

## OBSERVACIONES DEL LA PRÁCTICA

Juan Camilo Colmenares Ortiz - 202011866 - j.colmenares@uniandes.edu.co

Juan Andrés Ospina Sabogal - 202021727 - ja.ospinas1@uniandes.edu.co

## Preguntas de análisis

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

```
sys.setrecursionlimit(2 ** 20)
```

- La instrucción setrecursionlimit() del módulo system se usa para cambiar el límite de recursión de Python.

b) ¿Por qué considera que se debe hacer este cambio?

- Este cambio es necesario pues la mayoría de algoritmos sobre grafos (DFS, por ejemplo) son algoritmos recursivos. Si se trata de grafos muy pequeños, no sería necesario. Sin embargo, los grafos necesarios para implementar el programa son de gran tamaño. Seguramente excederían el límite de recursión predeterminado de Python, lo cual resultaría en un error.

c) ¿Cuál es el valor inicial que tiene Python como límite de recursión?

- Este límite de recursión viene fijado a 1000 de forma predeterminada.

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

Vértices	Arcos	Opción 4 tiempo de ejecución [ms]	Opción 6 tiempo de ejecución [ms]
74	73	48.349	9.749
146	146	53.598	10.235
295	382	89.944	13.334
984	1633	314.776	11.029
1954	3560	851.903	12.510
2922	5773	1657.463	14.611
6829	15334	9185.207	17.668
9767	22758	18612.268	22.826

13535	32270	35434.945	27.313
-------	-------	-----------	--------

- En base a la anterior tabla en donde se hizo una toma de datos de la cantidad de vértices, arcos, y tiempos de ejecución de las opciones 4 y 5 para los diferentes tamaños de archivos, se concluye que a mayor cantidad de vértices mayor cantidad de arcos, lo cual genera tiempos de ejecución mayores.

e) ¿Qué características tiene el grafo definido?

- El grafo definido es aquel en el cual sus arcos tienen una relación unidireccional: apuntan en un solo sentido.

f) ¿Cuál es el tamaño inicial del grafo?

- El tamaño inicial del grafo está dado por el número de vértices que este recibe, en este caso los vértices son las estaciones del bus de Shanghai.

g) ¿Cuál es la estructura de datos utilizada?

- La estructura de datos utilizada es "ADJ\_LIST", implementada al tipo abstracto de datos (TAD) graph (grafo). Su nombre es "lista adyacente".

```
analyzer['connections'] = gr.newGraph(datastructure='ADJ_LIST',
                                     directed=True,
                                     size=14000,
                                     comparefunction=compareStopIds)
```

h) ¿Cuál es la función de comparación utilizada?

- La función de comparación es la que se ve en la pregunta g). Se llama "compareStopIds".

```
def compareStopIds(stop, keyvaluestop):
    """
    Compara dos estaciones
    """
    stopcode = keyvaluestop['key']
    if (stop == stopcode):
        return 0
    elif (stop > stopcode):
        return 1
    else:
        return -1
```