

DOCUMENTO ANÁLISIS RETO 3

Requerimiento 2 - Javier Cerino Cod 202020873

Requerimiento 3 - Marco Zuliani Cod 202022412

Integrantes del grupo:

Marco Zuliani (código estudiante: 202022412, Usuario Uniandes: m.zuliani, Nombre de usuario Git: poloiva)

Javier Cerino (código estudiante: 202020873, Usuario Uniandes: j.cerino, Nombre de usuario Git: 2jc26)

Requerimiento 1:

- Alumnos que trabajaron en este requerimiento: Marco Zuliani y Javier Cerino.
- Análisis de complejidad:

$O(N)$

Para este requerimiento comenzamos cambiando el número ingresado en consola por el usuario por la característica de contenido que este representa [$O(\text{constante})$]. Luego se recorren todos los eventos de escucha cargados con el fin de introducirlos en un RBT utilizando como llave el valor asociado a la característica especificada y además introducir los artistas de cada evento en una tabla de hash, con el fin de conocer el número de artistas únicos, ya que este último se debe mostrar al usuario [Complejidad $O(N)$]. Como se tienen los datos de los eventos de escucha ordenados por la característica de contenido podemos utilizar la función Values de los mapas para obtener una lista con los eventos cuya característica se encuentre dentro del límite especificado por el usuario [Complejidad $O(N)$]. Finalmente, debido a que se trata de una lista de listas se realiza un ciclo en el que se van metiendo los eventos en una tabla de Hash con el fin de eliminar los eventos que no son únicos y también para que al final del ciclo se pueda conocer el número de llaves de la tabla que representa el número de eventos únicos dentro del rango indicado, ya que este es el segundo resultado a mostrar al usuario [Complejidad $O(N)$]. Se debe resaltar que el N indicado en las complejidades no es el mismo en todos los casos ya que el número de elementos a analizar puede variar con el paso del requerimiento. Por lo tanto, se puede concluir que el requerimiento tiene una complejidad de $O(N)$.

- Análisis de Tiempo:
El requerimiento se demora en el peor de los casos 13 segundos en obtener el resultado, en promedio se demora 9 segundos.
- Análisis de Memoria:
Para este requerimiento el uso de memoria se podría considerar eficiente ya que se hace uso de dos tablas de hash, una para almacenar los artistas únicos y otra para los eventos únicos, un RBT para organizar los eventos y una lista con los eventos dentro del rango indicado. Memoria usada: 36995.102 kB.

Requerimiento 2:

- Alumnos que trabajaron en este requerimiento: Javier Cerino.
- Análisis de complejidad:

$O(N)$

Para este requerimiento comenzamos recorriendo todos los eventos de escucha cargados con el fin de introducirlos en un RBT utilizando como llave el valor asociado a la característica 'energy' [Complejidad $O(N)$]. Luego se recorre la lista de listas obtenida usando la función de ValueSet de

los mapas ordenados para crear un RBT con los mismos datos, pero esta vez utilizando como llave el valor asociado a la característica 'danceability'[Complejidad $O(N)$]. A este nuevo RBT le aplicamos, al igual que al anterior, la función Values de los mapas ordenados para obtener una lista con los eventos cuya característica 'danceability' se encuentre dentro del límite especificado por el usuario[Complejidad $O(N)$]. Finalmente se recorre la lista de listas obtenida para introducir dentro de un map los eventos de tal manera que solo se encuentre un evento por cada 'track_id'[Complejidad $O(N)$]. Se debe resaltar que el N indicado en las complejidades no es el mismo en todos los casos ya que el número de elementos a analizar puede ir disminuyendo con el paso del requerimiento. Por lo tanto se puede concluir que el requerimiento tiene una complejidad de $O(N)$.

- **Análisis de Tiempo:**
El requerimiento se demora en el peor de los casos 14 segundos en obtener el resultado, en promedio se demora 3 segundos.
- **Análisis de Memoria:**
Para este requerimiento se hace uso de las mismas estructuras de datos que el Requerimiento 1 pero el uso de memoria tal vez podría estar algo por encima de lo esperado ya que se hace uso de un RBT extra para organizar los eventos, ya que en este requerimiento se debe filtrar por el rango de dos características de contenido. Memoria usada: 19338.062 kB.

