**LABORATORIO NO. 11: GRAFOS**

**Preguntas del lab solucionadas por:**

Juan Pablo Lora Hernández- 202012524 - [j.lora@uniandes.edu.co](mailto:j.lora@uniandes.edu.co)

Cristian David Caro Blanco - 202011710 - [c.caro@uniandes.edu.co](mailto:c.caro@uniandes.edu.co)

**Preguntas + Respuestas**

**Pregunta 1:** ¿Qué características tiene el grafo definido?, ¿Tamaño inicial, es dirigido?, ¿Estructura de datos utilizada?

Tamaño Inicial:

-> 14000

Dirigido:

-> Si debido a que como se mencionó en un principio, entre 2 estaciones siempre hay una ruta, por lo que están conectadas, además es diferente ir a EstaciónA por la rutaA desde EstaciónB que ir a EstaciónB por la rutaA desde EstaciónA, lo que quiere decir esto, es que depende de una dirección.

Estructura:

-Stops: Map Probing

-Connections: Grafo dirigido

-Components:

-idscc: Map Probing

-marked: Map Probing

-grmarked: Map Probing

-components: int

-paths:

-source: Ej: 75009-10

-visited: Mapa probing

-iminpq: Cola de prioridad orientada a menor

**Pregunta 2:** ¿Qué instrucción se usa para cambiar el límite de recursión de Python?  ¿Por qué considera que se debe hacer este cambio?, ¿Cuál es el valor inicial que tiene Python cómo límite de recursión?

Instrucción:

-> sys.setrecursionlimit(New\_Límite)

¿Se debe hacer?

-> Sí, se debe cambiar para limitar las recursiones que puede hacer una función, de otro modo, sus recursiones podrían ser infinitas y el programa explotaría. Lo que deja la duda es cómo saber la cantidad límite sabiendo que la recursión “se ajusta a” 20000 (dato dado por los monitores), pero al usar “getrecursionlimit” el resultado es menor (es 1000), lo que podría decir esto es que cada función puede hacer 1000 recursiones de sí misma, pero si 20 funciones hacen el límite de recursión (1000), llegan a 20000, de ahí que, se cumple el límite y cualquier otra recursión no se hace. La duda es ese 20000, por qué no 21000 0 20001.

Valor inicial:

-> 1000 en todos los archivos cargables

**Pregunta 3:** ¿Qué relación creen que existe entre el número de vértices, arcos y el tiempo que toma la operación 4? (Ayuda: ¿es un crecimiento lineal?)

Vértices vs Arcos

Caso1: Al haber pocos datos, asociados según la prueba a menos de 300, los vértices son casi iguales a los arcos, en caso de no ser iguales, varían en +/- 1. EJ:

|  |  |  |
| --- | --- | --- |
|  | Archivo 50 | Archivo 150 |
| Vértices | 74 | 146 |
| Arcos | 73 | 146 |

Caso2: Al haber muchos datos, según las pruebas un ejemplo es, más de 300, los vértices son menores a la cantidad de datos y a los arcos, entre más datos hay más arcos y, menor es la cantidad de vértices comparados ya sea con los arcos o cantidad de datos, pero la proporción de vértices aumenta. EJ:

|  |  |  |
| --- | --- | --- |
|  | Archivo 1000 | Archivo 3000 |
| Vértices | 984 | 1954 |
| Arcos | 1633 | 3560 |

En el segundo archivo hay más vértices que el primer archivo claramente y, los vértices son la cantidad menor siempre:

1633/984 = 1.6 más arcos que vértices

3560/1954 = 1.8 más arcos que vértices

Proporción de vértices (1954/984) 1.98 veces mayor

Vértices, Arcos vs Tiempo Op.4

Caso1: Cuando los datos dados son menores a 300, el comportamiento del tiempo de la operación 4 es extraño, pues parece aumentar a medida que hay más datos, pero en un punto cae. EJ:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Archivo 50 | Archivo 150 | Archivo 300 |
| Tiempo Op.4 | 0.033 | 0.36 | 0.11 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Archivo 300 | Archivo 3000 | Archivo 14000 |
| Tiempo Op.4 | 0.11 | 0.92 | 21.6 |

Caso2: Con base en la información, cuando los datos son mayores a 300 el tiempo de la operación 4 aumenta linealmente. EJ:

Conclusión:

-> Su comportamiento es de crecimiento lineal cuando los archivos a analizar son de 300 datos en adelante, si son menores su análisis es más complejo por las variaciones del comportamiento de los datos.