Documento de análisis

Estructura de Datos y Algoritmos

Juan Camilo Neira Campos 201922746 j.neira@uniandes.edu.co

Daniel Dorado 201821010 df.dorado@uniandes.edu.co

Análisis de complejidad de cada uno de los requerimientos.

1. Requerimiento 1.

Para el requerimiento uno implementamos una Tabla de Hash, la cual almacena parejas de llave (ciudad) – valor. El valor de estas llaves es un árbol RBT que almacena los avistamientos por dateTime. Por tanto, la complejidad temporal de este requerimiento es logarítmica, ya que buscar un nodo en el árbol RBT tiene esta complejidad.

2. Requerimiento 2.

Este requerimiento tiene una complejidad de O(n). Esto se debe a que la función om.values() retorna, en este caso, una lista de listas. La solución fue recorrer los elementos de cada lista para almacenarlos en una lista plana. Entonces, si el rango abarca todos los datos posibles, se debe recorrer n elementos para imprimir los tres primeros y últimos elementos.

3. Requerimiento 3.

El requerimiento tiene una complejidad temporal O(n), puesto que retorna la llave más grande en O(1), encuentra el rango en O(log n), pero luego hace una “descomprensión” de la lista de valores que en el peor de los casos es O(n).

4. Requerimiento 4.

De manera similar al requerimiento anterior, este tiene complejidad O(n). Encuentra la menor llave en O(1), encuentra el rango de fechas en O(log n) y luego descomprime la lista de valores en O(n) en el peor de los casos.

5. Requerimiento 5.

Este también tiene complejidad O(n). Encuentra las llaves en cada rango en O(log n), pero debe descomprimir la lista de valores, lo cual es O(n) en el peor de los casos.

En caso de los requerimientos individuales, indicar quién lo implementó.

Requerimiento 2: Juan Camilo Neira.

Requerimiento 3: Daniel Dorado.