**OBSERVACIONES DEL LA PRACTICA**

Estudiante 1 Lindsay Vanessa Pinto Morato cód. 202023138

Estudiante 2 José Daniel Montero cód. 202012732

# **Preguntas de análisis**

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

El límite de recursión de Python puede ser alterado a través de la función del módulo system *sys.setrecursionlimit().* Este límite previene que cualquier programa entre en una recursión infinita, lo cual puede llevar a que el sistema colapse, o que de un error que prevenga que suceda esta recursión en primer lugar.

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

A la hora de trabajar con tipos de datos abstractos complejos se debe tener en cuenta que cada llamado recursivo ocupa espacio en memoria e incrementando a su vez el tiempo de ejecución que podría ser mejorado con otro tipo de algoritmos.

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

El límite de recursión por predeterminado es de 1000, debido a que el programador promedio no debe necesitar que una función se llame a sí misma más de 1000 veces, por lo que para minimizar el uso de memoria, se deja un límite bajo.

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

Registro de datos:

|  |  |  |  |
| --- | --- | --- | --- |
| **ARCHIVO** | **VÉRTICES** | **ARCOS** | **TIEMPO[Ms]** |
| bus\_routes\_50 | 74 | 73 | 24,121 |
| bus\_routes\_150 | 146 | 146 | 36,286 |
| bus\_routes\_300 | 295 | 382 | 52,499 |
| bus\_routes\_1000 | 984 | 1633 | 331,099 |
| bus\_routes\_2000 | 1954 | 3560 | 1012,296 |
| bus\_routes\_3000 | 2922 | 5773 | 1744,072 |
| bus\_routes\_7000 | 6829 | 15334 | 5533,881 |
| bus\_routes\_10000 | 9767 | 22758 | 17155,943 |
| bus\_routes\_14000 | 13535 | 32270 | 28817,668 |

Como se puede ver en los gráficos anteriores. En el cual se usó como variable independiente el número de arcos y vértices respectivamente y como variable dependiente el tiempo; se puede observar que a medida que el archivo aumenta de tamaño (y con ello la cantidad de vértices y arcos), el tiempo de ejecución es mayor teniendo una tendencia de polinomio grado 2.

Opción 6

|  |  |  |  |
| --- | --- | --- | --- |
| **ARCHIVO** | **VÉRTICES** | **ARCOS** | **TIEMPO[Ms]** |
| bus\_routes\_50 | 74 | 73 | 0,522 |
| bus\_routes\_150 | 146 | 146 | 0,559 |
| bus\_routes\_300 | 295 | 382 | 0,611 |
| bus\_routes\_1000 | 984 | 1633 | 0,664 |
| bus\_routes\_2000 | 1954 | 3560 | 0,845 |
| bus\_routes\_3000 | 2922 | 5773 | 0,789 |
| bus\_routes\_7000 | 6829 | 15334 | 0,971 |
| bus\_routes\_10000 | 9767 | 22758 | 1,06 |
| bus\_routes\_14000 | 13535 | 32270 | 1,277 |

Con respecto a la opción 6 se pueden ver tiempos de ejecución mucho más cortos que en la opción 4 y con un comportamiento de tendencia lineal

1. ¿Qué características tiene el grafo definido?

Un grafo es una representación de un conjunto de datos conocidos como vértices que se relacionan con otros nodos a través de un conjunto de conexiones conocidas como arcos. Los grafos permiten estudiar las relaciones que existen entre unidades que interactúan con otras.

Un grafo dirigido o digrafo es un tipo de grafo en el cual los arcos tienen un sentido definido el cual no necesariamente es bidireccional y pueden tener pesos variables dependiendo del arco.

1. ¿Cuál es el tamaño inicial del grafo?

Inicialmente el grafo tiene un tamaño de 14000 datos

1. ¿Cuál es la Estructura de datos utilizada?

Cuando se crea el nuevo grafo, se utiliza como estructura de datos una lista de adyacencia.

1. ¿Cuál es la función de comparación utilizada?

La función de comparación usada es compareStopIds que compara dos estaciones con los criterios stop y keyvaluestop