RAUL SANTIAGO INSUASTY - 202015512

JUAN ESTEBAN CURREA - 201922133

RAUL SANTIAGO INSUASTY - 202015512

JUAN ESTEBAN CURREA - 201922133

DOCUMENTO DE ENTREGA RETO 3

GRUPO 12

Link:

https://github.com/EDA2021-2-SEC02-G12/Reto3-G12.git

REQUERIMIENTO	COMPLEJIDAD
Req 1	O(log n)
Req 2 (Juan Currea)	O(n log n)
Req 3 (Raúl Insuasty)	O(log n)
Req 4	O(log n) ó O(k log n)
Req 5	O(log n)
Req 6 (bono)	------

Requerimiento 1:

Creamos un map con todas las ciudades y, después, dentro de cada ciudad, creamos un ordered map con la fecha de los respectivos avistamientos por ciudad. Para buscar dentro del map de las ciudades, la complejidad sería O(1) pero, para buscar dentro de las fechas (el ordered map dentro de cada ciudad) la complejidad es de O(log n). Para la construcción del mapa de ciudades, su complejidad temporal era de O(n) y, para el otro, la complejidad temporal es de O(log n) porque se usó un RBT.

Requerimiento 2:

Para construir el mapa de los segundos utilizamos un ordered map (RBT) con complejidad de búsqueda de O(log n) y espacial de O(n) en el peor caso. Dentro de cada llave se utilizó una lista para guardar los avistamientos y para ordenarla alfabéticamente se utilizó merge sort, el cual tiene como complejidad temporal, en su peor caso, O(n log n).

Requerimiento 3:

Para construir el mapa de los tiempos utilizamos un ordered map (RBT) pero, para poder encontrar datos en este, metimos un RBT dentro del mismo. Con esto, la complejidad temporal para el requerimiento sería de O(log n).

RAUL SANTIAGO INSUASTY - 202015512

JUAN ESTEBAN CURREA - 201922133

Requerimiento 4:

Se construye un mapa ordenado cuyas llaves son fechas (RBT) y, para acceder a un valor, su complejidad es de O(log n). Esta complejidad es para acceder a un solo dato, pero, debido a que debemos acceder a diferentes fechas (las que se encuentren en un rango dado), la complejidad sería de O(k log n) donde k es la cantidad de datos dentro del rango establecido.

Requerimiento 5:

Se construyeron 2 arboles principales, uno de longitud y otro de latitud. Para cada uno de estos, utilizando RBT, sabemos que la complejidad temporal de búsqueda es de O(log n). Para acceder a cada valor, se debía pasar por ambos arboles haciendo esto una complejidad temporal de O(log n)*O(log n).