

Reflexión Actividad 1.3

Emiliano García Aguirre A00827638

12 de septiembre del 2020

Luis Humberto González Guerra

Reflexión

La búsqueda y el ordenamiento de datos son actividades muy importantes en el trabajo de un programador. Gracias a algoritmos anteriormente diseñados podemos realizar estas funciones de manera mucho más eficiente. A diferencia de códigos más simples que no consideran factores como la memoria utilizada, el tiempo de ejecución o el número de comparaciones, los algoritmos de búsqueda y ordenamiento realizan las mismas tareas, pero de una manera más rápida y efectiva.

Un claro ejemplo de lo anteriormente mencionado es la comparación de los algoritmos de búsqueda, un algoritmo de búsqueda secuencial contra uno de búsqueda binaria. Mientras que una búsqueda secuencial recorre cada posibilidad existente, la binaria evita muchas comparaciones innecesarias y devuelve el mismo resultado. Esto puede variar dependiendo de la complejidad del problema y se limita a que la búsqueda binaria requiere que los datos estén ordenados. Aquí es cuando entran los algoritmos de ordenamiento: burbuja, quick sort, merge, etc. Estos nos permiten organizar a través de diferentes procesos una serie de datos de la manera que se requiera.

La complejidad de los algoritmos es muy importante pues nos dice el número de comparaciones realizadas. Por ejemplo, el algoritmo quick sort, que su complejidad en el peor caso es $O(n^2)$ y en su mejor caso es $O(n \log n)$ es mejor que un bubble sort el cual siempre será $O(n^2)$. La búsqueda binaria siempre tendrá una complejidad de $O(\log n)$ mientras que la búsqueda secuencial es $O(n)$ pero en la mayoría de los casos realiza muchas más comparaciones que una binaria.

Gracias a ambos tipos de algoritmos es posible para nosotros, los programadores, manipular los datos de manera más simple. De no tener estas herramientas sería una tarea muy difícil y tardada organizar cada dato de manera individual, en especial en ejercicios grandes. La clave se encuentra en saber analizar el problema de manera correcta para poder aplicar tanto el mejor algoritmo de búsqueda como de ordenamiento.