

Estudiante B: Andrés Felipe Gómez García, af.gomezg1@uniandes.edu.co, 202021189

Estudiante A: Tomás Bedoya Calixto, t.bedoya@uniandes.edu.co, 202020689

## **ANÁLISIS DE COMPLEJIDAD**

### **RETO 1**

#### **Requerimiento 1:**

1. Crear lista auxiliar:  $O(1)$
2. Obtener el ID de la categoría de la lista en el catálogo de IDs de categorías:  $O(n)$
3. Extraer los videos cuyas áreas de país e ID de categoría son iguales a las solicitadas e insertarlos en la lista auxiliar:  $O(n)$
4. Se realiza Quicksort sobre la lista auxiliar con respecto al mayor número de visitas por video:  $O(n\log(n))$
5. Imprimir el número de videos solicitado:  $O(n)$

**Complejidad General:**  $O(n\log(n))$

**Complejidad Peor Caso (Poco Probable):**  $O(n^2)$

#### **Requerimiento 2:**

1. Crear lista auxiliar:  $O(1)$
2. Extraer los videos cuyas áreas de país son iguales al solicitado e insertarlos en la lista auxiliar:  $O(n)$
3. Se realiza Quicksort sobre la lista auxiliar con respecto al ID de cada video (los videos con el mismo nombre quedan consecutivos en la lista):  $O(n\log(n))$
4. Se cuentan las repeticiones de los videos con el mismo nombre (mismo video) ignorando las fechas repetidas que podrían salir (no se cuenta un mismo día 2 veces). Se conserva el video con más repeticiones (días trending) y se imprime:  $O(n)$

**Complejidad General:**  $O(n\log(n))$

**Complejidad Peor Caso (Poco Probable):**  $O(n^2)$

**Requerimiento 3:**

1. Crear lista auxiliar:  $O(1)$
2. Obtener el ID de la categoría de la lista en el catálogo de IDs de categorías:  $O(n)$
3. Extraer los videos cuyas áreas de ID de categoría son iguales al solicitado e insertarlos en la lista auxiliar:  $O(n)$
4. Se realiza Mergesort sobre la lista auxiliar con respecto al ID de cada video (los videos con el mismo nombre quedan consecutivos en la lista):  $O(n\log(n))$
5. Se cuentan las repeticiones de los videos con el mismo nombre (mismo video) ignorando las fechas repetidas que podrían salir (no se cuenta un mismo día 2 veces). Se conserva el video con más repeticiones (días trending) y se imprime:  $O(n)$

**Complejidad General:**  $O(n\log(n))$

**Requerimiento 4:**

1. Crear lista auxiliar:  $O(1)$
2. Extraer los videos cuyas áreas de país y tag son iguales (o en el caso de tag, contienen) a las solicitadas e insertarlos en la lista auxiliar:  $O(n)$
3. Se realiza Quicksort sobre la lista auxiliar con respecto al mayor número de likes por video:  $O(n\log(n))$
4. Imprimir el número de videos solicitado revisando que no se impriman videos repetidos:  $O(n)$

**Complejidad General:**  $O(n\log(n))$

**Complejidad Peor Caso (Poco Probable):**  $O(n^2)$

## ANÁLISIS DE COMPLEJIDAD

### RETO 2

#### Requerimiento 1:

1. Crear lista auxiliar:  $O(1)$
2. Obtener el ID de la categoría del map de categorías:  $O(1)$
3. Extraer del map de países la lista de posiciones de los videos en la lista de videos que corresponden a un país:  $O(1)$
4. A partir de las posiciones, extraer los videos de la lista de videos cuyo ID de categoría sea igual al solicitado e insertarlos en la lista auxiliar:  $O(n)$
5. Se realiza Quicksort sobre la lista auxiliar con respecto al mayor número de visitas por video:  $O(n\log(n))$
6. Imprimir el número de videos solicitado:  $O(n)$

