**OBSERVACIONES DEL LA PRACTICA**

Valeria Caro Ramirez Cod 202111040

Sofia Velasquez Marin Cod 202113334

# **Preguntas de análisis**

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

Para cambiar el límite de recursión de Python, se debe importar el modulo sys y llamar al metodo setrecursionlimit (limit) para definir el nuevo limite de recursion. El parametro que recibe este metodo es limit, que es valor de tipo entero que denota el nuevo limite de la pila de interpretes de python. Cabe recordar que este metodo no retorna ningun valor.

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

Se debe hacer este cambio debido que al tener una gran cantidad de datos a comparar recursivamente, este supera el limite de recursion incial dado por python y por lo tanto este deba ser modificado para tener un limite mayor y asi evitar el error de ‘maximum recursion depth exceed in comparision’.

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

El valor incial que tiene Python como limite de recursion es 1000 llamadas recursivas, para evitar que por error de una mala implementacion de una funcion recursiva infinita, no acabe con la memoria del sistema.

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

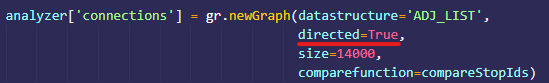
*Toma de datos*

|  |  |  |  |
| --- | --- | --- | --- |
| *Tamaño* | *# Vertices* | *# Arcos* | *Tiempo de Ejecución [ms]*  *Operación 4* |
| 50 | 74 | 73 | 31.25 |
| 150 | 146 | 146 | 46.875 |
| 300 | 295 | 382 | 62.5 |
| 1000 | 984 | 1633 | 218.5 |
| 2000 | 1954 | 3560 | 968.75 |
| 3000 | 2922 | 5773 | 1312.5 |
| 7000 | 6829 | 15334 | 4265.625 |
| 10000 | 9767 | 22758 | 12468.75 |
| 14000 | 13535 | 32270 | 21906.25 |

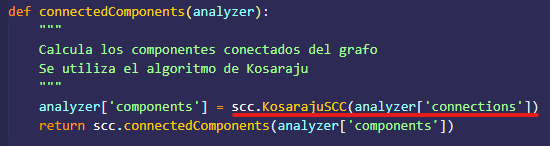
Como se puede observar en la toma de datos realizada, a excepcion del archivo con tamaño 50, para todos se cumple que el numero de arcos es mayor o igual a la cantidad de vertices (estaciones). A excepcion del archivo con tamaño 150, para todos se cumple que el numero de arcos es mayor al tamaño del archivo. El tiempo de ejecucion tiene una relacion proporcinal a el tamaño del archivo (entre mas grande el archivo, mas tiempo requiere para ejecutarse).

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

Cuando se crea el grafo, se pasa por parametro True or False a directed (argumento que indica si el grafo es dirigido o no). En este caso como este argumento esta definido en True, podemos afirmar que este es una grafo dirigido.

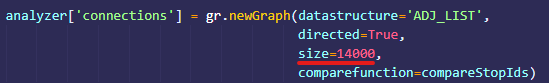


Ademas como la estructura de datos es una lista de adyacencia, el grafo definido es disperso, ya que esta estructura de datos se presenta en grafos que no son densos. Normalmente, si es denso, se opta por usar como estructura de datos matrices de adyacencia. Finalmente, como el grafo es dirigido este es fuertemente conectado ya que cada para cada par de vertices a y b existe un camino a hacia b y un camino de b hacia a. Asismismo, para encontrar los componentes fuertemente conectados se utiliza el algoritmo de Kosaraju.



1. **¿Cuál es el tamaño inicial del grafo?**

Al crear el grafo, se le pasa por parametro el size(el tamaño incial del grafo) que en este caso como podemos observar en la siguiente imagen es 14000.



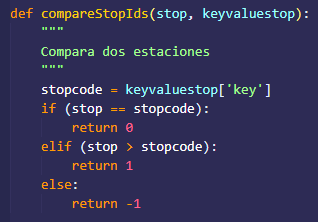
1. **¿Cuál es la Estructura de datos utilizada?**

La estructura de datos utilizada es una lista de adyacencia (‘ADJ\_LIST’), esta representa un grafo.

1. **¿Cuál es la función de comparación utilizada?**

La funcion de comparacion utilizada es compareStopIds para comparar las estaciones, tal como lo podemos observar en la imagen a continuacion.

Se crea la funcion de CompareStopIds



Luego se pasa el CompareStopIds, al parametro de comparefunction cuando se crea el grafo.

