**OBSERVACIONES DEL LA PRACTICA**

Nicolas Merchan Cuestas - 202112109

# **Preguntas de análisis**

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

El límite de recursión de Python puede ser cambiado mediante la instrucción sys.setrecursionlimit(recursión\_limit), donde la variable recursión\_limit representa el límite de recursión deseado. Se importó la librería sys.

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

El límite de recursión debe ser cambiado, porque la construcción y análisis de los gráfos que representan el sistema considerado requieren de muchos más procesos recursivos que el límite establecido por defecto en python.

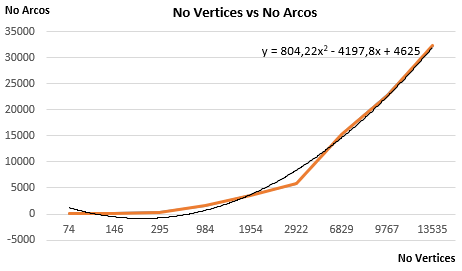
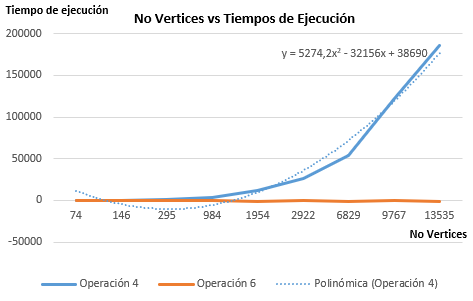
1. ¿Cuál es el valor inicial que tiene Python cómo límite de recursión?

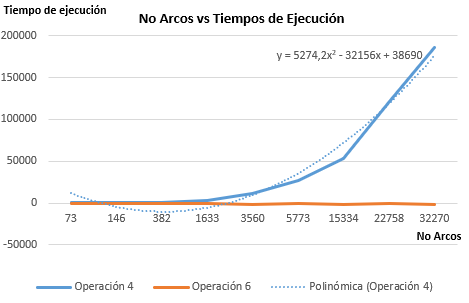
El valor inicial del límite de recursión en Python puede ser obtenido mediante la instrucción sys.getrecursionlimit(), he importado la librería sys. El resultado por defecto de la instrucción son 1000 recursiones.

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

Considero que el número de arcos crece exponencialmente con el incremente en el número de vértices. Así mismo, el tiempo de ejecución de la opción 4 incrementa exponencialmente tanto con el número de vértices como de arcos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. Paradas | Vertices | Arcos | Tiempo - Operación 4 (mseg) | Tiempo – Operación 6 (mseg) |
| 50 | 74 | 73 | 218,75 | 16 |
| 150 | 146 | 146 | 359 | 31,25 |
| 300 | 295 | 382 | 672 | 16 |
| 1000 | 984 | 1633 | 3343,75 | 16 |
| 2000 | 1954 | 3560 | 11968,75 | 0 |
| 3000 | 2922 | 5773 | 26843,75 | 16 |
| 7000 | 6829 | 15334 | 53468,75 | 0 |
| 10000 | 9767 | 22758 | 121843,625 | 31,25 |
| 14000 | 13535 | 32270 | 185625 | 0 |



1. ¿El grafo definido es denso o disperso?, ¿El grafo es dirigido o no dirigido?, ¿El grafo está fuertemente conectado?

El grafo definido es disperso, dado que para los 9 archivos analizados se da que

,

para A igual al número de arcos, V igual al número de vértices y V(V-1) igual al máximo número de arcos que pueden existir entre V vértices. El grafo es dirigido, dado que las rutas tienen una dirección específica entre las estaciones (enunciado). El grafo está fuertemente conectado, dado que siempre existe una ruta para llegar de una estación a otra y viceversa. Dicha ruta no tiene que ser igual cuando se busca retornar a la estación de origen desde la estación de destino (enunciado).

1. ¿Cuál es el tamaño inicial del grafo?

El tamaño inicial del grafo es de 14000 elementos. Ello se evidencia en el parámetro size=1400 (línea 68) de la función gr.newGraph() en model.py.

1. ¿Cuál es la Estructura de datos utilizada?

La estructura de datos utilizada es una lista de adyacencias. Ello se evidencia en el parámetro datastructure=’ADJ\_LIST’ (línea 66) de la función gr.newGraph() en model.py.

1. ¿Cuál es la función de comparación utilizada?

La función compareStopsIds() es utilizada en la creación del grafo definido y comprara los códigos referentes a las estaciones ingresadas por parámetro y retorna 0 cuando los códigos de las estaciones son iguales, 1 cuando el primer código es superior al código posterior ingresado por parámetro, y -1 cuando el primer código es inferior al código posterior ingresado por parámetro. Ello se evidencia en el parámetro comparefunction=compareStopsIds (línea 69) de la función gr.newGraph() en model.py