“Instituto Tecnológico y estudios superiores de Monterrey”

EVIDENCIA

“Reflexión Actividad 1.3”

Materia:
Programación de estructuras de datos y algoritmos fundamentales
(Gpo 570)

Profesor:
Dr. Eduardo Arturo Rodríguez Tello

Alumno/a:
Ayetza Yunnuen Infante García | A01709011
Luis Carlos Rico Almada | A01252831

09/Julio/2023

Ayetza Yunnuen Infante García | A01709011

El uso de algoritmos de ordenamiento y búsqueda es fundamental en problemáticas en las que es necesario procesar de manera eficiente grandes cantidades de datos. Estos algoritmos nos permiten organizar la información y recuperarla rápidamente cuando sea necesario. Se necesitan algoritmos de ordenamiento para clasificar un conjunto de datos en un orden ascendente o descendente particular. Esto es importante en situaciones en las que se necesita buscar, insertar o eliminar elementos de una lista de manera eficiente.

Los algoritmos de búsqueda, nos permiten encontrar un elemento específico en un conjunto de datos estructurados. La elección del algoritmo de búsqueda también es importante para obtener el máximo rendimiento. Si los datos están ordenados, puede usar la búsqueda binaria, que tiene una complejidad de tiempo de O (log n), donde n es el tamaño de los datos. Esto es mucho más rápido que la búsqueda secuencial que tiene una complejidad de tiempo O(n). Sin embargo, si los datos no están ordenados y solo se realiza una búsqueda, una búsqueda secuencial puede ser más eficiente en términos de tiempo de ejecución. En este caso se implementó el algoritmo de búsqueda binaria para localizar registros específicos dentro de la bitácora.

La elección del algoritmo de ordenamiento adecuado depende del tamaño de los datos y de las restricciones de tiempo y espacio. En esta situación problema, para gestionar y ordenar los datos de la bitácora, se implementaron los algoritmos de ordenamiento: Quick Sort, Bubble Sort; los cuales redujeron en gran medida el tiempo de búsqueda al garantizar que los datos estén preestablecidos.

Al comparar los resultados de la implementación de ambos algoritmos en la situación problema nos llevó a la conclusión de que Quicksort es más eficiente y adecuado para ordenar conjuntos grandes de datos. En los resultados de ambos algoritmos, se ha observado que Quicksort ha demostrado una mayor eficiencia en términos de tiempo de ejecución; esto coincide con el análisis de la complejidad temporal, que indica que Quicksort tiene una complejidad promedio de O(n log n). El Bubble Sort es fácil de entender e implementar, pero su eficiencia no es óptima. Tiene una complejidad de tiempo de O(n^2), donde "n" es el número de elementos en la lista. Esto significa que su desempeño empeora rápidamente a medida que el tamaño de la lista aumenta.

Por lo tanto, la importancia y eficacia de utilizar diferentes algoritmos de ordenamiento y búsqueda radica en su capacidad para procesar grandes cantidades de datos de forma rápida y eficiente. La selección correcta de estos algoritmos puede afectar el rendimiento de una aplicación o sistema en problemáticas donde la velocidad y la eficiencia son factores importantes. Además, es importante considerar las características de los datos y las limitaciones del entorno para elegir el algoritmo más adecuado en cada caso.

Luis Carlos Rico Almada | A01252831

En el ámbito de la ciencia computacional, los algoritmos de ordenamiento y búsqueda juegan un papel fundamental en la resolución de diversos problemas. En el caso específico de una situación problema relacionada con el manejo de una bitácora, estos algoritmos se vuelven especialmente relevantes. A través de la implementación y comparación de los algoritmos de ordenamiento Bubble Sort y Quicksort, así como del algoritmo de búsqueda binaria, hemos explorado su importancia y eficiencia en el contexto de la organización y búsqueda de registros en una bitácora.

La necesidad de ordenar registros en una bitácora radica en la importancia de mantener la información de manera estructurada y fácilmente accesible. Un ordenamiento adecuado facilita la identificación y recuperación de registros específicos, lo que es especialmente valioso en situaciones en las que se requiere analizar un rango de registros en función de una fecha y hora determinadas. Además, el ordenamiento adecuado también permite la detección rápida de patrones y la identificación de tendencias en los registros, lo que puede ser esencial para el análisis y la toma de decisiones.

En nuestra investigación, implementamos dos algoritmos de ordenamiento: Bubble Sort y Quick Sort. El Bubble Sort es un algoritmo sencillo que compara e intercambia elementos adyacentes hasta que todo el arreglo se encuentre ordenado. Por otro lado, el Quick Sort utiliza una estrategia de particionamiento recursivo para dividir el arreglo en subarreglos y ordenarlos. Al comparar los resultados de ambos algoritmos, observamos que Quicksort mostró una mayor eficiencia en términos de tiempo de ejecución. Esto coincide con el análisis de la complejidad temporal, que indica que el Quicksort tiene una complejidad promedio de O(n log n), mientras que el Bubble Sort tiene una complejidad de O(n^2). Estos resultados nos llevan a concluir que Quicksort es más eficiente y apropiado para ordenar grandes conjuntos de datos, como en el caso de una bitácora con numerosos registros.

En cuanto al algoritmo de búsqueda, utilizamos la búsqueda binaria para localizar registros específicos dentro de la bitácora. La búsqueda binaria se destaca por su eficiencia, ya que reduce a la mitad la cantidad de elementos a comparar en cada iteración. Esto se traduce en una complejidad temporal de O(log n), lo que lo convierte en una opción altamente eficiente para buscar registros en una bitácora ordenada. A través de un ejemplo de búsqueda, pudimos comprobar su eficacia y rapidez en la recuperación de información.

En conclusión, la implementación y comparación de los algoritmos de ordenamiento y búsqueda en una situación problema de bitácora nos han permitido apreciar su importancia y eficiencia. El ordenamiento adecuado de registros facilita la organización y el análisis de la información, mientras que la búsqueda eficiente mejora la accesibilidad y recuperación de registros específicos. En este sentido, Quick Sort y la búsqueda binaria se destacan como opciones altamente eficientes y apropiadas para estas tareas en una bitácora. Sin embargo, es importante tener en cuenta que la elección del algoritmo adecuado dependerá del tamaño de los datos y las necesidades específicas del problema.

Referencias

C Con Clase | Ordenamiento (introduccion). (2023). Retrieved 9 July 2023, from https://conclase.net/c/orden

Algoritmos de ordenación en C++. Utilizando genéricos, herencia y polimorfismo. (2023). Retrieved 9 July 2023, from https://dlopezcastellote.dev/blog/algoritmos-ordenacion-genericos-herencia-c++/

Bubble sort. (2020). Retrieved 9 July 2023, from https://www.include-poetry.com/Code/C++/Metodos/Ordenamientos/Bubble-sort/

QuickSort - Data Structure and Algorithm Tutorials - GeeksforGeeks. (2014). Retrieved 9 July 2023, from https://www.geeksforgeeks.org/quick-sort/

"Búsqueda Binaria (Artículo) | Algoritmos | Khan Academy". 2023. *Khan Academy*. https://es.khanacademy.org/computing/computer-science/algorithms/binary-search/a/binary-search#:~:text=La%20b%C3%BAsqueda%20binaria%20es%20un,ubicaciones%20posibles%20a%20solo%20una.

Navarro, Andrea. 2020. "Algoritmos De Ordenamiento - Junco TIC". *Junco TIC*. https://juncotic.com/algoritmos-de-ordenamiento/.

(2023). Retrieved 10 July 2023, from https://www.inf.utfsm.cl/~noell/IWI-131-p1/Tema8b.pdf