**OBSERVACIONES DEL LA PRACTICA**

Javier Cerino Cod 202020873

Marco Zuliani Cod 202022412

# **Preguntas de análisis**

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

La instrucción que se usa para cambiar el límite de recursión de Python es sys.setrecursionlimit(limit). En este caso se establece el parámetro limit como 220 lo que corresponde a 1048576. Esto permite que las funciones se puedan auto llamar hasta el límite establecido. Sin embargo, se debe tener en cuenta que el limite que se establezca puede llegar a provocar un fallo en el programa debido a alguna insuficiencia de la plataforma en la que se utilice.

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

En general los grafos tienden a utilizar muchas funciones recursivas tales como sccCount de la librería scc que permite contar el numero de componentes fuertemente conectados y dfo, entre otros.

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

Python por default generalmente tiene un límite de recursión de 1000. Esto significa que una función no puede llamarse a si misma más de 1000 veces.

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre  Archivo | Numero  Vértices | Numero  Arcos | Opción 4  Tiempo [ms] op.4 | Opción 6  Tiempo [ms] |
| bus\_routes\_50 | 74 | 73 | 46.875 | 0.0 |
| bus\_routes\_150 | 146 | 146 | 46.875 | 0.0 |
| bus\_routes\_300 | 295 | 382 | 93.75 | 15.625 |
| bus\_routes\_1000 | 984 | 1633 | 437.5 | 15.625 |
| bus\_routes\_2000 | 1954 | 3560 | 1437.5 | 15.625 |
| bus\_routes\_3000 | 2922 | 5773 | 2437.5 | 0.0 |
| bus\_routes\_7000 | 6829 | 15334 | 6968.75 | 0.0 |
| bus\_routes\_10000 | 9767 | 22758 | 21312.5 | 15.625 |
| bus\_routes\_14000 | 13535 | 32270 | 37156.25 | 15.625 |

Con el aumento del número de vértices y arcos aumenta también el tiempo de ejecución de la opción numero 4.

1. ¿Qué características tiene el grafo definido?

El grafo definido es un dígrafo ya que sus arcos son dirigidos. Tiene una estructura particular ya que varias rutas de buses tienen mismas paradas y es por esto por lo que se definió que los vértices del grafo almacenarían el código de parada del bus y el número de servicio (<BusStopCode>-<ServiceNo>). Por otro lado los arcos representan las rutas que comunican dos paradas y además como su peso almacenan la distancia entre las dos paradas que conectan.

1. ¿Cuál es el tamaño inicial del grafo?

El tamaño inicial del grafo es de 14000, pero este valor hace referencia al caso en el que se utilice el archivo llamado “bus\_routes\_14000” al momento de la carga de datos. Debido a lo anterior si se utilizara otro archivo al momento de cargar los datos se podría cambiar el tamaño inicial.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. ¿Cuál es la Estructura de datos utilizada?

La estructura de datos utilizada es una lista de adyacencias la cual almacena únicamente la información que es realmente relevante en el grafo a diferencia de las matrices de adyacencias que almacena información irrelevante, la cual aumenta en gran medida el espacio en memoria especialmente en grafos poco conectados. Por ejemplo: si tenemos un grafo en el que tenemos conectados Bogotá con Argentina y Argentina con Chile, en una lista de adyacencias se almacenaría la información de que Bogotá está conectado con Argentina y que Argentina está conectado con Chile; por otro lado en una matriz de adyacencias veríamos que se almacena, además de la información mencionada anteriormente, información que indica que Bogotá no está conectado con Chile ni Chile con Bogotá.

1. ¿Cuál es la función de comparación utilizada?

La función de comparación utilizada en el grafo es “compareStopIds” que como el propio nombre lo indica permite saber si dos estaciones tienen el mismo código o si uno de los códigos es mayor que el otro.

Texto

Descripción generada automáticamente