**OBSERVACIONES DEL LA PRACTICA**

Juan Camilo Falla C.E. 201922219

Nicolas Klopstock C.E. 202021352

# **Preguntas de análisis:**

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

La función que devuelve y la que se puede usar para cambiar el límite de recursión de Python es getrecursionlimit() de la librería sys de Python. Esta librería una del estándar de Python. Sys.getrecursionlimit() devuelve el valor del límite de recursión. El límite se ajusta en setrecursionlimit(), también de la librería sys. Este límite es la máxima profundidad de las colas (stacks) del intérprete de Python. Adicionalmente, previene infinitos ciclos de recursión, lo que podría causar un ‘sobrecalentamiento’ de la cola (stack) de C y un ‘crasheo ‘de Python, adicionalmente este límite evita que se dé un consumo excesivo de memoria RAM ya que las recursiones consumen bastante memoria de este tipo.

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

Por lo dicho anteriormente, si no se cambia o ajusta el límite, podría ocupar muchísima memoria y causar un ‘crasheo’ de Python. No obstante, dependiendo del sistema, se puede realizar un ajuste sobre este límite, el cual permite que se realicen operaciones que requieran de un límite de recursión mayor.

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

El valor que Python predetermina como límite de recursión de la función getrecursionlimit() es ‘1000’. En el caso del laboratorio, el límite de recursión está en 2 \*\* 20.

***Toma de datos al ejecutar la opción 4:***

|  |  |  |  |
| --- | --- | --- | --- |
| Archivo | Vertices | Arcos | Tiempo (ms) |
| 50 | 74 | 73 | 31.25 |
| 150 | 146 | 146 | 46.875 |
| 300 | 295 | 385 | 78.125 |
| 1000 | 984 | 1633 | 359.375 |
| 2000 | 1954 | 3560 | 1187.5 |
| 3000 | 2922 | 5773 | 2156.25 |
| 7000 | 6829 | 15334 | 6406.25 |
| 10000 | 9767 | 22759 | 18781.25 |
| 14000 | 13535 | 32270 | 30453.125 |

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

Los resultados de la toma de datos de ejecución de la operación 4 muestran que, entre más vértices y arcos haya, más tiempo va a tardar en ejecutar. Esto se puede deber a que, como se tiene cada vez más y más vértices y arcos, el algoritmo de Dijkstra se va a tardar más en encontrar los caminos de costo mínimo desde la estación inicial a todos los demás vértices del grafo. Entre más vértices se tenga, se van a tener más arcos y más se va a tener que recorrer y revisar para devolver el resultado esperado.

Sobre el número de vértices/nodos y arcos, es bastante difícil predecir. Esto dado que el grafo es un grafo no completo. Un grafo completo es el único grafo en el que se puede saber con exactitud el número de vértices y de arcos que se van a tener. A medida que se hace la carga se van haciendo las conexiones, por eso varía.

***Toma de datos al ejecutar la opción 6:***

|  |  |  |  |
| --- | --- | --- | --- |
| Archivo | Vertices | Arcos | Tiempo (ms) |
| 50 | 74 | 73 | 0.000528 |
| 150 | 146 | 146 | 0.0023 |
| 300 | 295 | 385 | 0.0053 |
| 1000 | 984 | 1633 | 0.0229 |
| 2000 | 1954 | 3560 | 0.0553 |
| 3000 | 2922 | 5773 | 0.0825 |
| 7000 | 6829 | 15334 | 0.1025 |
| 10000 | 9767 | 22759 | 0.1632 |
| 14000 | 13535 | 32270 | 0.2231 |

***¿Qué características tiene el grafo definido?***

¿Se refiere a dirigido? Si sí, una de las características que tiene un grafo dirigido es que los arcos que conectan los nodos/vértices tienen una direccionalidad clara y no se puede tomar el mismo arco para devolverse, es necesario buscar un camino alterno, aunque puede que este camino alterno de regreso no exista.

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

El tamaño inicial del grafo es 14000.

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

En el laboratorio hay dos estructuras que se pueden evidenciar. En primer lugar, se puede ver un grafo principal que está configurado como un grafo de listas adyacentes, en el analyzer también hay una tabla de Hash que utiliza *linear probing* como método de resolución de colisiones.

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

La función de comparación utilizada para construir el grafo es compareStopIds. Esta compara dos estaciones que entran como vértices.