

Integrantes grupo 6:

- Sebastián Heredia Vargas | s.herediav@uniandes.edu.co | 202012346
- Felix Samuel Rojas Casadiego | fs.rojas@uniandes.edu.co | 202013414

Requerimiento 1

• ¿Qué TAD utilizaron en la solución del requerimiento?

Se utilizaron dos TAD's: una lista y un mapa. Inicialmente para el TAD lista se trabajó con la estructura de datos ARRAY_LIST y SINGLE_LINKED que provee el DISClib, pero luego de hacer las pruebas se demoraba 60 segundos en la carga de datos en el archivo small a la hora de utilizar la función IsPresent, la cual verifica si un taxi que entra como parte de un trip está repetido. Por esto, se probó usar la lista por defecto que tiene Python y el operador "in", para verificar si el taxi ya está dentro de la lista y si no, agregarlo. De esta forma, se logró bajar el tiempo de carga a 8 segundos.

• ¿Por qué eligieron esa estructura de datos?

Porque al utilizar el TAD map logramos retornar el valor de la llave (que es el nombre de cada compañía) con un orden de crecimiento de $O(1)$. De esa forma, se pueden obtener los datos de cada compañía (taxi asociados y número de servicios) con una complejidad baja, para luego almacenar esos datos en un diccionario que pueda ser ordenado con el algoritmo de ordenamiento timsort.

• ¿Cuál es la complejidad estimada del algoritmo implementado?

La complejidad temporal es $O(n \log n)$, siendo n el número de compañías. La complejidad espacial es $O(n)$.

Requerimiento 2

• ¿Qué TAD utilizaron en la solución del requerimiento?

Se utilizó un mapa ordenado y un mapa.

• ¿Por qué eligieron esa estructura de datos?

Porque el uso de un mapa ordenado nos permite consultar en un rango de fechas los taxis con más puntos con una complejidad logarítmica gracias al uso de un RBT. En este mapa ordenado los nodos son las fechas y dentro de cada nodo hay un mapa que contiene la información acerca de las millas, el total de dinero recibido y el total de servicios prestados de un taxi. Al igual que en el requerimiento anterior, el uso del TAD mapa nos permite retornar cada taxi en $O(1)$ y pasar esa información a un diccionario, para luego usar un algoritmo de ordenamiento y ver cuál tiene más puntos.

• ¿Cuál es la complejidad estimada del algoritmo implementado?

La complejidad temporal es $O(n \log n)$ por el uso de timsort, siendo n el número de taxis en la fecha o el rango de fechas seleccionado. La complejidad espacial es $O(n)$.

Requerimiento 3

- **¿Qué TAD utilizaron en la solución del requerimiento?**

Se utilizó un grafo dirigido y con peso.

- **¿Por qué eligieron esa estructura de datos?**

El uso de un grafo es la mejor manera de representar las relaciones entre lugares y gracias a los algoritmos de busca nos permite hacer búsquedas del costo entre estos sitios

- **¿Cuál es la complejidad estimada del algoritmo implementado?**

La complejidad temporal es $O(n \cdot v \cdot \log e)$ por el uso de Dijkstra y el recorrido del grafo en su totalidad, siendo n el número de viaje, v los vértices y e los arcos.