**Complejidad General:**  $O(n\log(n))$

**Complejidad Peor Caso (Poco Probable):**  $O(n^2)$

#### Requerimiento 2:

1. Crear lista auxiliar:  $O(1)$
2. Extraer del map de países las posiciones de los videos en la lista de videos que corresponden a un país:  $O(1)$
3. A partir de las posiciones, extraer los videos de la lista de videos e insertarlos en la lista auxiliar:  $O(n)$
4. Se realiza Quicksort sobre la lista auxiliar con respecto al ID de cada video (los videos con el mismo nombre quedan consecutivos en la lista):  $O(n\log(n))$
5. Se cuentan las repeticiones de los videos con el mismo nombre (mismo video) ignorando las fechas repetidas que podrían salir (no se cuenta un mismo día 2 veces). Se conserva el video con más repeticiones (días trending) y se imprime:  $O(n)$

**Complejidad General:**  $O(n\log(n))$

**Complejidad Peor Caso (Poco Probable):**  $O(n^2)$

#### Requerimiento 3:

1. Crear lista auxiliar:  $O(1)$

2. Obtener el ID de la categoría del map de categorías:  $O(1)$
3. Extraer del map de IDs de categorías la lista de posiciones de los videos en la lista de videos que corresponden a la categoría dada:  $O(1)$
4. A partir de las posiciones, extraer los videos de la lista de videos e insertarlos en la lista auxiliar:  $O(n)$
5. Se realiza Mergesort sobre la lista auxiliar con respecto al ID de cada video (los videos con el mismo nombre quedan consecutivos en la lista):  $O(n\log(n))$
6. Se cuentan las repeticiones de los videos con el mismo nombre (mismo video) ignorando las fechas repetidas que podrían salir (no se cuenta un mismo día 2 veces). Se conserva el video con más repeticiones (días trending) y se imprime:  $O(n)$

**Complejidad General:**  $O(n\log(n))$

#### **Requerimiento 4:**

1. Crear lista auxiliar:  $O(1)$
2. Extraer del map de países la lista de posiciones de los videos en la lista de videos que corresponden a un país:  $O(1)$
3. A partir de las posiciones, extraer los videos de la lista de videos que incluyan el tag solicitado e insertarlos en la lista auxiliar:  $O(n)$
4. Se realiza Quicksort sobre la lista auxiliar con respecto al mayor número de likes por video:  $O(n\log(n))$
5. Imprimir el número de videos solicitado revisando que no se impriman videos repetidos:  $O(n)$

**Complejidad General:**  $O(n\log(n))$

**Complejidad Peor Caso (Poco Probable):**  $O(n^2)$

## DATOS DE RENDIMIENTO

<b>Reto 1</b>	Tiempo de Ejecución [ms]	Consumo de Datos [kB]
Carga de Datos	50014.277	542028.302
Requerimiento 1	1272.790	51.509
Requerimiento 2	4928.490	352.449
Requerimiento 3	4468.750	370.391
Requerimiento 4	1156.250	68.817

<b>Reto 2</b>	Tiempo de Ejecución [ms]	Consumo de Datos [kB]
Carga de Datos	72535.957	713636.020
Requerimiento 1	420.318	50.849
Requerimiento 2	4546.875	343.574
Requerimiento 3	3437.500	369.754
Requerimiento 4	562.500	67.505

Al referirse a las medidas de rendimiento del reto 1 y del reto 2, se observa fácilmente que la rápida búsqueda dada un parámetro permitida por los maps posibilita que los requerimientos se realicen en un tiempo notablemente menor. Adicionalmente, como la información de los videos se guarda una sola vez en la lista de videos y los maps contienen referencias a las posiciones de los videos correspondientes en dicha lista, el uso de maps, además de favorecer el tiempo de ejecución, lo hace sin comprometer el uso de memoria RAM del dispositivo. Cabe recalcar que la reducción temporal más significativa se da gracias a que los maps permiten obtener los videos a los cuales se les va a aplicar el sorting sin necesidad de recorrer todos los datos.