

## **Grupo 8** - Proyecto final

Nikolas Santacruz - Juan David Quintero

### **Preguntas:**

#### **Parte A:**

¿Qué TAD utilizaron en la solución del requerimiento?

Para solucionar el primer requerimiento se utiliza una lista encadenada.

¿Por qué eligieron esa estructura de datos?

Utilizamos esta estructura de datos porque de esta forma podíamos almacenar los datos procesados en una estructura de datos lineal y con ello poder ordenarla de tal forma que cumpla con los requerimientos del usuario. Como en este caso nos piden retornar un número de dato que corresponde con las compañías de taxis más grandes, podemos utilizar la lista enlazada a nuestro favor para poder iterar esta estructura según los datos recibidos como parámetro.

¿Cuál es la complejidad estimada del algoritmo implementado?

Al momento de ejecutar el algoritmo se tienen que primero organizar los datos encontrados en una lista encadenada, para esto se utilizó en dos ocasiones el algoritmo de ordenamiento llamado MergeSort, el cual tiene una complejidad temporal de  $O(n \log(n))$ . Después se itera la misma lista lineal en dos ocasiones distintas, por lo que a la complejidad temporal de este procedimiento es de  $2n$ . El costo temporal total de este requisito es  $O(n \log(n) + 2n)$ , aunque la constante 2 se puede depreciar por la notación big O.

#### **Parte B:**

¿Qué TAD utilizaron en la solución del requerimiento?

En el segundo requerimiento se utilizó un árbol binario rojo negro

¿Por qué eligieron esa estructura de datos?

Para poder retornar la información de los taxis según las fechas suministradas por el usuario, elegimos utilizar un árbol binario para poder encontrar la fecha inicial y a partir de aquí poder retornar la rama inferior a partir de la fecha encontrada. Se utilizó específicamente el árbol binario rojo negro debido a que este tipo de estructura de datos tiende a ser un árbol balanceado a pesar de que el tiempo de carga de los datos será mayor.

¿Cuál es la complejidad estimada del algoritmo implementado?

La complejidad temporal de ejecutar este requisito es de  $O(\log(n))$ , ya que el árbol está balanceado y cada que se avanza en profundidad por cada rama, se descarta la mitad de las posibilidades, por eso tiene un orden logarítmico.

### **Parte C:**

¿Qué TAD utilizaron en la solución del requerimiento?

Para el ultimo requerimiento se utilizó un grafo con lista de adyacencias

¿Por qué eligieron esa estructura de datos?

En este escenario en donde se tenia que modelar una ruta de viaje, se utilizó un grafo bidireccional para poder representar los puntos de partida y destinos, con el fin de determinar el tiempo requerido se viajar entre estos dos lugares, siempre y cuando permanezcan al mismo componente conectado.

¿Cuál es la complejidad estimada del algoritmo implementado?

Debido a que se utilizó el algoritmo de Dijkstra para poder encontrar una ruta de costo mínimo entre las estaciones dadas como parámetro, el primero costo temporal está asociado con este procedimiento y es igual a  $O(E \log V)$  donde E son los nodos y V son los vértices. Una vez obtenida la ruta de costo mínimo ahora se procederá la pila que se recibe como retorno del algoritmo mencionado anteriormente, en donde se tiene que iterar una estructura de datos lineal y por tanto su costo temporal es igual a  $O(n)$ , Por lo tanto, la complejidad temporal total es igual a  $O(E \log(V) + n)$