Sofia Moreno Lopez A01028251

Es importante notar que no hay una respuesta definitiva cuando se escoge un algoritmo, aunque esto no es decir que no hay algoritmos probados matemáticamente como los más rápidos posibles como es el ejemplo de mergesort con los algoritmos que usan un vector auxiliar, ya que tienen gastos en diferentes áreas. Por ejemplo , mientras que usualmente se habla de la complejidad del tiempo y de qué algoritmo tenga un BigO más pequeño, no solo se debe de considerar este factor a la hora de usar algoritmos de este tipo. Influyen otras variables como la cantidad de datos que se pretenden manipular, su estructura, su tipo de dato (o datos, en caso de ser más de un tipo), implementación en el lenguaje de programación específico (que puede probar menos o más óptimo), y su uso de la memoria.

Tomemos como ejemplo la situación que se quiere resolver en esta actividad: Se cuenta con una bitácora con una gran cantidad de datos que se busca ordenar. Algoritmos como el ordenamiento burbuja podrían servir como una solución, pero serían altamente ineficientes. Tomaría relativamente demasiado tiempo ordenar una gran cantidad de datos, sin mencionar que tiene una complejidad de tiempo cuadrática en el peor de los casos.

En nuestro código usamos insertion sort, otro de los algoritmos que son poco eficientes, con una complejidad de n al cuadrado, sin embargo usando poca memoria. Además de que fue implementado bogosort, que obviamente no es un algoritmo de ordenamiento viable, pero ciertamente puede demostrar la importancia del uso de algoritmos adecuados.

El uso de algoritmos de ordenamiento abarca casi todas las áreas de uso de datos: desde grandes bases de datos por ejemplo que se usan para ordenar los jugadores con más puntos en un sitio de wpm, las funciones que hacen posible el uso de filtros en excel, bases de datos de pacientes que esperan donaciones de órganos y necesitan ser ordenados por calificación de prioridad, o simplemente búsquedas de fechas en un calendario digital.