**OBSERVACIONES DEL LA PRACTICA**

**LabGraph-S13-G01**

Isai Daniel Chacón Silva Cod 201912015

Nicolás Aparicio Claros Cod 201911357

# **Preguntas de análisis**

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

La instrucción utilizada para cambiar el límite de recursión de Python se encuentra dentro del bloque *main* del código en el archivo *view.py*. Esta es, más específicamente, la función *setrecursionlimit* del módulo *sys* con un valor de como parámetro.

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

Es necesario realizar este cambio, ya que al trabajar con este tipo de datos y en especial con estructuras de datos tales como los grafos, es posible alcanzar el tope de recursión por defecto en Python. Es por esto que, el módulo *sys*, permite acceder y modificar valores preestablecidos para el intérprete que se esté utilizando. Redefinir el límite de recursión previene al programa de entrar en un loop infinito, lo cual podría llevar al desbordamiento de la pila C, con la cual trabaja Python. Es por esto que se busca establecer un límite más alto cuando se tiene un programa que requiere una recursividad profunda y una plataforma que admita un límite más alto.

1. ¿Cuál es el valor inicial que tiene Python cómo límite de recursión?

Para obtener el valor inicial del límite de recursión de Python se utilizó la función *sys.getrecursionlimit()* antes de mapear este valor a su nuevo límite más alto. El valor obtenido fue: 1000.

1. ¿Qué relación creen que existe entre el número de vértices, arcos y el tiempo que toma la operación 4?
2. ¿El grafo definido es denso o disperso?, ¿El grafo es dirigido o no dirigido?, ¿El grafo está fuertemente conectado?

Para el grafo de este laboratorio, los arcos representan segmentos de ruta que comunican dos estaciones en el sistema de rutas de Singapur; para las cuales se tiene como peso la distancia entre dos estaciones respectivamente. También, es importante resaltar que el grafo es dirigido, dado que las rutas tienen una dirección definida en particular entre las estaciones.

1. ¿Cuál es el tamaño inicial del grafo?

Inicialmente, al instanciar el *analyzer* en el *model.py*, se puede observar que el número de elementos para el mapa de paradas se inicializa con *numelements=14000* y

Al utilizar el archivo *bus\_routes\_14000*, aquel con mayor cantidad de datos, se obtuvieron los siguientes tamaños del grafo:

Texto

Descripción generada automáticamente

Figura 1.

1. ¿Cuál es la Estructura de datos utilizada?

Se utiliza un analizador que es un diccionario nativo de Python. Este, por su parte, posee una *keys* alusiva a las paradas de autobús y conexiones entre estas, cuyos *values* son un **mapa** y un **grafo** respectivamente, de la librería de DISCLIB.ADT. Para este caso de análisis, nos centraremos en la representación del grafo de conexiones.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Figura 2.

En el código se puede observar que la estructura de datos utilizado para el grafo es una lista de adyacencia, dirigida, de tamaño 14,000.

1. ¿Cuál es la función de comparación utilizada?