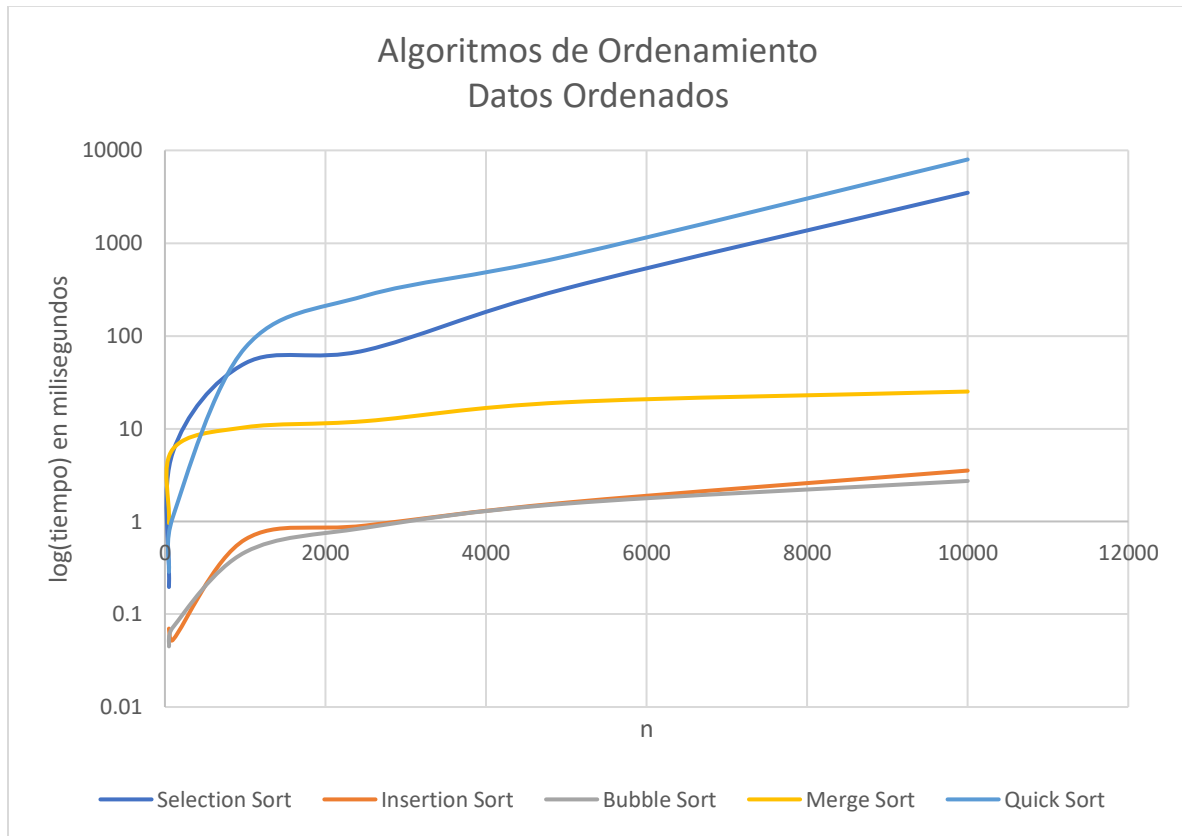


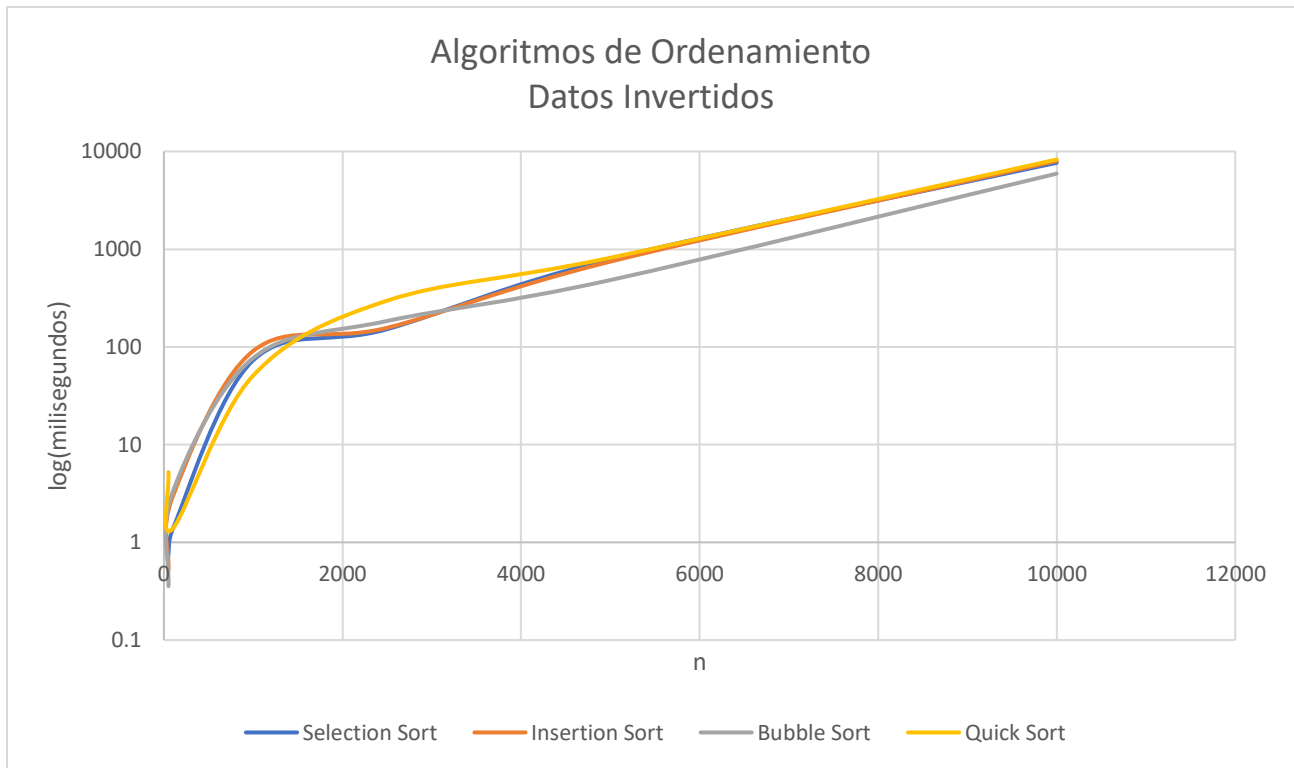
Comparación de los Algoritmos de Ordenamiento**Tarea #1**

Ejercicio 1: Probar los algoritmos de ordenamiento con los datos ordenados.



En el presente gráfico, se puede observar como para los datos ordenados el insertion sort y bubble sort fueron los algoritmos que menos tardaron en ejecutarse. Por