



Instituto Tecnológico y de Estudios Superiores de Monterrey

Programación de estructuras de datos y algoritmos fundamentales (Gpo 602)

Act 1.3 - Actividad Integral de Conceptos Básicos y Algoritmos Fundamentales

Adela Alejandra Solorio Alcázar A01637205

Marcela Beatriz De La Rosa Barrios A01637239

Luz Patricia Hernández Ramírez A01637277

07/09/2023

Actividad Integral de Conceptos Básicos y Algoritmos Fundamentales

Los algoritmos de ordenamiento son una herramienta básica en la computación que permite, como lo dice su nombre, ordenar o arreglar una serie de datos en órdenes secuenciales; existen distintos algoritmos que se pueden aplicar que son clasificados de acuerdo a su complejidad computacional, su uso de memoria, su estabilidad, si son para procesamiento serial o paralelo y si son comparativos o no comparativos, el método o algoritmo que se aplicará debe ser elegido dependiendo la acción que se quiera hacer y considerando todas esas características de cada uno (Navarro, 2020).

De igual manera, los algoritmos de búsqueda son herramientas de gran utilidad en la computación que permiten encontrar y analizar datos en grandes series; también existen diversos algoritmos que varían por sus distintas complejidades, eficiencias, y tamaños del dominio de búsqueda (Alfaro, s.f.).

En esta actividad se buscó aprovechar estos algoritmos para crear un programa que sea capaz de ordenar una serie de datos o registros que fue dada y de permitirle al usuario consultar los registros entre fechas que especifique y las reciba de forma ordenada, a continuación explicamos con más detalle cómo lo implementamos:

El código que se creó lee datos de un archivo llamado "bitacora.txt", los procesa y los ordena por fecha utilizando el algoritmo de ordenamiento Bubble Sort y realiza una búsqueda binaria iterativa para encontrar registros dentro de un rango de fechas específico.

Se implementó el algoritmo de ordenamiento Bubble Sort para ordenar los registros por hora, el cual funciona comparando e intercambiando elementos adyacentes si están en el orden incorrecto. Esto se repite hasta que no se deba de intercambiar ningún otro dato de la lista y todos estos se encuentren ordenados. Es por esto que la complejidad del algoritmo Bubble Sort, en el peor caso, tiene una complejidad de $O(n^2)$ y en el mejor caso, si la lista ya está ordenada, la complejidad es $O(n)$.

Para la realización de búsqueda del rango de fechas se utilizó la búsqueda binaria iterativa, debido a que reduce a la mitad el espacio de búsqueda en cada iteración. La complejidad de este algoritmo es de $O(\log n)$.

Una vez que los datos ya fueron ordenados, se acomodan en un nuevo documento llamado "ordenados.txt" y de ahí se llaman los datos para que se impriman en consola cuando el usuario ingresa un rango de fechas.

Consideramos que los algoritmos de ordenamiento y búsqueda elegidos eran los más pertinentes para lo que se buscaba que realizara el programa, pero a continuación se describen otras opciones:

Búsqueda:

- Búsqueda lineal o secuencial: Comprueba secuencialmente cada elemento de los datos hasta que encuentra lo que busca o hasta que ya comparó todos los elementos.
 - Complejidad: $O(n)$.

(Ponce, 2021)

Ordenamiento:

- Quicksort: Se basa en la técnica divide y vencerás pues divide los datos a ordenar en pequeñas partes y así los ordena de manera más sencilla.
 - Complejidad (para el caso promedio): $O(n \log n)$.
- Insertion sort: Compara una key con los elementos anteriores y, si son mayores, mueve el elemento anterior a la siguiente posición, yendo de izquierda a derecha inicialmente.
 - Complejidad (para el caso promedio): $O(n^2)$.

(BlackeyeB, 2023)

Como se puede observar con esta aplicación y con lo mencionado anteriormente, los algoritmos de ordenamiento y búsqueda son de gran utilidad y es de gran importancia elegir de manera correcta los algoritmos que mejor se apliquen a la situación.

Reflexiones individuales

Luz: Los algoritmos que elegimos fueron seleccionados por su afinidad para el tipo de datos, siendo bubble sort simple de añadir y con poca necesidad de memoria extra, mientras que la búsqueda binaria es muy conveniente para buscar en listas ordenadas partiendo de rangos. Dados los análisis que tuvimos que llevar a cabo para plantear el programa y decidir qué algoritmos usar, puedo concluir que es muy importante ser creativos y procurar encontrar la solución más práctica, de modo que al implementar los algoritmos sea sencilla la integración de todos los elementos del programa.

Marce: En este trabajo se pudo realizar una aplicación de los algoritmos de búsqueda y ordenamiento en el contexto de la separación de bases de datos extensas. Para esto se utilizó Bubble Sort para el ordenamiento y Binary Search para la búsqueda, los cuales

ayudaron a organizar y crear un nuevo “txt” con la base de datos ordenada en base a la fecha. Con esto se puede comprender la importancia de aplicar adecuadamente los algoritmos de búsqueda y ordenamiento en la gestión de datos, además de la importancia en la elección de algoritmos debido a eficiencia y rendimiento de operaciones para el procesamiento de datos.

Adela: Para realizar este trabajo, primero validamos cuales serían los mejores algoritmos para resolver la problemática y la mejor manera de incorporarlos en el programa como tal; al desarrollar el programa tuvimos diversos contratiempos con la impresión de los datos ordenados y con la creación del archivo con los datos ordenados, pero, gracias a que cambiamos nuestra manera de leer/almacenar los datos, pudimos resolverlo.

Referencias

Alfaro, T. (s.f.). Algoritmos de Búsqueda y Ordenamiento. Departamento de Informática Universidad Técnica Federico Santa María. Recuperado de <https://www.inf.utfsm.cl/~noell/IWI-131-p1/Tema8b.pdf>

BlackeyeB. (2023). Algoritmos de ordenación explicados con ejemplos en JavaScript, Python, Java y C++. *Free Code Camp*. Recuperado de <https://www.freecodecamp.org/espanol/news/algoritmos-de-ordenacion-explicados-con-ejemplos-en-javascript-python-java-y-c/>

Navarro, A. (2020). Algoritmos de ordenamiento. *Junco TIC*. Recuperado de <https://juncotic.com/algoritmos-de-ordenamiento/>

Ponce, J. (2021). Algoritmos de búsqueda. *Jahaziel Ponce*. Recuperado de <https://jahazielponce.com/algoritmos-de-busqueda/>