

Tarea 1
Algoritmos de ordenamiento

Datos Ordenados

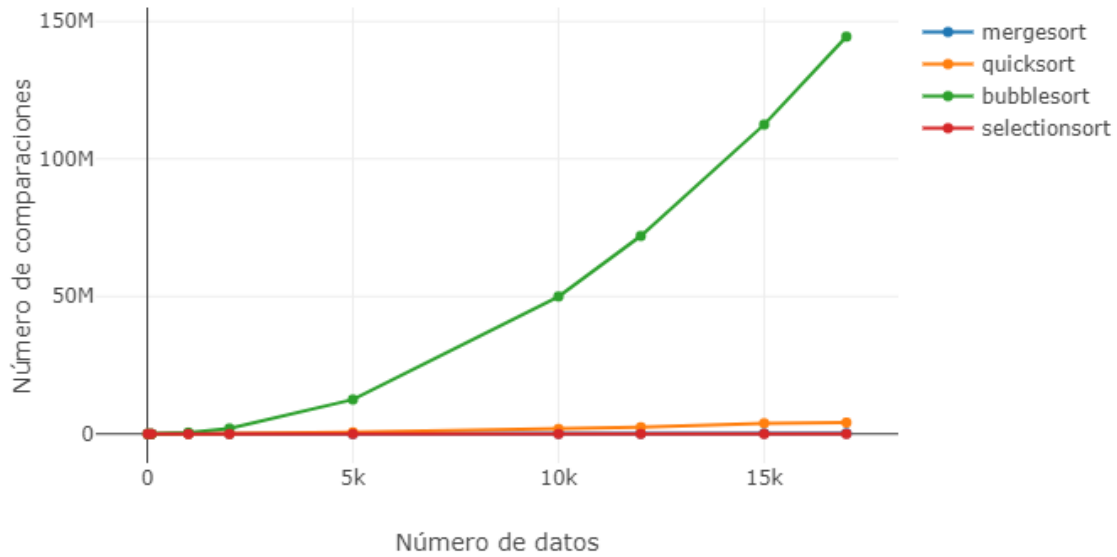


Figura A.

