

Santiago Rodriguez Bernal – 202011182
Juan Sebastian Nuñez – 202021672

Laboratorio 9

- **¿Qué instrucción se usa para cambiar el límite de recursión de Python?**

La instrucción utilizada para cambiar el límite de recursión en Python fue `sys.setrecursionlimit`. Esta instrucción tiene como propósito aumentar el valor correspondiente al límite de la recursión.

- **¿Por qué considera que se debe hacer este cambio?**

Este cambio se hace con el propósito de que una función se pueda llamar a sí misma muchas más veces que las que Python permite. Esto debido a que el intérprete de Python establece un límite con el propósito de evitar que se entre a un ciclo infinito de recursiones. Sin embargo, debido a que se necesitan procesar grandes cantidades de datos, hace que sea necesario aumentar este límite para procesar la totalidad de los datos.

- **¿Cuál es el valor inicial que tiene Python cómo límite de recursión?**

El valor inicial que tiene Python es de 1000.

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

Tamaño de la muestra	Numero de vertices	Numero de arcos	Tiempo (Opción 4)
50	74	73	15,62
150	146	146	46,88
300	295	382	78,12
1000	984	1633	375,00
2000	1954	3560	1140,62
3000	2922	5773	2140,62
7000	6829	15334	6234,38
10000	9767	22758	16250,00
14000	13535	32270	32312,50

Entre más grande el tamaño de la muestra, se hace necesario tener un mayor número de vértices y arcos para almacenar los datos. Por lo tanto, a mayor número de vértices y arcos la operación tomara más tiempo. Esto debido a que la operación 4 calcula la ruta más corta desde la estación dada por consola a todas las demás estaciones, por lo cual se deben revisar todos los vértices del grafo. Así pues, se espera que el tiempo aumente a medida que el número de vértices y arco lo hace. Esto lo podemos evidenciar en la tabla donde se evidencia que el tiempo va en un aumento.

- **¿El grafo definido es denso o disperso?, ¿El grafo es dirigido o no dirigido?, ¿El grafo está fuertemente conectado?**

¿El grafo definido es denso o disperso?

Tenemos que $N = \#Arcos$ y $V = \#Vertices$. Por lo tanto, la densidad está dada por la siguiente formula:

$$x = \frac{N}{V(V-1)}$$

$$x = \frac{32270}{13535(13535-1)} = 0,00018$$

Dado que el valor está por debajo de 0,3 se concluye que el grafo es disperso.

¿El grafo es dirigido o no dirigido?

El grafo es dirigido dado que las rutas tienen una dirección definida entre las estaciones.

¿El grafo está fuertemente conectado?

No está fuertemente conectado ya que no hay un único componente fuertemente conectado. Sin embargo, el grafo está construido sobre varios componentes fuertemente conectados que en este caso son 30 para el archivo más grande.

- **¿Cuál es el tamaño inicial del grafo?**

El tamaño del grafo esta dado por el número de vértices y arcos que tenga. En este caso al evaluar el archivo de 14000 elementos podemos ver que tiene 13535 vértices y 32270 arcos.

- **¿Cuál es la Estructura de datos utilizada?**

La estructura de datos utilizada al momento de crear el grafo es la lista de adyacencia(adjlist.py). Esta permite almacenar la información relevante, es decir los vértices y los arcos presentes.

- **¿Cuál es la función de comparación utilizada?**

La función utilizada fue la de “compareStopsIds”. Esta función se encarga de comparar el valor de la llave “key” de dos estaciones distintas.