**OBSERVACIONES DEL LA PRACTICA**

Estudiante Tomás Bedoya Calixto Cod 202020689

Estudiante Andrés Felipe Gómez García Cod 202021189

# **Preguntas de análisis**

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

sys.setrecursionlimit(n) donde n es el nuevo límite de recursión.

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

Porque al tener una cantidad muy grande datos, es posible que la cantidad de funciones que las analizan requieran un mayor número de recursiones que la cantidad por defecto para funcionar correctamente.

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

El valor por defecto es 1000.

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

Los arcos aumentan con una mayor rapidez que el número de vértices de modo que la relación entre vértices y arcos de inclina hacia los arcos cuando el número de vértices aumenta. Por otro lado, el tiempo de ejecución de la operación 4 aumenta a medida que se incrementan el número de vértices y, por ende, arcos. Igualmente, el tiempo aumenta en mayor proporción que el número de arcos.

1. ¿Qué características tiene el grafo definido?

Es un grafo dirigido guardado en el analyzer como analyzer[“connections”]. El grafo está compuesto por vértices que representan las paradas de bus junto con las líneas de bus a las que hacen parte. Los arcos ilustran las rutas entre los diferentes paraderos y sus distancias. Como un paradero puede hacer parte de varias líneas de bus, se tienen varios vértices que corresponden a un mismo paradero pero que difieren en el identificador del vértice al incluir la línea a la que pertenecen. Dichos vértices de un mismo paradero están conectados entre si para representar los posibles transbordos a las diferentes rutas en una misma ubicación.

1. ¿Cuál es el tamaño inicial del grafo?

14000.

1. ¿Cuál es la Estructura de datos utilizada?

Listas de adyacencia.

1. ¿Cuál es la función de comparación utilizada?

compareStopIds que compara dos estaciones entre sí.

**Datos Función 4**

50:

Num Vertices: 74

Num Arcos: 73

Tiempo de ejecución: 62.82289999999921 ms

150:

Num Vertices: 146

Num Arcos: 146

Tiempo de ejecución: 76.30500000000029 ms

300:

Num Vertices: 295

Num Arcos: 382

Tiempo de ejecución: 200.30249999993248 ms

1000:

Num Vertices: 984

Num Arcos: 1633

Tiempo de ejecución: 392.39849999999933 ms

2000:

Num Vertices: 1954

Num Arcos: 3560

Tiempo de ejecución: 1291.22220000000002 ms

3000:

Num Vertices: 2922

Num Arcos: 5773

Tiempo de ejecución: 2395.361 ms

7000:

Num Vertices: 6829

Num Arcos: 15334

Tiempo de ejecución: 8175.5190000000002 ms

10000:

Num Vertices: 9767

Num Arcos: 22758

Tiempo de ejecución: 28250.371 ms

14000:

Num Vertices: 13525

Num Arcos: 32270

Tiempo de ejecución: 58929.89149999999 ms

**Datos Función 6**

50:

Num Vertices: 74

Num Arcos: 73

Tiempo de ejecución: 14.35930000000076 ms

150:

Num Vertices: 146

Num Arcos: 146

Tiempo de ejecución: 38.444800000000112 ms

300:

Num Vertices: 295

Num Arcos: 382

Tiempo de ejecución: 14.407899999998335 ms

1000:

Num Vertices: 984

Num Arcos: 1633

Tiempo de ejecución: 13.951699999999619 ms

2000:

Num Vertices: 1954

Num Arcos: 3560

Tiempo de ejecución: 18.758899999997084 ms

3000:

Num Vertices: 2922

Num Arcos: 5773

Tiempo de ejecución: 17.597699999999844 ms

7000:

Num Vertices: 6829

Num Arcos: 15334

Tiempo de ejecución: 55.90399999999863 ms

10000:

Num Vertices: 9767

Num Arcos: 22758

Tiempo de ejecución: 26.460500000001048 ms

14000:

Num Vertices: 13525

Num Arcos: 32270

Tiempo de ejecución: 31.629499999999546 ms