

Documento de análisis reto 2.

Integrantes.

- Ehimar Andres Varga Malaver, e.vargasm@uniandes.edu.co, 202014902

Requerimientos.

1. Requerimiento 1

La complejidad es de $O(n \log(n))$ debido a el algoritmo de ordenamiento usado, Merge sort, que tiene como peor caso la complejidad ya descrita, el resto de las operaciones como asignaciones o comparaciones se hacen en $O(1)$.

2. Requerimiento 2

La complejidad de este requerimiento es $O(n)$; sin embargo, el n es cerca de un octavo de la totalidad de los datos, por lo que la complejidad, aunque parezca alta, es muy baja en tiempos de respuesta.

3. Requerimiento 3

La complejidad de este requerimiento es $O(n)$; sin embargo, el n es cerca de un sexto de la totalidad de los datos, por lo que la complejidad, aunque parezca alta, es muy baja en tiempos de respuesta.

4. Requerimiento 4

La complejidad de este requerimiento es $O(n \log(n))$ debido al algoritmo de ordenamiento usado, Merge sort, que tiene como peor caso la complejidad ya descrita; sin embargo, al momento de buscar los elementos que coinciden con los términos de búsqueda el algoritmo puede llegar a ser $O(n)$; sin embargo, los tiempos de respuesta son muy rápidos.

Tiempos de carga y respuesta.

- La carga de la totalidad de los datos demora en valor promedio 58 segundos, y el total de memoria consumida es de 2.27 Gbs de memoria.
- Requerimiento 1 presenta un tiempo de respuesta menor a 1 segundo.
- Requerimiento 2 presenta un tiempo de respuesta menor a 0.1 segundos.
- Requerimiento 3 presenta un tiempo de respuesta menor a 0.1 segundos.
- Requerimiento 4 presenta un tiempo de respuesta menor a 1.5 segundos.

Comparación de rendimiento y complejidad respecto al reto 1.

- Requerimiento 1: la complejidad de este requerimiento era de $O(n^2)$, sin embargo, los tiempos de respuesta varían por 2 segundos, siendo el requerimiento 2, 2 segundos mas rápido.
- Requerimiento 2: la complejidad de este requerimiento era de $O(n^2)$ por que se debía aplicar el ordenamiento que estaba con Quicksort, siendo esa su complejidad en el peor caso, sin embargo, los tiempos de respuesta eran similares.
- Requerimiento 3: la complejidad de este requerimiento era de $O(n^2)$ por que se debía aplicar el ordenamiento que estaba con Quicksort, siendo esa su complejidad en el peor

caso, sin embargo, en reto 2, es de $O(n)$ por lo tanto, los tiempos de respuesta eran muy similares.

- Requerimiento 4: la complejidad de este requerimiento era de $O(n^3)$ ya que el algoritmo de ordenamiento usado era Quicksort y además se tenía que recorrer la lista, lo cual tiene una complejidad de $O(n)$, sin embargo, en el reto 2 este tiempo paso a ser de $O(n \log n)$ ya que la tabla de hash permite obtener información en $O(1)$.