



Actividad Integral de Conceptos Básicos y Algoritmos Fundamentales

Pablo Gutiérrez Costales

Instituto Tecnológico y de Estudios Superiores de
MonterreyCampus GDL

A01275119

Profesor: Dr. Eduardo Arturo Rodríguez Tello

Programación de estructuras de datos y algoritmos fundamentales
(Gpo 12) Viernes 24 de Septiembre del 2021
A01275119@itesm.mx

Reflexión

La decisión de métodos de ordenamiento es realmente importante para la decisión y procesamiento del desempeño del código. La decisión de ocupar el mejor método de ordenamiento fue bastante reñida respecto al método de Quicksort y Merge debido a que los dos poseen un tiempo de complejidad bastante similar siendo los dos algoritmos bastante decentes para la cantidad de datos que se necesitaran analizar, se decidió ocupar el método Quicksort debido a una mayor facilidad de implementación en el código, eso se basa principalmente en la eficacia y rapidez que necesitamos debido a la cantidad de datos analizados, aunque si bien el método depende bastante de la asignación del pivote respecto a la complejidad del algoritmo siendo que en los peores de los casos se puede llegar a una complejidad de $\Theta(n^2)$ (siendo una complejidad muy común en otros algoritmos de búsqueda como bubblesort) consideramos que por la naturaleza de los datos no nos supondrá gran problema en la asignación del pivote adecuado para evitar el peor de los casos, logrando en los mejores de los casos una complejidad de $\Theta(n \log n)$.

Respecto a la propuesta de algoritmo de búsqueda, la decisión fue bastante marcada hacia el método de búsqueda binaria siendo un claro ganador en comparación de sus complejidades computacionales, teniendo este método las siguientes complejidades:

Mejor caso: una complejidad de $\Theta(1)$

Caso promedio: $\Theta(\log(n/2))$

Peor caso: $\Theta(\log n)$,

Es de real importancia tener presente los métodos y algoritmos que sean ocupados y planteados para la solución de una problemática, teniendo presente encontrar el cual se acople principalmente a la mejor solución del problema a resolver, buscando la manera más eficaz para la resolución encontrando la optimización de los procesos para permitir una reducción de tiempo como el costo informático.

Referencias:

- GeeksforGeeks. (2021, 29 abril). Quick Sort vs Merge Sort. <https://www.geeksforgeeks.org/quick-sort-vs-merge-sort/>
- GeeksforGeeks. (2021a, abril 9). QuickSort Tail Call Optimization (Reducing worst case space to $\log n$). <https://www.geeksforgeeks.org/quicksort-tail-call-optimization-reducing-worst-case-space-log-n/>
- GeeksforGeeks. (2021c, septiembre 17). Merge Sort. <https://www.geeksforgeeks.org/merge-sort/>
- GeeksforGeeks. (2021c, agosto 10). QuickSort. <https://www.geeksforgeeks.org/quick-sort/> Análisis del ordenamiento rápido (artículo). (s. f.). Khan Academy. Recuperado 24 de septiembre de 2021, de <https://es.khanacademy.org/computing/computer-science/algorithms/quick-sort/a/analysis-of-quicksort>
- Ats, C. P. (2021, 2 agosto). 95. Programación en C++ | Ordenamientos | Ordenación Rápida (QuickSort) - Codificación [Vídeo]. YouTube. <https://www.youtube.com/watch?v=eSfqwAZBDGE&feature=youtu.be>
- 2270, C. S. D. B. C. U. (2018, 2 enero). Data Structures Tutorial - 14 - How to Read in a Comma Separated .txt File in c++ [Vídeo]. YouTube. https://www.youtube.com/watch?v=_IzYGiuX8QM&feature=youtu.be