Requerimiento 3:

- Alumnos que trabajaron en este requerimiento: Marco Zuliani.
- **Análisis de complejidad:**
 $O(N)$
Para este requerimiento comenzamos recorriendo todos los eventos de escucha cargados con el fin de introducirlos en un RBT utilizando como llave el valor asociado a la característica 'energy' [Complejidad $O(N)$]. Luego se recorre la lista de listas obtenida usando la función de ValueSet de los mapas ordenados para crear un RBT con los mismos datos, pero esta vez utilizando como llave el valor asociado a la característica 'danceability'[Complejidad $O(N)$]. A este nuevo RBT le aplicamos, al igual que al anterior, la función Values de los mapas ordenados para obtener una lista con los eventos cuya característica 'tempo' se encuentre dentro del límite especificado por el usuario[Complejidad $O(N)$]. Finalmente se recorre la lista de listas obtenida para introducir dentro de un map los eventos de tal manera que solo se encuentre un evento por cada 'track_id'[Complejidad $O(N)$]. Se debe resaltar que el N indicado en las complejidades no es el mismo en todos los casos ya que el número de elementos a analizar puede ir disminuyendo con el paso del requerimiento. Por lo tanto, se puede concluir que el requerimiento tiene una complejidad de $O(N)$.
- **Análisis de Tiempo:**
El requerimiento se demora en el peor de los casos 14 segundos en obtener el resultado, en promedio se demora 3 segundos.
- **Análisis de Memoria:**
Para este requerimiento se hace uso de las mismas estructuras de datos que el Requerimiento 1 pero el uso de memoria tal vez podría estar algo por encima de lo esperado ya que se hace uso

de un RBT extra para organizar los eventos, ya que en este requerimiento se debe filtrar por el rango de dos características de contenido. Memoria usada: 34574.727 kB.

Requerimiento 4:

- Alumnos que trabajaron en este requerimiento: Javier Cerino y Marco Zuliani.
- Análisis de complejidad:
O(N)

Para este requerimiento comenzamos cambiando el número ingresado en consola por el usuario por el género que este representa y obteniendo el tiempo mínimo y máximo de este género[O(constante)]. Luego se recorren todos los eventos de escucha cargados con el fin de introducirlos en un RBT utilizando como llave el valor asociado a la característica 'tempo' y además introducir los artistas de cada evento en una tabla de hash, con el fin de conocer el número de artistas únicos ya que se debe mostrar al usuario[Complejidad O(N)]. Como se tienen los datos de los eventos de escucha ordenados por la característica de contenido podemos utilizar la función Values de los mapas para obtener una lista con los eventos cuyo 'tempo' se encuentre dentro del límite obtenido anteriormente[Complejidad O(N)]. Debido a que se trata de una lista de listas se realiza un ciclo para conocer el tamaño de las listas internas, que en su totalidad representa el número de eventos dentro del rango indicado, ya que este es el segundo resultado a mostrar al usuario[Complejidad O(N)]. Finalmente, se imprimen 10 'artist_id' aleatorios o en caso de que el número máximo de artistas sea menor que 10 se imprimen todos[Complejidad O(Constante)]. Se debe resaltar que el N indicado en las complejidades no es el mismo en todos los casos ya que el número de elementos a analizar puede ir disminuyendo con el paso del requerimiento. Por lo tanto, se puede concluir que el requerimiento tiene una complejidad de O(N). En este Requerimiento se debe tener en cuenta que este proceso se repite por cada género que desee consultar el usuario.

- Análisis de Tiempo:
El requerimiento se demora en el peor de los casos 12 segundos, pero el tiempo promedio en consultar para los géneros por default es de 2 segundos y esto multiplicado por cada género que se desee consultar.
- Análisis de Memoria:
Para este requerimiento el uso de memoria se podría considerar eficiente ya que se hace uso de una tabla de hash para almacenar los artistas únicos, un RBT para organizar los eventos y una lista con los eventos dentro del rango indicado. Memoria usada: 32713.504 kB.

