

Análisis del reto 3

Alumno: Pablo Pedreros Díaz – 202112491 - p.pedreros@uniandes.edu.co

Carga de datos.

En la carga de datos agregaremos varios índices que agilizarán enormemente tanto la implementación, como el tiempo de respuesta de los diferentes requerimientos. Para cada avistamiento que se cargue en los datos, se acoplará en algún punto de todos los índices. El primer índice funciona como un mapa que tiene como llaves las ciudades y como valores árboles ordenados que tienen como llaves los diferentes datetime de los avistamientos y como valores listas con todos los avistamientos en la ciudad de la primera llave con la duración de la segunda llave. El segundo índice funciona como un mapa ordenado que tiene como llaves las diferentes duraciones de los encuentros y como valores otro mapa ordenado con llaves (ciudad,país), ordenado por orden alfabético de la ciudad y luego del país. Los valores de este segundo árbol son listas que contienen todos los avistamientos de la duración de la primera llave en la combinación (ciudad,país) de la segunda llave. El tercer índice es un mapa ordenado que tiene como llaves las fechas en las que hubo encuentros, y como valores listas con todos los encuentros en esa fecha específica.

*Como hay que iterar sobre todas las llaves de los mapas se escogió que el primer mapa resolviera colisiones por linear probing y que los dos árboles fueran de tipo rojo-negro para mejorar la velocidad de las funciones.

La tabla con los tiempos de la carga de datos es la siguiente:

Porcentaje de datos (%)	Tiempo (ms)
5	343.75
10	781.25
20	1562.5
30	2484.375
50	3890.625
80	4640.625
100	5734.375

Requerimiento 1. (N = Número de valores datetime para el árbol de una ciudad; M = Número de ciudades en las que hubo avistamientos)

En el requerimiento 1, simplemente aprovecharemos el primer índice que creamos en el reto para buscar el árbol donde se encuentran los encuentros de la ciudad que buscamos ordenados por datetime, para de ahí imprimir los 3 primeros y los 3 últimos elementos. Esta primera parte será de complejidad $O(\log N)$ que es lo que toma una búsqueda. La segunda parte será buscar la ciudad con más avistamientos, para lo que iteraremos por las ciudades comparando sus size en busca del más grande, lo que nos daría una complejidad de $O(M)$. Así, la complejidad total del requerimiento sería de $O(M + \log N)$

El promedio de tiempo con la solución más eficiente luego de 3 intentos para el requerimiento está dado por la siguiente tabla, usando como ejemplo la ciudad de Las Vegas:

Porcentaje de datos (%)	Número de obras	Tiempo (ms)
5	7572	31.25
10	15008	46.875
20	29489	62.5
30	43704	93.75
50	71432	140.625
80	111781	187.5
100	138150	218.75

Requerimiento 2. (N = Número de posibles duraciones; M = Número de duraciones dentro del rango; P = Número de avistamientos dentro del rango)

En el requerimiento 2 usaremos el segundo índice que creamos en la carga de datos, de forma que hallaremos todas las llaves dentro del rango de duraciones usando reiteradamente la función ceiling hasta llegar al punto máximo. Suponiendo que la función ceiling tiene la complejidad de una búsqueda ($O(\log N)$), la complejidad de esta parte sería de $O(M \log N)$. Por lo tanto, el orden de complejidad del requerimiento será de $O(M \log N)$.

El promedio de tiempo luego de 3 intentos para el requerimiento por cada tamaño de archivo está dado por la siguiente tabla, usando como ejemplo un rango entre los 30.0 y 150.0 segundos:

Porcentaje de datos (%)	Número de artistas	Número de obras	Número de obras del artista	Tiempo (ms)
5	4996	7572	159	0
10	6656	15008	317	15.625
20	8724	29489	650	31.25
30	10063	43704	982	46.875
50	12137	71432	1633	62.5
80	14143	111781	2654	125.0
100	15223	138150	3337	156.25

Requerimiento 3. (N = Número de posibles fechas; M = Número de fechas dentro del rango; P = Número de avistamientos dentro del rango)

En el requerimiento 3 usaremos el tercer índice que creamos en la carga de datos de forma parecida que en el requerimiento 2. Nuevamente hallaremos todas las llaves dentro del rango, esta vez de fechas, usando reiteradamente la función ceiling hasta llegar a la fecha máxima. Suponiendo otra vez que la función ceiling tiene la complejidad de una búsqueda ($O(\log N)$), la complejidad de esta parte sería de $O(M \log N)$, pues la implementación hace un ceiling para cada fecha dentro del rango. Por lo tanto, el orden de complejidad del requerimiento será de $O(M \log N)$.

El promedio de tiempo luego de 3 intentos para el requerimiento por cada tamaño de archivo está dado por la siguiente tabla, usando como ejemplo un rango entre 1945-08-06 y 1984-11-15:

Porcentaje de datos (%)	Número de artistas	Número de obras	Número de obras del artista	Tiempo (ms)
5	4996	7572	159	0
10	6656	15008	317	15.625
20	8724	29489	650	15.625
30	10063	43704	982	15.625
50	12137	71432	1633	31.25
80	14143	111781	2654	46.875
100	15223	138150	3337	46.875

