



Tecnológico de Monterrey

Campus Monterrey

Reflexión Individual

Act 1.4 - Actividad Integral - Conceptos Básicos y Algoritmos Fundamentales

(Evidencia Competencia)

Paula Guerrero Martínez
A00839099

Dr. Eduardo Arturo Rodríguez Tello
Programación de estructuras de datos y algoritmos fundamentales
(Gpo 573)

19 de enero del 2025

Importancia y eficiencia del uso de los diferentes algoritmos de ordenamiento y búsqueda en una situación problema de esta naturaleza.

Dentro del desarrollo de aplicaciones para el análisis de bitácoras se manejan miles de registros de acceso; por ende, es súper importante manejar una eficiencia algorítmica. En esta actividad, implementamos una solución en C++, con programación orientada a objetos, para gestionar los registros confirmados por fecha (mes, día y hora). El objetivo de la actividad fue ordenar estos registros cronológicamente y permitir que el usuario realizara búsquedas mediante diferentes rangos de fechas.

Algoritmos de ordenamiento

Para ordenar los datos utilizamos el método de burbuja y merge sort. El método de burbuja fue sencillo de implementar, pero ineficiente con una complejidad de $O(n^2)$. En las pruebas realizadas con los datos de la bitácora proporcionados encontramos un número excesivo de operaciones ya que requirió 10,504 ms para finalizar: el algoritmo realizó 141,262,836 comparaciones y 70,183,700 intercambios, confirmando que el costo computacional no es viable para archivos con extensa cantidad de datos.

Merge sort presentó un mejor desempeño al momento de ordenar los datos, ya que se alineó con su complejidad de $O(n \log n)$. Al usar la estrategia de “divide y vencerás”, Merge Sort redujo la carga del procesamiento completando la tarea en 165 ms, registrando solamente 121,911 comparaciones y cero intercambios de manera directa. Al final existió una diferencia notable en el tiempo de ejecución, lo que terminó de validar que $O(n \log n)$ crece mucho más lento que $O(n^2)$.

Búsqueda y selección de rangos

La eficiencia del ordenamiento previo fue un prerequisite indispensable para conseguir optimizar las consultas; una búsqueda secuencial de $O(n)$ hubiera requerido recorrer línea por línea toda la bitácora para lograr encontrar las fechas; el tener los datos ordenados nos permitió implementar la búsqueda binaria.

Aplicar búsqueda binaria fue indispensable para la funcionalidad de buscar por rango y así localizar de manera eficiente el límite inferior (fecha de inicio) y el superior (fecha de fin) dentro del vector para así extraer e imprimir solamente el subconjunto de registros deseados sin tener que realizar iteraciones sobre datos innecesarios.

Reflexión

Como reflexión general los resultados de esta evidencia confirman la teoría que dicta que la elección del algoritmo determina la usabilidad del sistema; para una bitácora la cual se encuentra en constante crecimiento, la combinación de mergesort para el pre procesamiento y la búsqueda binaria para las consultas es la arquitectura óptima que puede garantizar tiempos rápidos de respuesta incluso con grandes cargas de trabajo.

BIBLIOGRAFÍA

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms (3rd ed.). MIT Press.
- Sedgewick, R., & Wayne, K. (2011). Algorithms (4th ed.). Addison-Wesley.
- GeeksforGeeks. (2023). Sorting Algorithms.
<https://www.geeksforgeeks.org/sorting-algorithms/>
- GeeksforGeeks. (2023). Searching Algorithms.
<https://www.geeksforgeeks.org/searching-algorithms/>