

Lab 9

ISIS 1225 (4)

| Ronald Pardo | 202111309 | r.diazp@uniandes.edu.co |
| Juan Andres Ruiz Uribe | 201914351 | ja.ruizu@uniandes.edu.co |

- a) La instrucción utilizada para cambiar el límite de recursión de python es:

```
sys.setrecursionlimit(2 ** 20)
```

Este método establece la profundidad máxima de la pila de intérpretes de Python en $2^{**} 20$. Este límite evita que la recursividad infinita cause un desbordamiento de la pila de C y bloquee Python. El límite más alto posible depende de la plataforma.

- b) Cambiar el límite de recursión es importante porque puede obstaculizar el funcionamiento recursivo del recorrido de búsqueda DFS, el cual se encuentra implementado en este laboratorio por medio de los archivos dfs.py y dfo.py. Si bien este está para prevenir loops infinitos y/o problemas parecidos, ciertos métodos necesitan ser llamados una cantidad de veces superior al límite por default.
- c) El valor inicial que trae Python como límite de recursión es 1000, lo que significa que un método puede ser llamado un máximo de mil veces.
- d) La operación 4 se encarga de encontrar todas las rutas de costo mínimo de cada vértice con el vértice base por medio del algoritmo Dijkstra, el cual de acuerdo a Wikipedia¹ cuenta con una complejidad de $O((V+E)\log V)$ que indica que esta es directamente proporcional al número de arcos y vértices y por lo tanto tendrá una complejidad de tiempo \mathcal{O} mas alta en comparación a los demás requerimientos.
- e) De acuerdo a la carga de datos se reporta que

$$V = 13535,$$

$$E = 32270,$$

adicionalmente, se aprecia que en la línea 66 del model se decide crear el graph con tipo dirigido, es decir, de arcos con un solo sentido, por lo que la fórmula para calcular la densidad de graphs dirigidos es la siguiente:

$$D = \frac{|E|}{|V|(|V|-1)},$$
$$D = \frac{32270}{13535(13534)} = 0.00018.$$

Por lo tanto se percibe una densidad menor a 0.3, lo cual nos indica que se trata de un grafo disperso.

Ademas, al ejecutar la opcion 3 obtenemos que el numero de componentes conectados es igual a 30, este hace referencia a un conjunto de vertices de los cuales a partir decada uno de ellos podemos llegar a los otros vertices de ese componente, esto significa que hay 30 de estos conjuntos y que por lo tanto no se cumple que para cada

¹ Algoritmo Dijkstra (1959), recuperado de: https://en.wikipedia.org/wiki/Dijkstra's_algorithm

uno de los caminos del grafo completo se puede ir a todos los demas caminos del grafo. Por lo tanto se deduce que el grafo no es fuertemente conectado.

- f) De acuerdo a la lineal 66 de model.py, se observa que el parámetro size=14000, por lo que ese es el tamaño inicial del grafo.
- g) De acuerdo a la línea 66 de model.py y a la implementación realizada en el archivo graphstructure.py de la librería ADT, se observa que el tipo de estructura de datos es un ADJ_LIST, la cual hace referencia a una forma de representar los grafos por medio de listas de adyacencias.
- h) De acuerdo a la línea 66 de model.py, la función de comparación utilizada es compareStopIds, la cual se encarga de comparar las estaciones por medio de sus paradas.