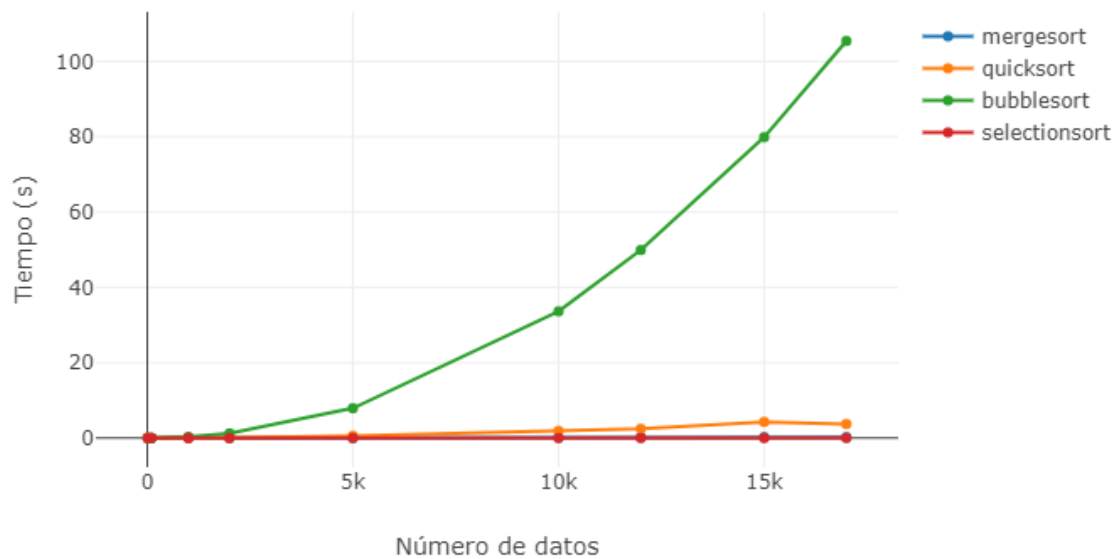


Figura B

Tarea 1
Algoritmos de ordenamiento

Orden Inverso

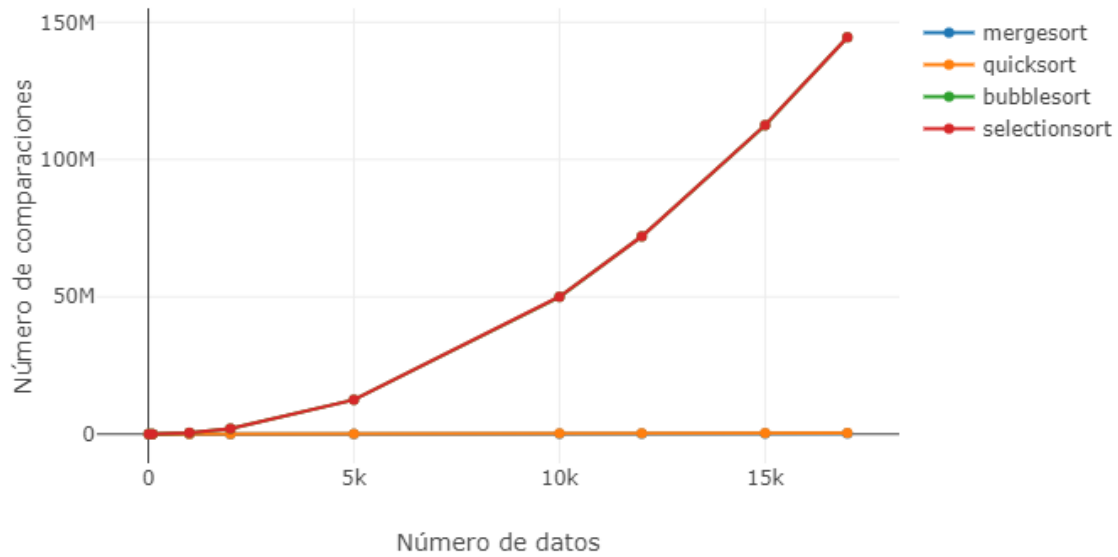


Figura A.