otro lado, el selection sort y el quick sort fueron aquellos algoritmos que mas tiempo tardaron en ejecutarse. En medio de estos dos grupos, esta el merge sort, el cual tiene un tiempo promedio respecto al resto de los algoritmos.

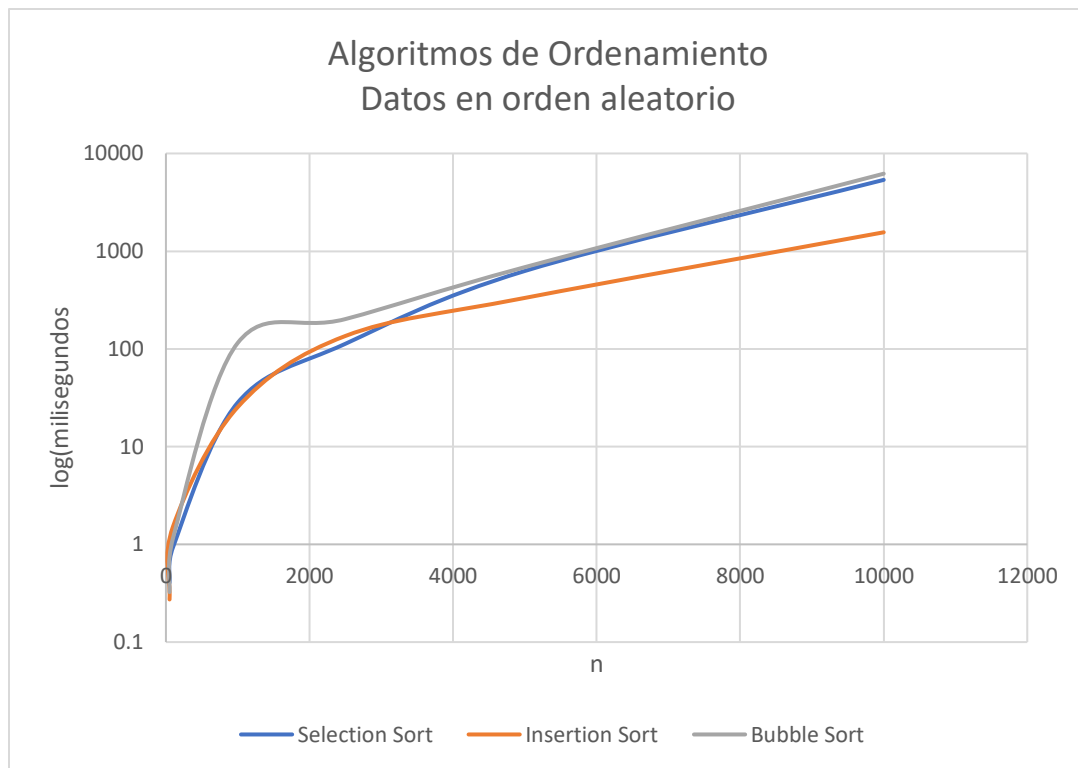
Ejercicio 2: Probar los algoritmos de ordenamiento con los datos invertidos al ordenado del ejercicio 1.



