

Análisis de resultados Reto 3

François Morales – f.moraless - 202211168

Santiago Navarrete – s.navarretev - 202211202

Gabriela García – g.garcias2 – 202210869

Introducción

El Reto 3 se basa en el manejo de un gran volumen de datos con el fin de realizar requerimientos útiles de búsqueda y análisis de ellos. Es importante resaltar que el tipo de estructura usado para este reto fue “Mapas Ordenados (Árboles)”, asimismo, el tipo de árbol utilizado fue “RBT” (Red Black Tree). Por otro lado, cuando así se requería, se utilizó únicamente la función de ordenamiento MERGE SORT, debido a su cualidad de estabilidad.

Por otro lado, con relación a los tiempos de ejecución, el reto se ejecutó en un solo computador cuyas características se describen a continuación:

- Procesador: AMD Ryzen 5 4600H with Radeon Graphics 3.00 GHz
- Memoria Ram: 16,0 GB (15,4 GB utilizable)
- Sistema Operativo: Sistema operativo de 64 bits, procesador x64 (Windows 11)

Requerimientos

Carga de datos

Para el análisis de los tiempos de ejecución del reto 3 se realizó la carga del 100% de los datos. Estos datos se agregaron a listas, mapas y árboles, según era necesario, ocupando una memoria de aproximadamente 112181.46 Kb. Con respecto a los árboles, se crearon de tipo “RBT” debido a su eficiencia en la búsqueda de información. En cuanto a cantidad, los datos se cargaron en: 2 listas, 5 árboles y 4 tablas de hash. Es importante aclarar que se usó un archivo adicional recuperado de internet con el fin de tener los datos geográficos de cada país.

Requerimiento 1 (Grupal)

- Complejidad

RETO 3
$O(n*m + N\log N + k)$

Pasos:

1. Recorrer los juegos filtrados por fecha para guardarlos en una lista y llevar la cuenta: $O(N*M)$
2. Organizar la lista por fecha de lanzamiento con merge sort: $O(N\log N)$

- Tiempo de ejecución

RETO 3
12,16 ms

Requerimiento 2 (Grupal)

- Complejidad

RETO 3
$O(N\log N + k)$

Pasos:

1. Organizar la lista de récords por jugador de acuerdo al tiempo demorado: $O(N\log N)$

- Tiempo de ejecución

RETO 3
0,08 ms

Requerimiento 3 (François)

- Complejidad

RETO 3
$O(n^2 + N\log N + k)$

Pasos:

1. Recorrer los juegos filtrados por intentos para guardarlos en una lista y llevar la cuenta: $O(N*M)$
2. Organizar las listas internas por tiempo de ejecución con merge sort: $O(N\log N)$
3. Organizar la lista externa por número de intentos con merge sort: $O(M\log M)$

- Tiempo de ejecución

RETO 3
32,67 ms

Requerimiento 4 (François)

- Complejidad

RETO 3
$O(n * m + N\log N + M\log M + k)$

Pasos:

1. Recorrer los récords filtrados por fecha para guardarlos en una lista y llevar la cuenta: $O(N*M)$
2. Organizar las listas internas por tiempo de ejecución con merge sort: $O(N\log N)$
3. Organizar la lista externa por fecha del mejor intento con merge sort: $O(M\log M)$

- Tiempo de ejecución

RETO 3
505,47 ms

Requerimiento 5 (Santiago)

- Complejidad

RETO 3
$O(n^2 + N\log N + k)$

Pasos:

1. Recorrer los récords filtrados por tiempo para guardarlos en una lista y llevar la cuenta: $O(N*M)$
2. Organizar las listas internas por fecha del mejor intento con merge sort: $O(N\log N)$
3. Organizar la lista externa por tiempo de ejecución con merge sort: $O(M\log M)$

- Tiempo de ejecución

RETO 3
120,26 ms

Requerimiento 6 (Grupal)

- Complejidad

RETO 3
$O(n*m + \text{segmentos} + k)$

Pasos:

1. Recorrer los récords filtrados por fecha (repetidos) para guardarlos en una lista y sacar la información pedida por el usuario: $O(N*M)$
2. Recorrido por el número de segmentos pedido por el usuario para crear los rangos del histograma: $O(\text{segmentos})$

- Tiempo de ejecución

RETO 3
221,75 ms

Requerimiento 7 (Grupal)

- Complejidad

RETO 3
$O(3*n + N\log N + k)$

Pasos:

1. Filtrar los récords por la plataforma pedida: $o(1)$

2. Recorrer los récords filtrados por plataforma para saber la cantidad de récords que no son de 'Misc' y agregarlos al diccionario cuenta: $O(N)$
3. Recorrer el diccionario cuenta para reasignar el valor dado como: el valor inicial sobre el total de récords que no son 'Misc': $O(N)$
4. Recorrer los récords para hacer la cuenta del stream_revenue por cada juego: $O(N)$
5. Organizar la lista mediante el algoritmo de ordenamiento merge sort.

- Tiempo de ejecución

RETO 3
468,57 ms

Requerimiento 8 (Bono Grupal)

- Complejidad

RETO 3
$O(n*m*o + n*m + k)$

Pasos:

1. Recorrer la lista de records filtrados por tiempo de ejecución y hacer recorrido por países si dicha casilla tiene más de un elemento: $O(N*M*O)$
2. Recorrer los países guardados y sus valores para realizar el mapa interactivo: $O(N*M)$

- Tiempo de ejecución

RETO 3
612,71 ms

Tabla de Tiempos de Ejecución

Requerimiento	Tiempo de ejecución (ms)
Carga de Datos	5367,02
1	12,16

2	0,08
3	32,67
4	505,47
5	120,26
6	221,75
7	468,57
8 (Bono)	612,71

Gráfica de Tiempos de Ejecución



Conclusiones

- El modelo de trabajo MVC permite la fácil implementación y modificación de la aplicación en general, adicionalmente, permite encontrar y solucionar errores de manera más fácil y eficiente
- Las complejidades teóricas del TAD árbol coinciden con los tiempos de ejecución, por ejemplo, la complejidad teórica más grande se da en el requerimiento 8 (BONO) y así lo demuestran sus resultados con respecto a los tiempos de ejecución de los demás requerimientos.

- Para este reto se implementaron arboles de tipo rojo y negro, ya que, estos al estar balanceados garantizan tiempos de búsqueda logarítmicos, debido a que, la longitud de una rama desde su raíz hasta su hoja es de $\log N$ lo cual implica un muy rápido acceso a la información.
- La función de ordenamiento más eficiente en este Reto fue "Merge Sort", ya que, al ser una función estable, no requiere que se lleve a cabo un análisis de como vienen los datos de entrada. Adicionalmente, obtuvo los mejores tiempos de ejecución para cada requerimiento.
- Teniendo en cuenta lo anteriormente mencionado y comparando el uso del TAD árbol con distintas estructuras de datos utilizadas como: mapas no ordenados y arreglos, los árboles rojo – negro representan la alternativa más rápida y eficiente para la búsqueda de datos entre un rango de parámetros.