Héctor Gutierrez

A01253031

Reflexión Actividad 1.3

La importancia de elegir buenos algoritmos que sean eficientes es para poder optimizar recursos, los algoritmos eficientes consumen menos recursos computacionales, como tiempo de CPU y memoria. Esto es de suma importancia en entornos donde los recursos suelen ser limitados, al utilizar algoritmos eficientes, se maximiza la capacidad de utilizar los recursos disponibles de manera más efectiva. (Gómez, 2022)

En el contexto del problema en el cual ocupamos ordenar y buscar una gran cantidad de datos, para hacerlo de una manera mas eficiente, elegimos los algoritmos que tengan una buena notación computacional y que este acorde al problema.

Por ejemplo, pasando con los algoritmos de ordenamiento sabemos que no ocupamos preocuparnos por el espacio, algo que es clave cuando se trata de elegir algoritmos, ya que por ejemplo el algoritmo de bubble sort tiene una buena notación con respecto al espacio, pero como nuestro enfoque principal es la complejidad con respecto al tiempo, el algoritmo bubble sort es irrelevante, ya que en el peor caso es O(n^2), entonces descartamos bubble sort, insertion sort y selection sort.

Ahora se pondrán a preguntar porque no elimine Quick Sort, ya que este algoritmo en el peor caso también es O(n^2), esto es porque el algoritmo utiliza la técnica del pivote, en el cual si se elige un mal pivote su complejidad en el peor caso es de lo peor, entonces como no tenemos mucha información sobre que pivote podría ser el mejor, igual tomamos la decisión de eliminarlo para evitar esa complejidad en el peor caso de O(n^2).

Pues como saben el algoritmo que nos queda es Merge Sort, este algoritmo es muy bueno porque en todos los casos tiene una complejidad Big O de O (n log n), que utiliza la técnica de divide & conquer, como nos preocupaba mucho el peor

caso este era el mejor, su peor caso es de O(n log n) que le gana a todos los algoritmos de ordenamiento que vimos en clase.

Ahora pasamos a los algoritmos de búsqueda que son dos, primero la búsqueda secuencial es muy buena cuando estamos hablando de datos desordenados, pero en medida en que los datos crecen el tiempo de búsqueda aumenta significativamente y esto nos arroja un tiempo de ejecución lineal en el peor caso de O(n), en contraste con el algoritmo de búsqueda binaria, en su peor caso tiene un tiempo de ejecución constante de O(log n), ya que divide repetidamente la lista en mitades, lo que significa que se reduce a la mitad el espacio de búsqueda en cada iteración, esto es especialmente efectivo cuando hablamos de entradas grandes, donde la búsqueda secuencial podría requerir muchas iteraciones, por lo tanto el elegido fue la búsqueda binaria.

Referencias:

- Gómez, I. (2022, 9 febrero). 💻 *Eficiencia de los algoritmos: ¿Qué es?* https://www.crehana.com. https://www.crehana.com/blog/transformacion-digital/eficiencia-de-los-algoritmos/