

## OBSERVACIONES DEL LA PRACTICA

1- Jerónimo Vargas Rendon, 202113305, [j.vargar2@uniandes.edu.co](mailto:j.vargar2@uniandes.edu.co)

2- Maicol Antonio Rincón, 202027329, [m.antonio@uniandes.edu.co](mailto:m.antonio@uniandes.edu.co)

### Preguntas de análisis

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

```
156 if __name__ == "__main__":  
157     threading.stack_size(67108864) # 64MB stack  
158     sys.setrecursionlimit(2 ** 20)  
159     thread = threading.Thread(target=thread_cycle)  
160     thread.start()
```

Este condicional es el que cambia el límite de recursión antes de ejecutar la aplicación, en especial el método `setrecursionlimit()` del módulo `sys`.

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

Este cambio es necesario, ya que en el momento de trabajar con grafos muchos de sus funciones para manipularlas son recursivas. Un caso en especial es el recorrido de los vértices en un grafo, este se realiza de forma recursiva y en este ejemplo al ser muchos datos es necesario aumentar el límite de recursión para lograr cumplir las tareas.

- c. ¿Cuál es el valor inicial que tiene Python como límite de recursión?

El límite de recursión por omisión en Python es de 1000 llamadas recursivas.

### (OPCION 4)

- `bus_routes_50.csv`

```
Numero de arcos: 73
```

```
Time 0.09375
```

- `bus_routes_150.csv`

```
Numero de arcos: 146
```

```
Time 0.109375
```

- `bus_routes_300.csv`

```
Numero de arcos: 382
```

```
Time 0.234375
```

- `bus_routes_1000.csv`

```
Numero de arcos: 1633
```

Time 1.015625

- bus\_routes\_2000.csv

Numero de arcos: 3560

Time 3.53125

- bus\_routes\_3000.csv

Numero de arcos: 5773

Time 6.96875

- bus\_routes\_7000.csv

Numero de arcos: 15334

Time 16.234375

- bus\_routes\_10000.csv

Numero de arcos: 22758

Time 43.1875

- bus\_routes\_14000.csv

Numero de arcos: 32270

Time 91.9375

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

Ha medida que aumenta el número de vértices, los arcos tienden a ser casi el doble de la cantidad de vértices una vez hay varias rutas. Y por otro lado entre más vértices y arcos se demora más tiempo en realizarse la operación 4.

(OPCION 6)

- bus\_routes\_50.csv

Numero de arcos: 73

TIME: 0

- bus\_routes\_150.csv

Numero de arcos: 146

TIME: 0.015625

- bus\_routes\_300.csv

Numero de arcos: 382

TIME: 0.0

- bus\_routes\_1000.csv

Numero de arcos: 1633

TIME: 0.0

- bus\_routes\_2000.csv

Numero de arcos: 3560

TIME: 0.0

- bus\_routes\_3000.csv

Numero de arcos: 5773

TIME: 0.015625

- bus\_routes\_7000.csv

Numero de arcos: 15334

TIME: 0.0

- bus\_routes\_10000.csv

Numero de arcos: 22758

TIME: 0.0

- bus\_routes\_14000.csv

Numero de arcos: 32270

TIME: 0.015625

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

El grafo es disperso, ya que teniendo la ecuación de aristas para un grafo completo vemos que las aristas están menos de la mitad del máximo de aristas posibles. El grafo es dirigido ya que los servicios de transporten se identifican por una estación de inicio y una final. Y el grafo es fuertemente conectado ya que un par de vértices tienen arcos que van de ida y de vuelta, esto se evidencia en los condicionales planteado en el controller para agregar una parada.

- f. ¿Cuál es el tamaño inicial del grafo? Es de 14000, ya que es dado por parámetro en el momento de crear el grafo.

g. ¿Cuál es la Estructura de datos utilizada?

Es un Grafo dirigido y un mapa con la información de cada parada.

h. ¿Cuál es la función de comparación utilizada?

Es una función que compara los vértices que viene siendo los ID's de Las paradas, y esto permite darle una noción de orden al grafo, en el momento que se requiera evaluar.