OBSERVACIONES DE LA PRÁCTICA 9

Integrantes del grupo

Grupo 11: **Federico Melo Barrero**

202021525

f.melo@uniandes.edu.co

Juan Camilo Prieto Avella

201814815

Jc.prietoa@uniandes.edu.co

Preguntas de análisis

a) ¿Qué instrucción se usa para cambiar el límite de recursión de Python?

Se importa la librería **sys**, que es propia de Python, y se utiliza la función **setrecursionlimit()** de dicha librería con el límite de recursión que se quiere establecer como parámetro.

e.g: **sys.setrecursionlimit(2 ** 20)**.

b) ¿Por qué considera que se debe hacer este cambio?

El recorrido Depth First Search (DFS) sobre grafos, que es la base de numerosos algoritmos que pueden usarse para recorrer el grafo y dar respuesta a las consultas del usuario, funciona de manera recursiva. Si el número de recursiones efectuadas por DFS supera el límite de recursión de Python, el programa detendrá su ejecución y presentará un **RecursionError.** Para evitar esto, es necesario modificar el límite de recursión preestablecido de Python. El grafo generado con base en el archivo **bus_route_14000.csv** tiene más de 13000 vértices y 32000 arcos, por lo que el número de recursiones realizadas al recorrer un grafo como ese usando Depth First Search (DFS) o algoritmos similares está muy por encima del límite de recursión predeterminado de Python (véase la respuesta a la pregunta c). Ergo se justifica hacer el cambio en el límite de recursión.

c) ¿Cuál es el valor inicial que tiene Python cómo límite de recursión?

El valor establecido como límite de recursión predeterminado en Python es 1000.

```
(base) fmb@fmb-pc:~$ python
Python 3.8.8 (default, Apr 13 2021, 19:58:26)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> print(sys.getrecursionlimit())
1000
>>>
```

d) ¿Qué relación creen que existe entre el número de vértices, arcos y el tiempo que toma la operación 4?

Los resultados de las pruebas de ejecución de la opción 4 con la estación **75009-10** y un límite de recursión de 1048576 se muestran en la Tabla 1.

Tabla 1. Pruebas de ejecución de la opción 4 con los diferentes archivos bus_routes

Número de líneas del archivo csv	Número de vértices	Número de arcos	Tiempo de ejecución de la opción 4
50	74	73	0.068190356 s
150	146	146	0.133660883 s
300	295	382	0.160557964 s
1000	984	1633	0.371578635 s
2000	1954	3560	1.214320163 s
3000	2922	5773	1.831 s
7000	6829	15334	8.504 s
10000	9767	22758	16.523 s
14000	13535	32270	24.642 s

A partir de las pruebas realizadas se hace evidente que entre menor es la cantidad de vértices y arcos, menor es el tiempo que demora en ejecutarse la opción 4.

Tabla 2. Pruebas de ejecución de la opción 6 con los diferentes archivos bus_routes

Número de líneas del archivo csv	Número de vértices	Número de arcos	Tiempo de ejecución de la opción 6
50	74	73	0.0034 s
150	146	146	0.0024 s
300	295	382	0.0030 s
1000	984	1633	0.0025 s
2000	1954	3560	0.0036 s
3000	2922	5773	0.0027 s
7000	6829	15334	0.0030 s
10000	9767	22758	0.0045 s
14000	13535	32270	0.0058 s

e) ¿El grafo definido es denso o disperso?, ¿El grafo es dirigido o no dirigido?, ¿El grafo está fuertemente conectado?

El grafo es dirigido porque los buses tienen una dirección específica entre las estaciones. Es decir, si un bus que va de una estación particular a alguna otra no tiene porque existir una ruta que se devuelva de dicha estación a la incial.

El grafo cuenta con v = 13.535 vértices y e = 32.270 arcos. Su densidad está dada por el número de arcos del grafo sobre el número total de arcos que puede tener, $v(v-1) = 13.535(13.534) = 1.832 \times 10^8$. Es decir, la densidad del grafo es

$$densidad = e / (v(v-1))$$

$$densidad = 32.270 / (13.535(13.534))$$

$$densidad = 1.762 \times 10^{-4}.$$

Tomando en cuenta que si un grafo cuenta con una densidad mayor o igual a 0.75 se considera denso y si cuenta con una densidad menor a 0.3 se considera disperso, se evidencia que el grafo en cuestión es disperso.

El grafo no está fuertemente conectado puesto que cuenta con múltiples componentes conectados. Si el grafo fuese fuertemente conectado únicamente tendría un componente conectado.

f) ¿Cuál es el tamaño inicial del grafo?

El grafo se inicia con un tamaño inicial de 14000, que es el tamaño aproximado del archivo más grande de rutas de buses, **bus_routes_14000.csv**.

g) ¿Cuál es la Estructura de datos utilizada?

Para definir el grafo se utiliza una lista de adyacencias **'ADJ_LIST'** que tiene tamaño n*n siendo n los vertices del grafo y los valores dentro de la matriz determinan en peso que hay entre un vértice y el otro o 0 si no hay una arista entre estos vértices.

h) ¿Cuál es la función de comparación utilizada?

En la construcción del grafo, se hace uso de la función de comparación **compareStopIds**, la cual se usa como función de comparacion de las dos estructuras de datos (una tabla de hash con linear probing y un grafo) y esta se usa para identificar las paradas de bus únicas e incluir por dentro todos los buses que pasan por ahí.