

# OBSERVACIONES DEL LA PRACTICA

Estudiante 1: Eliana Palacio Pinzón

Código: 202112428

## Preguntas de análisis

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

Rta: La instrucción que se usa para cambiar el límite de recursión se llama setrecursionlimit (int).

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

Rta: Es importante cambiar el límite de recursión para tener un control sobre la memoria y la velocidad de respuesta del algoritmo. Adicionalmente, el límite de recursión puede depender de la precisión que se requiera en los cálculos ya que entre más se repita un experimento, más precisos serán sus resultados.

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

Rta: El valor mínimo de recursiones que permite ejecutar el código sin error es 16 ( $2^4$ ), mientras que el valor inicial que Python tiene como límite es 1048576 ( $2^{20}$ )

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

Rta: Entre más cantidad de vértices tenga el grafo aumenta la cantidad de arcos y a su vez, el tiempo de ejecución. En conclusión, la cantidad de arcos y el tiempo de ejecución son directamente proporcionales a la cantidad de vértices del grafo.

- e) ¿Qué características tiene el grafo definido?

Rta: En un grafo definido, sus vértices tienen un sentido definido.

- f) ¿Cuál es el tamaño inicial del grafo?

Rta: El tamaño inicial del grafo es de 32270

- g) ¿Cuál es la Estructura de datos utilizada?

Rta: El laboratorio utiliza dos tipos de estructuras de datos: Tablas de Hash y Grafos.

- h) ¿Cuál es la función de comparación utilizada?

Rta:

```
def compareStopIds(stop, keyvaluestop):  
    """  
    Compara dos estaciones  
    """  
    stopcode = keyvaluestop['key']  
    if (stop == stopcode):  
        return 0  
    elif (stop > stopcode):  
        return 1  
    else:  
        return -1  
  
def compareroutes(route1, route2):  
    """  
    Compara dos rutas  
    """  
    if (route1 == route2):  
        return 0  
    elif (route1 > route2):  
        return 1  
    else:  
        return -1
```