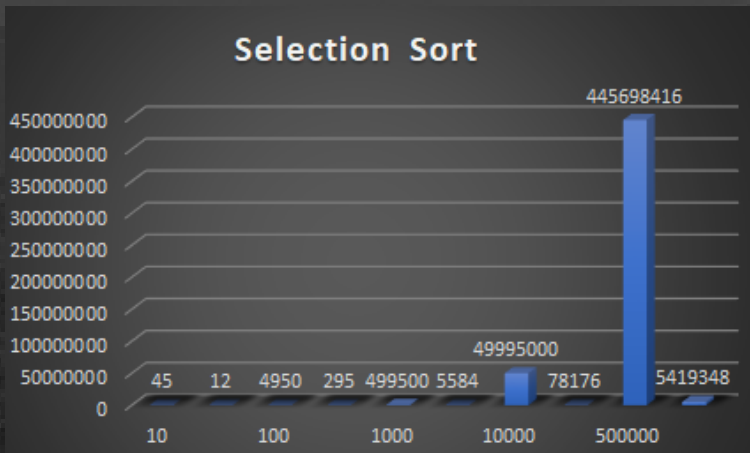


ALGORITMOS DE SORTEO: *ITERATIVOS VS RECURSIVOS*

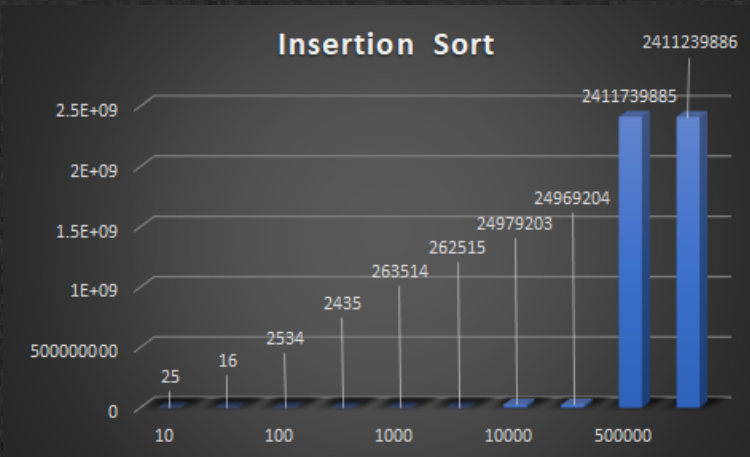
- Se notaron muy pocos cambios en cuanto a tiempos de ejecución, por lo cual se enfocó más en la cantidad de iteraciones y comparaciones para el contraste entre algoritmos
- Las iteraciones para los algoritmos iterativos siempre fueron mayores que para los recursivos (duh, está en el nombre)
- Aunque los algoritmos recursivos hacen menos iteraciones, consumen mucha más memoria, razón por la cual el programa de estos se crasheaba al intentar sortear un conjunto de datos mayor a 250k.
- Por ello se recomienda usar: recursión para sortear menores cantidades de datos, e iteración para conjuntos más grandes (por el consumo de memoria)

SELECTION SORT



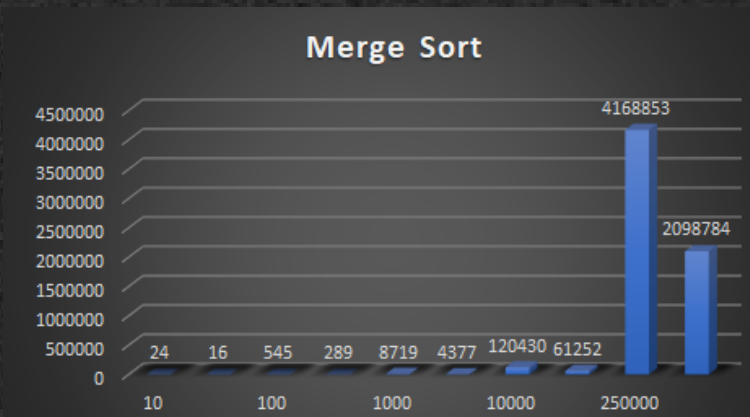
La cantidad de comparaciones es relativamente baja hasta llegar a los últimos dos conjuntos de datos. El runtime para todos los conjuntos de datos salvo el último fue menor a dos segundos. Sin embargo, para 500k datos, el algoritmo tardó 5 minutos y 24 segundos en sortear los elementos

INSERTION SORT



Gracias a la naturaleza de insertion sort, la cantidad de iteraciones crece a una velocidad mucho más rápida que el caso de selection sort. Este algoritmo consume más memoria y a la vez es más lento que su contraparte de iteración en este análisis. Adicionalmente, se observa que las iteraciones y los intercambios son casi que iguales (de nuevo, por la naturaleza del algoritmo). Vale resaltar que este algoritmo es el que más iteraciones hace por una gran margen, llegando a 2 mil millones de iteraciones para 500k datos

MERGE SORT



Aunque las iteraciones para 250k datos son bastantes, al compararlas con aquellos de 500k datos en algoritmos iterativos, no son siquiera datos de la misma orden de magnitud. Esto es porque los algoritmos recursivos requieren de menos iteración, a expensas de memoria.



*Solo hasta cierto punto, pues cuando se agranda mucho el conjunto de datos, los algoritmos recursivos son capaces de consumir toda la memoria de un sistema