grafo de palabras se define de la siguiente manera: cada vértice es una palabra en el idioma Inglés y dos palabras son adyacentes si difieren exactamente en una posición. Por ejemplo, las **cords** y los **corps** son adyacentes, mientras que los **corps** y **crops** no lo son.

   a) Dibuje el grafo definido por las siguientes palabras: words cords corps coops crops drops drips grips gripe grape graph



   b) Mostrar la lista de adyacencia del grafo.

Words	Words	
Cords	Words	Corps
Corps	Cords	Coops
Coops	Cords	Crops
Crops	Coops	Drops
Drops	Crops	Drips
Drips	Drops	Grips
Grips	Drips	Gripe
Gripe	Grips	Grape
Grape	Gripe	Graph
Graph	Grape	

5. Realizar un metodo en la clase Grafo. Este metodo permitira saber si un grafo esta incluido en otro. Los parametros de entrada son 2 grafos y la salida del metodo es true si hay inclusion y false el caso contrario. (4 puntos)

VI

CUESTIONARIO

1. ¿Cuantas variantes del algoritmo de Dijkstra hay y cuál es la diferencia entre ellas? (1 puntos)

Con cola de prioridad y sin cola de prioridad:

La principal diferencia es que en la variación que no posee cola de prioridad se usa un vector que guarda la distancia del nodo de salida al resto, inicializando el vector con distancias iniciales en valor booleano,

2. Invetigue sobre los ALGORITMOS DE CAMINOS MINIMOS e indique, ¿Qué similitudes encuentra, qué diferencias, en qué casos utilizar y porque? (2 puntos)

Se presenta los diferentes algoritmos y sus principals caracteristicas para hacer la diferenciación:

Bellman-Ford: resuelve el problema de los caminos más cortos desde un origen permitiendo que la ponderación de los nodos sea negativa.
Algoritmo de Búsqueda A*: resuelve el problema de los caminos más cortos entre un par de nodos usando la heurística para agilizar la búsqueda.
Algoritmo de Floyd-Warshall: resuelve el problema de los caminos más cortos entre todos los nodos.
Algoritmo de Johnson: resuelve el problema de los caminos más cortos entre todos los nodos y puede ser más rápido que el de Floyd-Warshall en grafos de baja densidad.
Algoritmo de Viterbi: resuelve el problema del camino estocástico más corto con un peso probabilístico adicional en cada nodo.
Dijkstra: resuelve el problema de los caminos más cortos desde un único nodo origen hasta todos los otros nodos del grafo (aunque aplicando una regla de repetición del algoritmo, se puede automatizar la resolución del problema desde todos los nodos de origen hasta todos los nodos del grafo).

VIII

BIBLIOGRAFÍA

[1] Weiss M., Data Structures & Problem Solving Using Java, 2010, Addison-Wesley.
[2] Escuela de Pedagogía en Educación Matemática, Marcelino Álvarez, et.al., http://repobib.ubiobio.cl/jspui/bitstream/123456789/1953/3/Alvarez_Nunez_Marcelino.pdf
[3] http://www.oia.unsam.edu.ar/wp-content/uploads/2017/11/dijkstra-prim.pdf