**Documento Análisis Laboratorio 10**

**Jonathan Rivera Otero, j.riverao@uniandes.edu.co, 202022864**

**Sofia López García, s.lopezg@uniandes.edu.co, 201922558**

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

La función **getrecursionlimit()** Que se importa del módulo local de Python “sys”, y recibe como parámetro el nuevo límite de recursión que se desea imponer.

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

Al trabajar con ciertos algoritmos y estructuras de datos que utilizan la recursión como herramienta de funcionamiento, se entiende que dependiendo de la cantidad de datos que se maneje y de la forma en la que funcionan los algoritmos recursivos implementados, existe la posibilidad de que una función se llame a sí misma en numerosas ocasiones que pueden llegar a exceder el límite que posee Python por defecto, por lo que es necesario cambiar dicho límite para que el programa funcione.

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

El valor por defecto del limite de recursion que posee python es de 1000.

**Toma de datos opción 4:**

**Tamaño Muestra: 14000**

Numero de vértices: 13535

Numero de arcos: 32270

Tiempo: 38292 ms

**Tamaño Muestra: 10000**

Numero de vértices: 9767

Numero de arcos: 22758

Tiempo: 24517 ms

**Tamaño Muestra: 7000**

Numero de vértices: 6829

Numero de arcos: 15334

Tiempo: 8113 ms

**Tamaño Muestra: 3000**

Numero de vértices: 2922

Numero de arcos: 5773

Tiempo: 2197 ms

**Tamaño Muestra: 2000**

Numero de vértices: 1954

Numero de arcos: 3560

Tiempo: 1185 ms

**Tamaño Muestra: 1000**

Numero de vértices: 984

Numero de arcos: 1633

Tiempo: 363 ms

**Tamaño Muestra: 300**

Numero de vértices: 295

Numero de arcos: 382

Tiempo: 91 ms

**Tamaño Muestra: 150**

Numero de vértices: 146

Numero de arcos: 146

Tiempo: 60 ms

**Tamaño Muestra: 50**

Numero de vértices: 74

Numero de arcos: 73

Tiempo: 45 ms

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

Entre más grande sea la muestra de datos, claramente el tiempo de ejecución va a aumentar progresivamente. En cuanto al número de arcos y vértices, si bien con muestras de datos relativamente pequeñas van a estar nivelados, conforme aumenta la muestra el número de arcos va a ir aumentando en mayor cantidad que el número de vértices por lo que eventualmente la cantidad de arcos va a ser considerablemente mayor a la de los vértices. El tiempo de ejecución aumenta ya que la creación de arcos y vértices es una función recursiva por lo que tendrá que repetirse cada vez más veces.

**Toma de datos opción 6:**

**Tamaño Muestra: 14000**

Numero de vértices: 13535

Numero de arcos: 32270

Tiempo: 23.407 ms

**Tamaño Muestra: 10000**

Numero de vértices: 9767

Numero de arcos: 22758

Tiempo: 16.645 ms

**Tamaño Muestra: 7000**

Numero de vértices: 6829

Numero de arcos: 15334

Tiempo: 17.117 ms

**Tamaño Muestra: 3000**

Numero de vértices: 2922

Numero de arcos: 5773

Tiempo: 13.708 ms

**Tamaño Muestra: 2000**

Numero de vértices: 1954

Numero de arcos: 3560

Tiempo: 11.769 ms

**Tamaño Muestra: 1000**

Numero de vértices: 984

Numero de arcos: 1633

Tiempo: 12.975 ms

**Tamaño Muestra: 300**

Numero de vértices: 295

Numero de arcos: 382

Tiempo: 9.407 ms

**Tamaño Muestra: 150**

Numero de vértices: 146

Numero de arcos: 146

Tiempo: 8.040 ms

**Tamaño Muestra: 50**

Numero de vértices: 74

Numero de arcos: 73

Tiempo: 9.665 ms

e. ¿Qué características tiene el grafo definido?

El grafo que se encuentra en el analizador, posee las siguientes caracteristicas:

-Su estructura de datos es una lista adyacente

-Se trata de un grafo dirigido, lo que quiere decir que los arcos que se dirigen a los vértices poseen dirección y no van en ambas direcciones/sentido.

-Posee un tamaño inicial de 14000

-Posee una función de comparación para los Ids de las paradas

f. ¿Cuál es el tamaño inicial del grafo?

14000

g. ¿Cuál es la Estructura de datos utilizada?

Lista adyacente

h. ¿Cuál es la función de comparación utilizada?

CompareStopIds: compara dos valores y analiza cual es mayor y menor, o si poseen el mismo valor