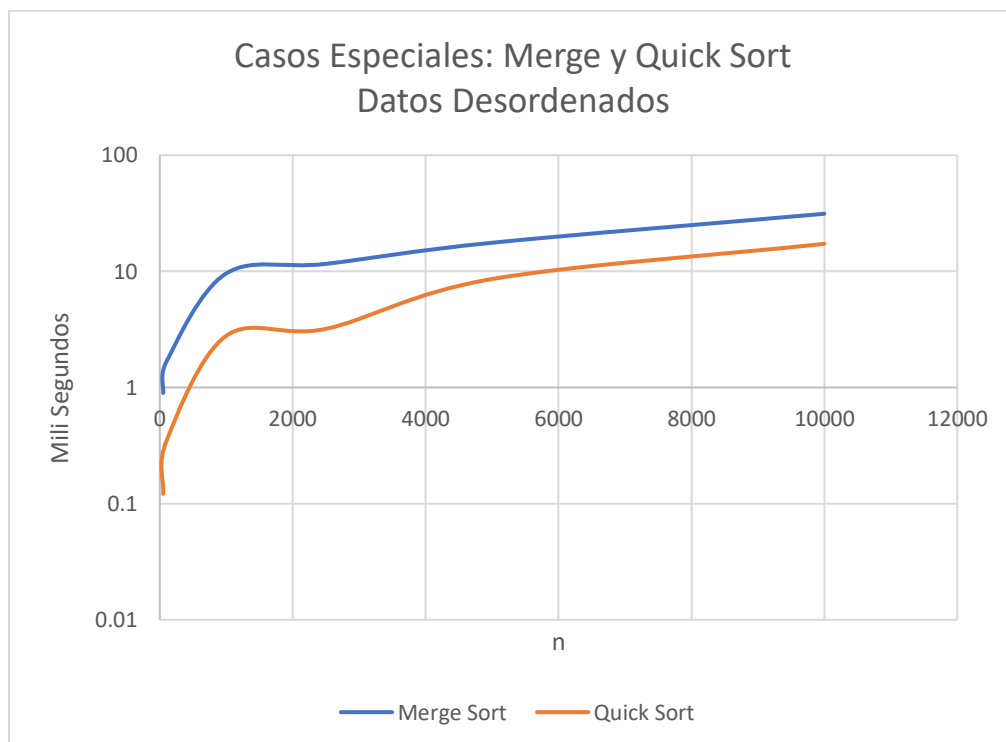
En la presente gráfica, se puede mostrar como cuatro de los 5 algoritmos implementados tienen un comportamiento similar al usar los datos de manera invertida. Se puede notar como el quick sort es por una diferencia mínima el más rápido en ordenar. Le siguen el insertion sort y el selection sort. Con una diferencia mas notable esta el bubble sort, que es el mas lento de estos algoritmos. Sin embargo, no presente en este grafico el merge sort ya que tiene una considerable rapidez de acuerdo con el resto. En la segunda grafico se podrá notar como el merge sort es el más rápido de los algoritmos para los datos ordenados pero invertidos.



Ejercicio 3: Probar los algoritmos de ordenamiento con los datos ordenados de manera aleatoria.



En esta gráfica, se puede ver como los algoritmos bubble sort y selection sort son los que tardan mas de los tres en poder ordenar el arreglo con los datos en desorden. Cercano a ellos pero con una ligera diferencia, está el insertion sort el cual es más rápido pero tampoco alcanza la rapidez del merge sort y quick sort, los cuales son considerablemente mas rápidos. De estos últimos, el quick sort resulta ser el algoritmo más rápido para ordenar arreglos con elementos en desorden.



Finalmente, anexo la grafica en donde se puede ver como van cambiando las comparaciones de acuerdo con la complejidad de cada algoritmo. Se puede ver como el merge y quick sort son los que menos operaciones usan para llegar a ordenar todo el arreglo. Por otro lado, el selection sort, insertion sort y bubble sort tienen un numero mayor de operaciones que realizar para llegar al resultado final.

