OBSERVACIONES DEL RETO 4

María Catalina Ibáñez Piñeres Cod 201922462 Mail m.ibanez@uniandes.edu.co
María Alejandra Pérez Petro Cod 201923972 Mail ma.perezp@uniandes.edu.co

Análisis de complejidad de cada uno de los requerimientos en Notación O

Carga de datos

La complejidad de la carga de datos es O(Nlog(N)) donde N es el número de lineas (viajes) en el csv. Esta complejidad se debe a que primero se recorre cada linea del csv para filtrar los viajes y crear tres tablas de hash ("stations_table", "trip_table", "bike_id") y un arbol RBT ('dateTrips') en el catalogo. Luego, se recorre las llaves de "trip_table", para construir el grafo "connections" cuyos vertices son las estaciones de origen y de destino. Por lo que, la complejidad de la carga de datos es O(Nlog(N)), pues la operación que más aporta a la complejidad es la construcción del arbol RBT.

Requerimiento 1

La complejidad del primer requerimiento es O(N*log(N)) donde N es el número de estaciones (vertices) del grafo "connections". Esta complejidad se debe a que se recorren los vertices del grafo para calcular el outdegree de cada uno de ellos. Luego los vertices se organizan según el número de "Out trips" por medio de Merge sort y, finalmente, se seleccionan los primeros 5 vertices. El Merge Sort es la operación que más aporta a la complejidad, entonces la complejidad del requerimiento 1 es O(N*log(N)).

Requerimiento 2

La complejidad del segundo requerimiento es O(E*log(V)) donde V es el número de estaciones (vertices) y E es el número de arcos (de viajes sin repeticiones) del grafo "connections". Esta complejidad se debe a que se realiza el algoritmo de Dijkstra sobre el grafo "connections" y luego se halla el PathTo para cada vertice del grafo, asi como, la suma de los pesos de los arcos que componen el PathTo. El algoritmo de Dijkstra es la operación que más aporta a la complejidad, entonces la complejidad del requerimiento 2 es O(E*log(V)).

Requerimiento 3

La complejidad del tercer requerimiento es O(NM) donde N es el número de componentes conexos y M es el número de estaciones en el componente N. Esta complejidad se debe a que realiza el algoritmo de Kosaraju sobre el grfo "connections" con complejidad O(V+ E). Luego se recorre cada componente conexo y las esraciones dentro de el para hallar el número de estaciones en el componente, el identificador y nombre de la estación donde más viajes inician, asi como, el identificador y nombre de la estación donde más viajes terminan. Por lo que el recorrido es la operación que más aporta a la complejidad, por lo que la complejidad del requerimiento 3 es O(NM).

Requerimiento 4

La complejidad del cuarto requerimiento es O(E*log(V)) donde V es el número de estaciones (vertices) y E es el número de arcos (de viajes sin repeticiones) del grafo “connections”. Esta complejidad se debe a que se realiza el algoritmo de Dijkstra sobre el grafo “connections” y luego se halla el PathTo hacia la estación de destino. Finalmente, se recorren los arcos que componen dicho PathTo y se suman sus pesos. El algoritmo de Dijkstra es la operación que más aporta a la complejidad, entonces la complejidad del requerimiento 4 es O(E*log(V)).

Requerimiento 5

La complejidad del quinto requerimiento es O(NM) donde N es el número de fechas dentro del intervalo de interes y M es el número de viajes en la fecha N. Esta complejidad se debe a que primero se utiliza el arbol RBT 'dateTrips' para obtener los viajes que inician en el rango de fechas de interes. Luego, se recorren dichos viajes (es decir, cada fecha de inicio dentro del intervalo de interes y luego cada viaje en la fecha de inicio n), con el fin de filtrar y agregar solo los viajes que terminen en el intervalo de interes a una tabla de hash. Luego, se recorre las llaves de la tabla de hash creada de tamaño N*M para construir el grafo cuyos vertices son las estaciones de origen y de destino de los viajes en la fecha de interes. Finalmente, se recorren los vertices del grafo para hallar la estación de origen más frecuentada, la estación de destino más utilizada, La hora del día en la que más viajes inician y la hora del día en la que más viajes terminan. Los recorridos hechos son la operación que más aporta a la complejidad, por lo que la complejidad del requerimiento 5 es O(NM).

Requerimiento 6

La complejidad del sexto requerimiento es O(V*E) donde V es el número de vertices y E el número de arcos del grafo. Esta complejidad se debe a que primero se utiliza la tabla de hash 'bakeID' para obtener los viajes de la bicicleta de interes. Luego, se recorren dichos viajes para construir el grafo cuyos vertices son las estaciones de origen y de destino de los viajes hechos por la bicicleta de interes. Finalmente, se recorren los vertices y arcos del grafo para hallar el total de viajes que iniciaron en dicha estación, el total de viajes que terminaron en dicha estación, el viaje de mayor duración promedio saliendo de la estación de consulta, la estación donde terminaron la mayoría de los viajes que iniciaron en la estación. Este recorrido hecho es la operación que más aporta a la complejidad, por lo que la complejidad del requerimiento 6 es O(VE).