

Reto 2 Ordenes de Complejidad

Natali Mercado 202012682 n.mercado@uniandes.edu.co

Gabriel Villabon 202013898 g.villabon@uniandes.edu.co

Requerimiento 1

Este requerimiento se divide en 2 funciones principales una en el model y la otra en la view:

1. La primera función ubicada en el model utiliza el map catalog['videosCountry'] que está ubicado en el catalog, busca la llave que sea igual al país pasado por parámetro y retorna la lista con los videos, su orden de complejidad es $O(n)$. Posterior a esto, se itera sobre la lista con los videos por país y se filtra la lista por category, lo cual da un orden de complejidad de $O(m)$. Por último, organiza la lista por medio de quick sort, el cual tiene un orden de complejidad de $O(s \cdot \lg(s))$ generalmente o de $O(2s)$. Lo que da como resultado un orden de complejidad de $O(m)$ como máximo.
 $O(n)$ = cantidad de llaves
 $O(m)$ = cantidad de videos dentro de la lista que contiene los videos por country
 $O(s)$ = cantidad de videos dentro de la lista filtrada por país
2. La segunda función ubicada en la view utiliza la lista que se organiza en model y retorna los n primeros videos que pide el usuario lo cual da como complejidad un orden de $O(n)$ como máximo.
 $O(n)$ = cantidad n videos requeridos

Respecto al anterior reto, la cantidad de elementos a recorrer en la lista en este reto es menor, por ende, esta función es más rápido en términos de tiempo que la del reto 1.

Requerimiento 2

Este requerimiento se divide en 2 funciones importantes una en el model y una view:

1. La primera función ubicada en el model utiliza el map catalog['videosCountry'] que está ubicado en el catalog, busca la llave que sea igual al país pasado por parámetro y retorna la lista con los videos por el country, su orden de complejidad es $O(n)$. Posterior a esto, organiza la lista con Shell $O(m^{3/2})$ o $O(m \cdot \lg(m))$ y finalmente itera sobre la lista con los videos comparando los títulos de los videos y los días de trending, esto tiene un orden de complejidad de $O(m)$. El mayor orden seria $O(m)$.
 $O(n)$ = cantidad de llaves
 $O(m)$ = cantidad de videos
2. La segunda función de view muestra los valores pedidos y tiene orden de complejidad de $O(1)$.

Realizado por Natali Mercado

Respecto al reto anterior, el orden de complejidad es menor para este caso, ya que el recorrido anterior es $O(n)$ y el actual es $O(m)$.

$O(n)$ = la cantidad de videos totales.

$O(m)$ = la cantidad de videos por país.

Requerimiento 3

Este requerimiento se divide en 2 funciones importantes una en el model y una view:

1. La primera función ubicada en el model utiliza el map catalog['videosCategory'] que está ubicado en el catalog, busca la llave que sea igual al category pasado por parámetro y retorna la lista con los videos por el category, su orden de complejidad es $O(n)$. Posterior a esto, organiza la lista con quick sort $O(m \cdot \lg(m))$ o $O(2m)$ y finalmente itera sobre la lista con los videos comparando los títulos de los videos y los días de trending, esto tiene un orden de complejidad de $O(m)$. El mayor orden seria $O(2m)$.

$O(n)$ = cantidad de llaves

$O(m)$ = cantidad de videos

2. La segunda función de view muestra los valores pedidos y tiene orden de complejidad de $O(1)$.

Realizado por Gabriel Villabon

Respecto al reto anterior, el orden de complejidad es menor para este caso, ya que el recorrido anterior es $O(n)$ y el actual es $O(m)$.

$O(n)$ = la cantidad de videos totales.

$O(m)$ = la cantidad de videos por país.

Requerimiento 4

Este requerimiento se divide en 2 funciones importantes una en el model y otra en la view:

1. La primera función ubicada en el model utiliza el map catalog['videosCountry'] que está ubicado en el catalog, busca la llave que sea igual al país pasado por parámetro y retorna la lista con los videos, su orden de complejidad es $O(n)$. Posterior a esto, se itera sobre la lista con los videos por país y se filtra la lista por tag, lo cual da un orden de complejidad de $O(m)$. Por último, organiza la lista por medio de quick sort, el cual tiene un orden de complejidad de $O(s \cdot \lg(s))$ generalmente o de $O(2s)$. Lo que da como resultado un orden de complejidad de $O(m)$ como máximo.

$O(n)$ = cantidad de llaves

$O(m)$ = cantidad de videos dentro de la lista que contiene los videos por country

$O(s)$ = cantidad de videos dentro de la lista filtrada por tag.

2. La segunda función ubicada en la view utiliza la lista que se organiza en model y retorna los n primeros videos que pide el usuario lo cual da como complejidad un orden de $O(n)$ como máximo.

$O(n)$ = cantidad n videos requeridos