Requerimiento 5:

- Alumnos que trabajaron en este requerimiento: Javier Cerino y Marco Zuliani.
- Análisis de complejidad:
O(N) en el peor caso O(NlogN)

Para comenzar con el requerimiento se le pide al usuario ingresar las horas de interés que desea consultar, para tener una mayor facilidad al momento de comparar transformamos todas las horas de interés a una misma unidad de medida, segundos, para que así sean las comparaciones mucho más simples. Luego recorremos todos los eventos a un RBT con el fin de utilizar la función

Values de los mapas para obtener una lista con los eventos que correspondan con las horas. [Complejidad $O(N)$]. Sucederá un proceso similar con el otro archivo `user_track` aunque se omitirá un recorrido total inicial de la lista debido a que este archivo fue cargado directamente en un RBT con las fechas. Luego se recorren todos los eventos de escucha cargados con el fin de introducirlos en un RBT utilizando como llave el valor asociado a la característica 'tempo' [Complejidad $O(N)$]. Como se tienen los datos de los eventos de escucha ordenados por la característica de contenido podemos utilizar la función Values de los mapas para obtener una lista con los eventos cuyo 'tempo' se encuentre dentro del límite obtenido anteriormente [Complejidad $O(N)$]. Debido a que se trata de una lista de listas se realiza un ciclo para conocer el tamaño de las listas internas, que en su totalidad representa el número de eventos dentro del rango indicado, ya que este es el segundo resultado a mostrar al usuario [Complejidad $O(N)$]. Se debe resaltar que el N indicado en las complejidades no es el mismo en todos los casos ya que el número de elementos a analizar puede ir disminuyendo con el paso del requerimiento. Por lo tanto, esta parte del requerimiento tiene una complejidad de $O(N)$. Este proceso se repite por cada género existente en el sistema. Una vez concluido este proceso se procede a almacenar en memoria los elementos correspondientes al género con más reproducciones. Una vez concluido esto se procede a recorrer todos los elementos presentes en la tabla de hash que contiene los datos mayores [Complejidad $O(N)$] para crear así una tabla de hash que permita identificar los unique tracks presentes en el total de reproducciones. Para ahorrar recorridos se procede a buscar en los datos con tracks brindados por el RBT del segundo archivo para agregar los hashtags presentes en los eventos de escucha de interés. Una vez conseguido el hashtag perteneciente a cada evento de escucha se procede a buscar en el archivo de sentimientos estos hashtags gracias a que este archivo se cargó como una tabla de hash con llave los hashtags. Encontrar un hashtag en la tabla de hash dará como resultado una complejidad constante. Una vez creada la tabla de hash final que contiene los tracks únicos con sus respectivos hashtags y su average promedio se procede a realizar un nuevo sondeo lineal donde se comienzan a recorrer todos los eventos de esta nueva tabla de hash para así proceder a crear una lista de tuplas de tuplas que será la utilizada para devolver la respuesta final. Una vez realizada esta lista se procede a realizar un ordenamiento con Merge para ahorrar tiempo con parámetro el número de hashtags para que los mayores se encuentren en las primeras posiciones de las listas para que la impresión por consola sea mucho más fructífera. En conclusión en el peor de los casos el que determinará la complejidad del algoritmo puede llegar a ser el método Merge pero en el caso promedio el $n \log n$ de veces será mucho más inferior del N total de archivos.

- **Análisis de Tiempo:**

El requerimiento se demora alrededor de 7 segundos en promedio. No obstante, encontramos ciertas dificultades al correr el peor de los casos.

- **Análisis de Memoria:**

Para este requerimiento el uso de memoria se podría considerar eficiente ya que se hace uso de unas tablas de hash para almacenar los artistas únicos, unos RBT para organizar los eventos y unas listas con los eventos dentro del rango indicado. Memoria Usada: 7216.298 kB