Universidad de los Andes — José Nicolás Cárdenas (201922006 - j.cardenast)
Estr. de Datos y Algoritmos – Sección 6 — Pablo Pedreros (202112491 - p.pedreros)

Reto 4: Documento de Análisis

Análisis De Complejidad

❖ Carga de datos

La carga de datos constará de dos grafos (uno dirigido y un dígrafo), una lista con todas las ciudades de los datos y seis mapas para resolver los requerimientos, cada uno con uno de los siguientes valores como llaves:

- Nombre de los aereopuertos (con su respectivo diccionario como valor).
- Nombre de los aereopuertos (con un mapa como valor, este mapa tendrá una llave que indica el número de vuelos que llegan al aereopuerto y otra que indica el número de vuelos que salen de este).
- Nombre de los aereopuertos (con un mapa como valor, este mapa tendrá una llave por cada vuelo que sale de este, indicando el código IATA del aereopuerto de destino del vuelo, los valores de esas llaves serán la distancia de ese vuelo).
- Código IATA de los aereopuertos (con su respectivo diccionario como valor).
- Nombre de las ciudades (cada nombre tiene como valor una lista con todos los ID's de ciudades identificadas con ese nombre).
- ID's de las ciudades (cada ID tiene como valor el diccionario con la información de su respectiva ciudad).

❖ Requerimiento 1

Ya con el mapa en la carga de datos que indica cuántas conexiones hay de salida del aereopuerto y cuántas hay de llegada, solo habrá que iterar por todos los vértices del grafo para ver cuáles son los de más conexiones. Esto se hará agregando los cinco primeros vértices en orden a una lista, y comparándolos con todos los otros vértices por los que se itere para ver si tienen más conexiones. Esto seignificará que si hay V vértices en el grafo, el requerimiento, en su pero caso, hará $5N$ iteraciones, lo que significa que el requerimiento en notación O tiene un orden de complejidad de $O(V)$.

❖ Requerimiento 2

El algoritmo importante de este requerimiento es el de *Kosaraju*, el cual fue necesario implementar para encontrar los componentes conectados dentro del grafo *"dirigido"*. Por ende, se debe concluir que la compljidad de este en notación *Big O* es de $O(e + v)$.

❖ Requerimiento 3

Lo primera función que hará el tercer requerimiento será la de encontrar los aereopuertos más cercanos a la ciudad de origen y a la ciudad de destino respectivamente. Como no se pueden filtrar los aereopuertos por regiones, cada una de estas dos búsquedas constará de V comparaciones (tomando V como el número de aereopuertos en el archivo). Ya con ambos aereopuertos, el requerimiento aplicará el algoritmo de Dijsktra al grafo para encontrar la ruta más corta entre los dos aereopuertos, lo que tendrá una complejidad de $O(ElogV)$ (siendo E el número de arcos del grafo). Adicionalmente, el grafo recorrerá el número de ciudades homónimas en caso de que sea necesario y el número de arcos y de vértices de la ruta final, más estos valores no alcanzan números altos y no los tendremos en cuenta para la complejidad. Por esto, la complejidad final del requerimiento será la de encontrar ambos aereopuertos más la de la aplicación del algoritmo de Dijsktra, por lo que el requerimiento tendrá un orden de complejidad de $O(V+ElogV)$.

❖ Requerimiento 5

En este requerimiento se realizan diversas instrucciones que tienen complejidad constante, como acceder a parejas llave-valor de un mapa o añadir elementos en la última posición de una lista. Por ende, la única instrucción importante es la que hace referencia a acceder a los nodos adyacentes a un aeropuerto; aun así, debido a que el grafo dirigido se implementó como una lista de adyacencias, entonces esta operación también se hace en tiempo constante, ya que dicha no es más que acceder a una pareja llave-valor de una Tabla de Hash. Por ende, se concluye que la complejidad del algoritmo en $O(1)$.