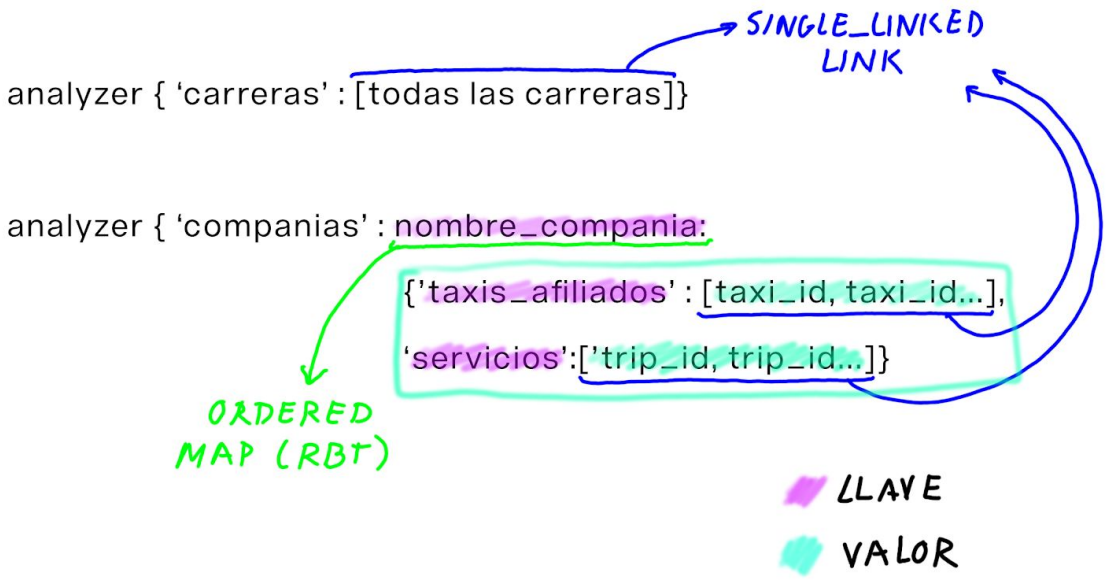
PROYECTO FINAL EDA - PARTE A

Marianna Velasco Zambrano 201921703

1. **¿Qué TAD utilizaron en la solución del requerimiento?**

La información de los archivos está guardada en el analizador (lista) y está dividida en dos etiquetas: las carreras y las compañías. Las carreras están almacenadas en una lista (single linked) y las compañías están almacenadas en un map ordenado (Árbol RBT), en donde la llave es el nombre de la compañía y el valor es un dict con dos llaves: los taxis afiliados y los servicios. El valor de la llave *"taxis_afiliados"* es una lista (single_linked) con los ids de los taxis por empresa. El valor de la llave *"servicios"* es una lista (single_linked) con los ids de cada servicio por compañía registrado en el archivo csv.

Es decir, la información está almacenada de la siguiente manera:



Para solucionar el Reto de los rankings y el número de taxis y servicios, cree otros dos mapas ordenados (BST). En uno guardaba los Servicios por Compañía y en otro los Taxis por Compañía. En estos maps, la llave era el número de taxis/servicios y el valor el nombre de la compañía correspondiente.

2. **¿Por qué eligieron esa estructura de datos?**

Elegí esta estructura de datos porque el requerimiento, al ser de ranking, podía ser fácilmente solucionado con el API de el ordered map. Con funciones como om.maxKey() o om.minKey(), fue rápido armar los rankings. En adición, también usé listas para almacenar datos a los que no tenía que acceder tan detalladamente. Como no pedían los ids de los servicios o los taxis, era

más sencillo utilizar listas para al final, usando el lt.size(), obtener el número total de taxis o de servicios.

Por otro lado, escogí esta estructura de datos, teniendo en cuenta el almacenamiento y la complejidad temporal, teniendo en cuenta que son archivos grandes con los que trabajamos.

3. **¿Cuál es la complejidad estimada del algoritmo implementado?**

La complejidad estimada del algoritmo en general es de O(n).