

Estructura de Datos y Algoritmos Fundamentales

TC1031 - Gpo 608

Nombre:

Alejandro Barrera Bejarano

Matrícula:

A01254672

Carrera:

ITC

Para ordenar y buscar eficientemente en los registros de logs en este problema, es importante considerar la complejidad computacional de los diferentes algoritmos.

Algoritmos de Ordenamiento:

- Bubble Sort: Compara elementos adyacentes e intercambia sus posiciones si están desordenados. Tiene una complejidad $O(n^2)$ en el peor de los casos. No es eficiente para grandes conjuntos de datos. (GeeksforGeeks, 2022).

- Insertion Sort: Toma elementos de la lista e inserta cada uno en su posición ordenada. Tiene complejidad $O(n^2)$ pero es más eficiente que bubble sort. Útil para pequeños conjuntos de datos. (TutorialsPoint, 2022).

- Quicksort: Selecciona un pivote y ordena elementos menores y mayores que éste de forma recursiva. Su complejidad promedio es $O(n \log n)$. Rápido y eficiente para la mayoría de los casos. (Programiz, 2022).

- Mergesort: Divide el conjunto a la mitad, ordena cada mitad y luego mezcla las partes ordenadas. Tiene complejidad $O(n \log n)$. Muy eficiente para ordenar datos grandes. (JavaTPoint, 2022).

Algoritmos de Búsqueda:

- Búsqueda Lineal: Recorre la lista secuencialmente comparando cada elemento. Complejidad $O(n)$ en promedio. Ineficiente para listas grandes.

- Búsqueda Binaria: En listas ordenadas, compara con punto medio para descartar mitad de búsqueda. Complejidad $O(\log n)$. Altamente eficiente incluso en conjuntos muy grandes.

Dado el volumen potencialmente grande de datos, algoritmos de ordenamiento cuadráticos como bubble sort e insertion sort no son recomendables. Quicksort y mergesort son más eficientes con complejidad $O(n \log n)$. Una vez ordenados los datos, la búsqueda binaria con complejidad $O(\log n)$ nos permite encontrar rangos de fechas/horas rápidamente. La combinación de un algoritmo de ordenamiento $O(n \log n)$ y búsqueda binaria $O(\log n)$ es óptima para la eficiencia en este problema.

Referencias:

Bubble Sort. (2022). GeeksforGeeks. Recuperado de:
<https://www.geeksforgeeks.org/bubble-sort/>

Merge Sort. (2022). JavaTPoint. Recuperado de:
<https://www.javatpoint.com/merge-sort>

Binary Search. (2022). Programiz. Recuperado de:
<https://www.programiz.com/dsa/binary-search>

Insertion Sort. (2022). TutorialsPoint. Recuperado de:
https://www.tutorialspoint.com/data_structures_algorithms/insertion_sort_algorithm.htm