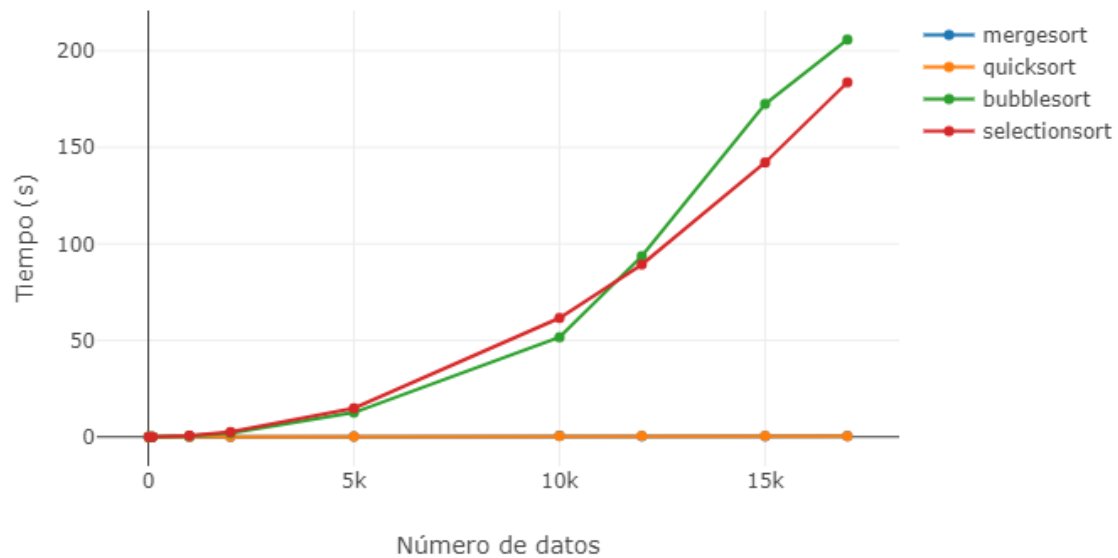


Figura B

Orden Aleatorio

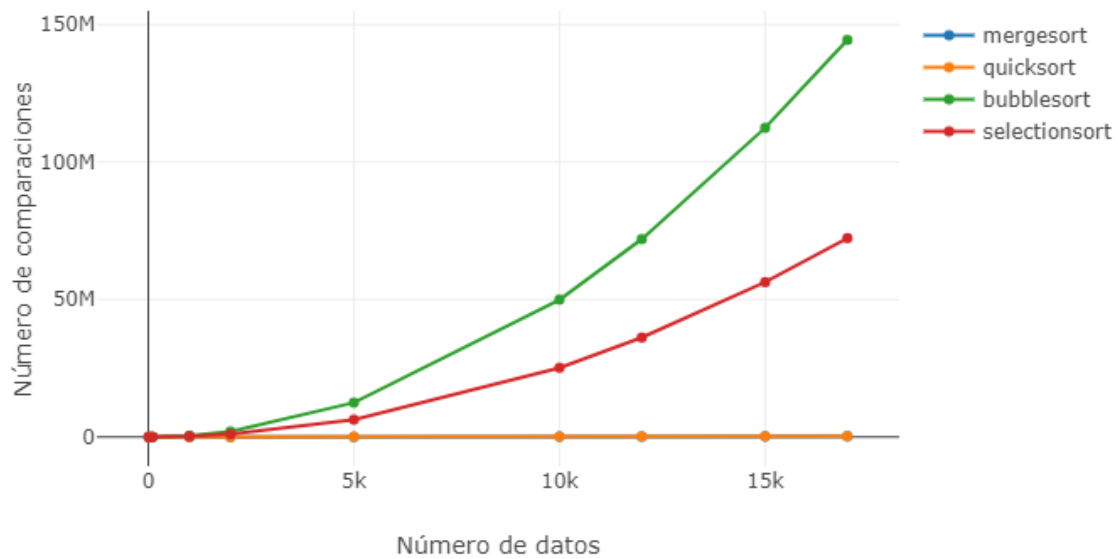


Figura A.

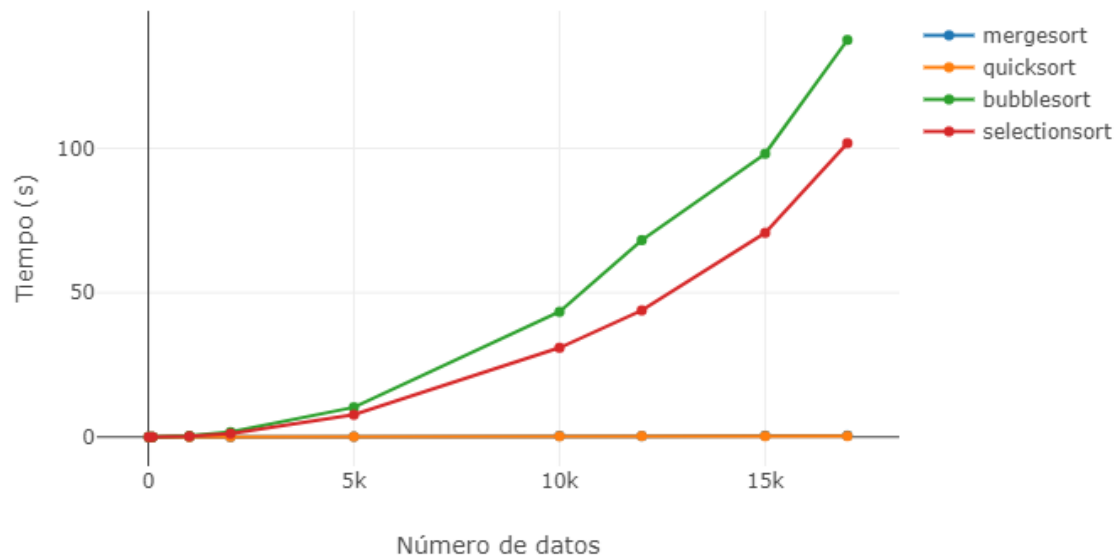


Figura B

Conclusiones

Analizando los resultados arrojados por el Bubble Sort se llega a las siguientes aseveraciones. Primero, el número de comparaciones y el tiempo de ejecución no depende del orden inicial de los datos; además, se puede ver que el número de comparaciones es exactamente igual en los tres incisos. Otra observación es que este algoritmo es el más lento en todos los casos. Por último, se puede inferir su complejidad, que es $O(n^2)$.

Para el Selection Sort se puede hacer las siguientes dos afirmaciones. La primera es que es el algoritmo más rápido y por lo tanto el que hace menos comparaciones si los datos están inicialmente ordenados; para este caso el algoritmo tiene complejidad lineal. La segunda es que, para los otros dos incisos, su gráfica se parece mucho al Bubble Sort. Sin embargo, en promedio (para los datos aleatorios), el selection sort es más eficiente. En conclusión, la complejidad de este algoritmo es $O(n^2)$.

En el Merge Sort se puede inferir por la gráfica que su comportamiento es muy semejante para los tres incisos, por lo que se puede asumir que es el más seguro y de los más eficientes ordenando datos, sin importar el orden inicial de estos. La complejidad de este algoritmo es $O(n \lg n)$.

Por último, el algoritmo Quick Sort resultó competir también por el primer puesto en complejidad. Obtuvo un mejor resultado que el merge sort en los dos últimos incisos. Una buena explicación del por qué pasa esto es que el merge sort consume más tiempo y memoria en su implementación, contrario al quick sort. Es importante decir que el algoritmo implementado es el quick sort que ocupa un pivote aleatorio. Por esta razón, no importa realmente la configuración inicial de los datos. Su complejidad, atestiguado por las gráficas, es $O(n \lg n)$.