**OBSERVACIONES DEL LA PRACTICA**

Estudiante 1 Cod XXXX

Estudiante 2 Cod XXXX

# **Preguntas de análisis**

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

if \_\_name\_\_ == "\_\_main\_\_":

    threading.stack\_size(67108864)  # 64MB stack

    sys.setrecursionlimit(2 \*\* 20)

    thread = threading.Thread(target=thread\_cycle)

    thread.start()

En particular usamos los threads en los que sí se puede cambiar el límite de recursión, a diferencia de un programa normal.

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

Muchos de los algoritmos en grafos son recursivos, especialmente los que se basan en DFS. Como se tienen grafos muy grandes y con muchas aristas, el algoritmo se llama muchas veces de forma recursiva y supera el límite de recursión de Python. Esto implica que Phyton (o el intérprete) supone que se entró en una recursión infinita de la cual no se va a salir, aunque no sea así y termina el programa de forma no deseada.

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

Usualmente el limite de recursión es 1000 llamados de la función.

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

Maquina 1:

|  |  |  |
| --- | --- | --- |
| Vértices | Arcos | Tiempo |
| 74 | 73 | 32.8945 |
| 146 | 146 | 49.2966 |
| 295 | 382 | 95.8141 |
| 984 | 1633 | 422.1032 |
| 1954 | 3560 | 1446.3904 |
| 2922 | 5773 | 2680.0802 |
| 6829 | 15334 | 10818.0433 |
| 9767 | 22758 | 21038.8027 |
| 13535 | 32270 | 39258.3347 |

Maquina 2:

|  |  |  |
| --- | --- | --- |
| Vértices | Arcos | Tiempo |
| 74 | 73 | 25.9491 |
| 146 | 146 | 53.7461 |
| 295 | 382 | 54.5407 |
| 984 | 1633 | 263.9947 |
| 1954 | 3560 | 885.9763 |
| 2922 | 5773 | 1451.1795 |
| 6829 | 15334 | 5181.7897 |
| 9767 | 22758 | 15479.5549 |
| 13535 | 32270 | 26171.8172 |

Se usa el algoritmo de Dijkstra que tiene complejidad O(AlogV), los datos se asemejan a lo esperado.

1. ¿Qué características tiene el grafo definido?

Es un grafo dirigido con pesos en sus aristas.

1. ¿Cuál es el tamaño inicial del grafo?

Depende del archivo, pero con el más grande se tiene que:

Numero de vertices: 13535

Numero de arcos: 32270

1. ¿Cuál es la Estructura de datos utilizada?

Se usa la graphstructure que está implementadamediangte la Adj\_list

1. ¿Cuál es la función de comparación utilizada?

def compareStopIds(stop, keyvaluestop):

    stopcode = keyvaluestop['key']

    if (stop == stopcode):

        return 0

    elif (stop > stopcode):

        return 1

    else:

        return -1

Esta función compara los códigos de las paradas de las rutas.