Comparación algoritmos de ordenamiento:

Problemas:

Tuve problemas para pensar la estructura del código para manejar los archivos que se iban generando con los números aleatorios y los que se generarían después de ordenarlos.

Decidí hacer todo en un mismo código, el generador de los archivos con números aleatorios y el que los ordena y además guarda las estadísticas.

También se me dificultó el escoger la cantidad de datos para el archivo final, dado que si ponía valores superiores por lo menos a 50 millones, el programa no lograba crear un arreglo de un tamaño tan grande. Por lo tanto, escogí 1 millón que mostraba bastante bien las diferencias entre los diferentes algoritmos.

Por último, tuve problema para hacer que ordenaran el mismo archivo y no que me tocara generar archivos aleatorios diferentes para cada algoritmo de ordenamiento.

Cosas que me gustaron:

Me gustó el manejo de los archivos de texto, cuando ví estructuras de datos hace 4 años, me costó muchísimo y creo que no los aprendí realmente bien, pero siento que esta vez con el desarrollo de este trabajo y las distintas pruebas que realicé aprendí muchas cosas.

Aprendizajes:

Son muchas cosas, como lo mencioné anteriormente el manejo de archivos de texto en C.

También aprendí a manejar la librería time.h, logrando calcular de forma más exacta el tiempo de ejecución de cada algorítmo de ordenamiento.

Por supuesto aprendí mucho más de los algoritmos de ordenamiento, dado los resultados que obtuve. Dentro de los cuales me gustaría destacar, el hecho de que para pequeñas cantidades de datos inserción es más rápido que merge sort, además, cuando el arreglo está ya ordenado, inserción es el que menos comparaciones hace.