

RETO 1

Juan Esteban Rodríguez Ospino j.rodriquezo 202011171

Juan Sebastián Peña Muñoz j.penam 202013078

Germán Alberto Rojas Cetina g.rojasc 202013415

2.2

C)

Requerimiento N°2

Revisando nuestro algoritmo en el requerimiento N°2 pudimos observar que, nuestro ordenamiento Shell Sort es el más adecuado para optimizar el proceso de carga. No obstante, el uso de funciones del TAD List en un arreglo son $O(1)$. Es por ello que, el orden de crecimiento temporal de nuestra función dos es: $O(N^{3/2})$.

Requerimiento N°3

Nuestro algoritmo en el requerimiento N°3 tiene un orden de crecimiento temporal de $O(N)$. Lo anterior se da porque, se hace uso de funciones del TAD List dentro de un ciclo. Sin embargo, en el peor de los casos tendrá que recorrer todo el arreglo en busca de que elemento a escoger.

Requerimiento N°4 (**BONO**)

El algoritmo en el requerimiento N°4 tiene un orden de crecimiento temporal de $O(N^2)$. Por otra parte, en el peor de los casos tendrá que recorrer ambos arreglos simultáneamente buscan coincidencia en el ID.

Requerimiento N°5

El algoritmo en el requerimiento N°5 tiene un orden de crecimiento temporal de $O(N)$. Puesto que, en el peor de los casos tendrá que buscar dentro del arreglo que el género coincida con el del usuario. Aunque todas las funciones implementadas del TAD List sean $O(1)$ el ciclo tendrá una complejidad $O(N)$.

Requerimiento N°6

Después de intentar con todos los sortings propuestos en el laboratorio se decidió implementar Shell Sort, puesto que este es el más óptimo para desarrollar la tarea. El orden de crecimiento es $O(N^{3/2})$ en el peor caso.

- Se decidió tomar la estructura de datos “ARRAY_LIST” puesto que es la más apropiada para resolver las funciones del TAD list en menor tiempo.