

Evidencia 1 - Actividad Integral de Conceptos Básicos y Algoritmos Fundamentales

Reflexión individual

Durante este primer bloque me percaté de dos herramientas muy útiles al lidiar con grandes cantidades de datos que se pueden ordenar, los algoritmos de búsqueda y ordenamiento. Investigando y analizando lo que aprendí puedo ver que los estos algoritmos permiten a un programa reducir la complejidad de un programa para aumentar su eficiencia tanto de tiempo como de memoria (Alumni Network, 2020). Lo cuál tiene implicaciones claras en un contexto laboral, donde la efectividad, eficiencia y desempeño de un programa afecta directamente que tanto valor tiene para posibles clientes.

Definitivamente considero que pude decentemente los algoritmos de ordenamiento que vimos en clase y por qué algunos son relativamente lentos. Por ejemplo, el bubble sort que itera en todas posiciones de la lista, moviendo hacia la “derecha” el valor mayor para no volverlo a checar, lo cuál implica el uso de dos ciclos que le dan una complejidad de $O(n^2)$. El cuál es más lento que el Quick Sort, que también itera en todas las posiciones para ordenar el pivote, pero en lugar de iterar nuevamente en todas las posiciones con otro ciclo, utiliza recursividad, dividiendo casi a la mitad el tamaño de la lista cada vez que se tiene que se utiliza la función y solo ordenando el pivote en cada llamada. Esto le da una complejidad de $O(n \log n)$.

También pude ver como el algoritmo de búsqueda binaria es mucho más rápido que el de búsqueda secuencial, porque divide a la mitad la cantidad de números que tiene que checar en cada iteración en lugar de checarlos todos uno por uno como el selection sort. Lo cuál le da complejidad de $O(\log n)$, la cuál es menor que la búsqueda secuencial $O(n)$.

Independientemente, pude poner en práctica lo que sé de C++ y aprender cómo analizar algoritmos para implementarlos, lo cuál me pareció bastante interesante.

Referencia

Alumni Network. (2020, enero 18). *Sorting algorithms explained*. Freecodecamp.org.
<https://www.freecodecamp.org/news/sorting-algorithms-explained/>