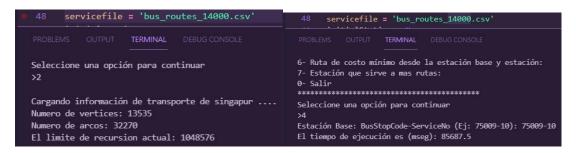
Preguntas del laboratorio 10

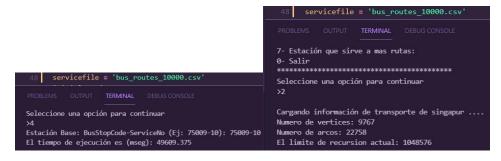
- a) ¿Qué instrucción se usa para cambiar el límite de recursión de Python?
 - sys. setrecursionlimit(n)
- b) ¿Por qué considera que se debe hacer este cambio?
 - Para evitar recursividades infinitas.
- c) ¿Cuál es el valor inicial que tiene Python cómo límite de recursión?
 - El límite máximo depende de la plataforma. Sin embargo, este límite máximo suele ser 1000.
- d) ¿Qué relación creen que existe entre el número de vértices, arcos y el tiempo que toma la operación 4?

El número de vértices, arcos y el tiempo que toma la operación 4, tienen una relación directamente proporcional, ya que mientras el número de arcos y vértices cargados disminuye, el tiempo también lo hace y viceversa.

Archivo 14000



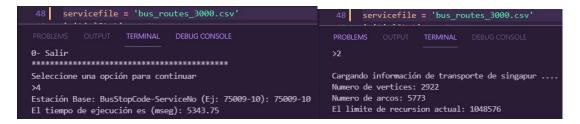
Archivo 10000



Archivo 7000



Archivo 3000



Archivo 2000



Archivo 1000



• Archivo 300



• Archivo 150

• Archivo 50

- e) ¿El grafo definido es denso o disperso?, ¿El grafo es dirigido o no dirigido?, ¿El grafo está fuertemente conectado?
 - Es un grafo dirigido debido a que las rutas tienen una dirección específica entre las estaciones.
 - El grafo está fuertemente conectado ya que existe un camino entre la estación de base y una estación especifica como en este caso (ej:15151-10).
 - Se halla la densidad del grafo, y según el resultado podemos evidenciar que el grafo es disperso ya que su densidad es por debajo de 0,3. Esto quiere decir que su número de aristas es bajo.

Densidad =
$$\frac{e}{V(V-1)}$$
 $\frac{V=Verfice}{e=arros}$ archivo 14000 $\frac{31270}{e=32270} = 0,00017616$

- f) ¿Cuál es el tamaño inicial del grafo?
 - El tamaño inicial es de 14000.
- g) ¿Cuál es la Estructura de datos utilizada?
 - El grafo se implementa sobre una lista de adyacencias (ADJ_LIST) y se utiliza una tabla de hash para guardar los vértices del grafo.
- h) ¿Cuál es la función de comparación utilizada?

Compara los vértices del grafo, que corresponden a las estaciones.

```
def compareStopIds(stop, keyvaluestop):
"""
Compara dos estaciones
"""
stopcode = keyvaluestop['key']
if (stop == stopcode):
    return 0
elif (stop > stopcode):
    return 1
else:
    return -1
```