

OBSERVACIONES DEL LAB-9

Santiago Ayala Ciendua - 20210110734

Nicolas Rivera Lesmes - 202116756

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

La instrucción que se usa para cambiar el límite de recursión en Python es `sys.setrecursionlimit(n)`.

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

Consideramos que se debe hacer este cambio pues al momento de operar sobre los grafos se llama varias veces la misma función superando el límite inicial de recursión que tiene Python por defecto.

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

El valor inicial que tiene Python como límite de recursión es de 1000. El límite de 1000 está puesto para evitar que por error (una función recursiva mal implementada que no termina nunca) acabe con la memoria del sistema.

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

TAMAÑO ARCHIVO	VERTICES	ARCOS	TIEMPO (mseg)
50	74	73	31.250
150	146	146	62.500
300	295	382	78.125
1000	984	1633	296.875
2000	1954	3560	875.125
3000	2922	5773	1656.250
7000	6829	15334	6062.500
10000	9767	22758	13781.250
14000	13535	32270	23703.000

Como se puede ver en la tabla a medida que se aumenta el número de vértices y de arcos aumenta el tiempo de manera directamente proporcional. Probablemente el aumento del tiempo también indica que se recorre todo el grafo antes de acabar la función.

e) ¿El grafo definido es denso o disperso?, ¿El grafo es dirigido o no dirigido?, ¿El grafo está fuertemente conectado?

El grafo es disperso debido a que usando la fórmula para calcular si un grafo es disperso o denso ($E/V(V-1)$) donde E es el número de arcos y V el número de Vértices, el resultado para el grafo

utilizado es aproximadamente $1,761e^{-4}$. Por otra parte, sabemos que el grafo es dirigido porque en la definición del grafo en el código se indica que es dirigido (ver captura al final del documento). Por último, gracias a que sabemos el número de componentes conectados (30 para el grafo con los datos 14000) y sabiendo el número de arcos (32270) y vértices (13535), determinamos que el grafo no es fuertemente conectado.

f) ¿Cuál es el tamaño inicial del grafo?

El tamaño inicial de grafo es de 14000

g) ¿Cuál es la Estructura de datos utilizada?

La estructura de datos utilizada es un grafo, uno que más específicamente utiliza las adjacent lists.

h) ¿Cuál es la función de comparación utilizada?

La función de comparación utilizada se llama **compareStopIds**, la cual compara los ids de dos estaciones para organizarlas

Foto para preguntas f,g,h:

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