Pregunta 1: ¿Qué características tiene el grafo definido?, ¿Tamaño inicial, es dirigido?, ¿Estructura de datos utilizada?

R: El grafo definido posee elementos llamados vértices, que están conectados con otros a través de arcos. El tamaño inicial es de 14000, es un grafo dirigido porque hay una conexión especifica entre rutas. Se utiliza la estructura de mapas y grafos en el ejemplo de prueba.

1.1 Ejecutar el Ejemplo

Ahora desde el archivo view.py ejecuten el ejemplo. El archivo view.py tiene el nombre del archivo a leer. En el directorio Data, encuentran archivos de diferentes tamaños: 50, 150, 300, 1000, etc. El nombre del archivo indica aproximadamente el número de líneas del archivo CSV.

Cambien en el archivo view.py, de archivo, comenzando con el más pequeño y luego con el siguiente en tamaño hasta probar con todos.

Al ejecutar el ejemplo con cada archivo, encontrarán las siguientes opciones:

```
****************************************
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estación base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a más rutas:
0- Salir
****************************************
```

Ejecuten la opción 1 para crear las estructuras.

Ejecuten la opción 2 para crear el grafo a partir del archivo de rutas. Por ejemplo, al ejecutar el programa con el archivo: 'bus_routes_300.csv'

Obtienen la siguiente información:

```
****************************************
Cargando información de transporte de singapur ....
Numero de vértices: 295
Numero de arcos: 382
El límite de recursión actual: 1000
El límite de recursión se ajusta a: 20000
```

Tiempo de ejecución: 0.05674999799999991

****

- En este caso verán varios mensajes.
- El primero es que se actualiza el número de llamados recursivos de Python. Identifiquen en el programa, dónde se realiza el cambio.

Pregunta 2: ¿Qué instrucción se usa para cambiar el límite de recursión de Python? ¿Por qué considera que se debe hacer este cambio?, ¿Cuál es el valor inicial que tiene Python cómo límite de recursión?

La instrucción usada para cambiar el límite de recursión es getrecursionlimit, este cambio se debe realizar para evitar una recursión infinita y que Python falle, además, el valor inicial que tiene Python como límite de recursión es 20000.

Cada instrucción del programa arroja el tiempo que tomó su ejecución. La opción 4, que luego trabajaremos más a fondo en el curso, es la que más tiempo toma.

Al ejecutarla verán algo como:

****

Estación Base: BusStopCode-ServiceNo (Ej: 75009-10): 75009-10

Tiempo de ejecución: 0.06441147900000033

****

Utilicen el vértice propuesto en el ejemplo: 75009-10 y tomen nota del tiempo que toma esta instrucción con cada uno de los archivos CSV.

Nota: Deben ejecutar siempre las opciones 1 y 2 antes de usar la opción 4.

Esta operación, calcula la ruta más corta desde la estación indicada (75009-10) a todas las otras estaciones (todos los vértices del grafo).

Si luego utilizan la opción 6, y ponen por ejemplo como estación destino: 15151-10, verán el camino propuesto para ir de la estación 75009-10 a la estación 15151-10.

Ahora, ejecuten el programa con cada uno de los archivos CSV. Por cada archivo, tomen nota del número de vértices y el número de arcos del grafo (reportado cuando ejecutan la opción 2) y el tiempo de ejecución que toma la opción 4. (Esta prueba se debe hacer siempre en el mismo computador, idealmente con todos los programas cerrados, solo ejecutando VSCode)

Pregunta 3: ¿Qué relación creen que existe entre el número de vértices, arcos y el tiempo que toma la operación 4? (Ayuda: ¿es un crecimiento lineal?)

R: Sucede que el número de vértices y arcos cambia de distinta manera. Para que crezca de manera lineal se espera que la cantidad de datos en el archivo más pequeño multiplicada por 280 sea igual a la cantidad de datos en el archivo más grande (50 * 280 = 14000), y por ende crezca de la misma manera la cantidad de vértices y arcos. El primer archivo tiene 74 vértices y 73 arcos, pero este crecimiento lineal no se sostiene con el archivo más grande. Hay 13000 vertices y 32000 arcos aproximadamente. Haciendo los cálculos esto no coincide con nuestra predicción (74*280 = 20720 > 13000 y 73*280= 20440 < 32000) ya que la cantidad de vértices es menor y la de arcos es mayor a la esperada.

También podemos revisar los tiempos de carga para comprobar que sucede. El tiempo de carga para el primer archivo es de 0.04 segundos (0.04 * 280 = 11,2 segundos aproximados para el ultimo archivo), pero el ultimo archivo carga en 28 segundos, por lo que podemos concluir que el número de arcos aumenta a una tasa mayor de lo que lo hacen la cantidad de datos, y los tiempos de carga se van haciendo mayores respecto al anterior archivo, por lo que su crecimiento es mayor al lineal.