Informe Laboratorio Grafos

ELABORADO POR: JUAN CAMILO ROJAS CATRO Y FRANCISCO JAVIER ROJAS

PRESENTADO A: SESBASTIAN CAMILO MARTINEZ REYES

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

AYED

PROGRAMA DE INGENIERÍA DE SISTEMAS

BOGOTÁ D. C.

5 DE NOVIEMBRE DE 2020

**Ruta Fácil**

**1. Contexto:**

A veces las tiendas tienen que hacer varios domicilios y esto puede llegar a ser problemático pues se pueden complicar haciendo las rutas, se confunden o no saben que ruta tomar, omitiendo clientes, perdiendo tiempo en la entrega gastando combustible y de cierta forma empeorando el servicio, para resolver este problema simplemente utilizaran el programa que tiene como objetivo crear la ruta más corta siempre y cuando los lugares de entregan tengan vías de acceso, también tiene una funcionalidad secundaria donde una persona dando las conexiones existentes en una mapa obtiene el mejor lugar para colocar la centrar de abastos, todo esto ingresando su ubicación (tienda) y las ubicaciones de los clientes para la primera función y solo ingresando las ubicaciones para la segunda función.

**2. Algoritmo:**

* **Estrategia:**

A medida que se leen los datos de la entrada se guardan en una matriz de adyacencia, luego por el recorrido BFS se encuentra la ruta mas corta desde la tienda hasta pasar por todos los clientes.

* **Estructuras de datos:**

Se utilizarán diccionarios, listas unidimensionales y listas bidimensionales.

* **Especificaciones:**

El grafo es mixto, se toma como punto de partida la tienda, no existe prioridad entre los clientes y para simplificar el problema 0 es la tienda y los n clientes son números de 1 hasta n.

* **Entrada:**

La entrada consiste en primero un numero único por línea que decide la funcionalidad (encontrar la ruta o encontrar de domicilios o encontrar la ruta fácil hasta la central), luego líneas consecutivas donde cada línea tendrá dos números, los cuales pueden ser clientes y/o la tienda, cada línea representa una conexión por vías entre estos. Por último, dependiendo de lo funcionalidad escogida le preguntará por

* **Salida:**

La salida dependerá de la funcionalidad escogida, eso debe tenerlo en cuenta el usuario, sin embargo, en ambos casos la salida es la ruta más corta.

**3. Casos de prueba:**

* **Entrada:**

1

4 5

0 2

2 1

2 3

1 2

3 0

2

4 5

0 2

2 1

2 3

1 2

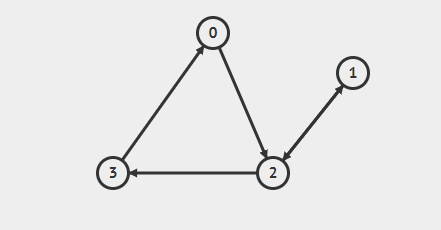
3 0

* **Salida:**

[0, 1, 2, 1, 2] ['tienda', 'Cliente2', 'Cliente1', 'Cliente3', 'Cliente2', 'Cliente1', 'Cliente3']

las centrales de abastos pueden ser [0]

* **Grafo:**



**¡Para una mejor compresión sobre la entrada y la salida se adjunta un archivo llamado casos-prueba-simulacion.txt!**

**4. Análisis:**

* **Temporal:**

Tanto el recorrido BFS como la función principal tiene una complejidad de O(n^2)