

Documento de análisis.

Pablo Alejandro Guatibonza Briceño

P.guatibonza@uniandes.edu.co

202014393

Kaled Andres Correal Sosa

k.correal@uniandes.edu.co

201920829

Requerimiento 1 (Grupal)

1. En primer lugar se hace un recorrido de las categorías para determinar su id. Se puede considerar como $O(n)$, pero como los datos son tan pequeños, el tiempo se puede despreciar
2. Despues, se hace un recorrido de toda la lista de videos, donde se filtran si cumplen la condición de ser de x país y tener z id. Teniendo una complejidad de $O(n)$
3. Se reduce la lista a una que solo tenga los videos con las características mencionadas anteriormente. $O(1)$
4. Finalmente, se hace un ordenamiento de los datos según la cantidad de vistas, mediante el algoritmo deseado por el usuario. Sin embargo, escogimos el mergeSort con complejidad $O(n \log n)$

Requerimiento 2(Pablo Alejandro Guatibonza Briceño)

1. Se reduce la lista a una que solo contenga los videos con el país deseado. Presenta complejidad $O(n)$
2. La lista resultante es ordenada para que los videos con igual id esten consecutivos, por medio de un MergeSort. $O(n \log n)$
3. Despues se hace un recorrido de todos los datos, repitiéndose cada iteración según el numero de datos iguales que haya a la derecha del video. Aproximadamente, la complejidad es $O(n)$

Requerimiento3(Kaled Andres Correal Sosa)

Para completar este requerimiento:

- 1.Se redujo la lista para que tenga solo los videos con la categoría seleccionada. Presenta una complejidad de $O(n)$

2. Esa lista es ordenada para que los videos con el mismo título estén consecutivos por medio de un MergeSort que tiene una complejidad de $O(N \log N)$
- 3, Consecutivamente se recorre nuevamente los datos de la lista organizada repitiéndose cada iteración según el número de datos iguales que existan a la derecha del video su complejidad es $O(n)$

Requerimiento 4

1. Se reduce la lista a una con los videos que su país sea el elegido por el usuario. $O(n)$
2. La lista resultante es filtrada para crear otra nueva con los videos que tienen un tag específico. $O(n)$
3. Los videos son organizados por la cantidad de likes ($O(n \log n)$) y luego por su id ($O(n \log n)$) mediante un mergeSort.
4. Finalmente, se filtran los videos para que no aparezcan repetidos. ($O(n \log n)$)