Algoritmos de ordenamiento

Córdova Sánchez Ilse Mariana 181901

Estructura de datos avanzada

Instituto Tecnológico Autónomo de México

18 de septiembre de 2019

Introducción

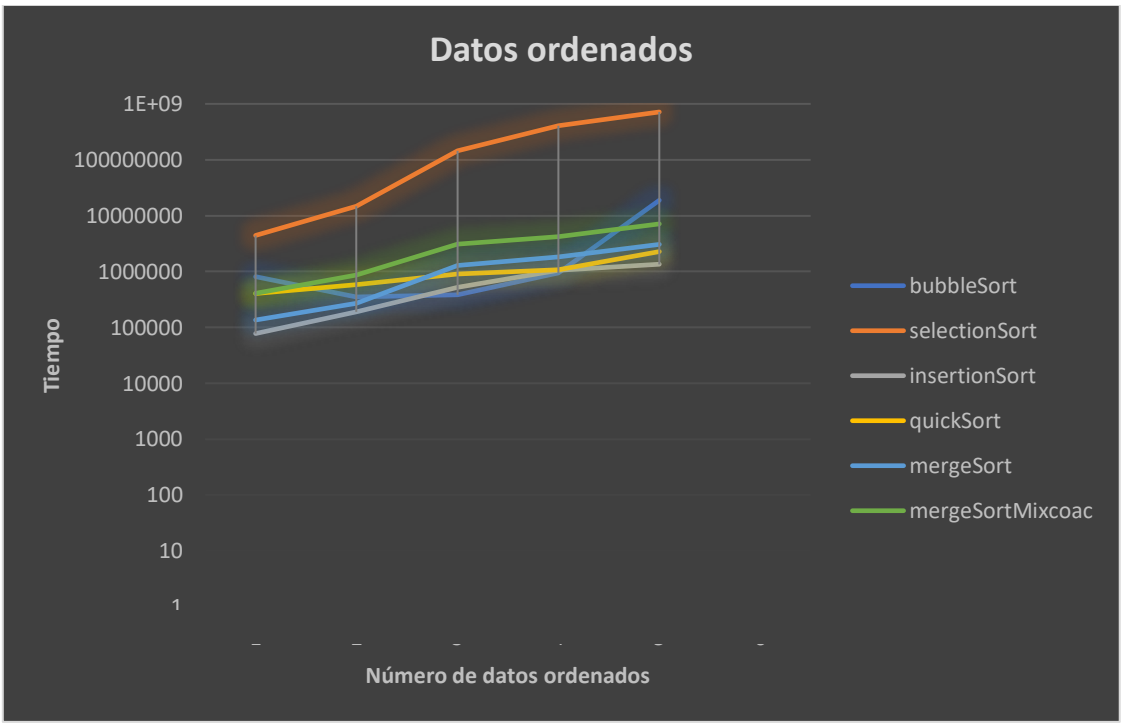
A continuación se presentan seis gráficas correspondientes a las pruebas realizadas con los algoritmos de ordenamiento vistos en clase: bubble sort, selection sort, insertion sort, quick sort, merge sort y merge sort mixcoac.

La cantidad de datos analizada fue de 500, 1000, 1500, 2000, 4000 y 6000. Además, tres de las gráficas corresponden a la relación de tiempo y cantidad de datos ordenados, mientras que las tres gráficas restantes muestran la relación de las comparaciones realizadas y la cantidad de datos ordenados.

Nota: La escala del eje vertical de las gráficas está en forma logarítmica y el tiempo fue medido en nanosegundos.

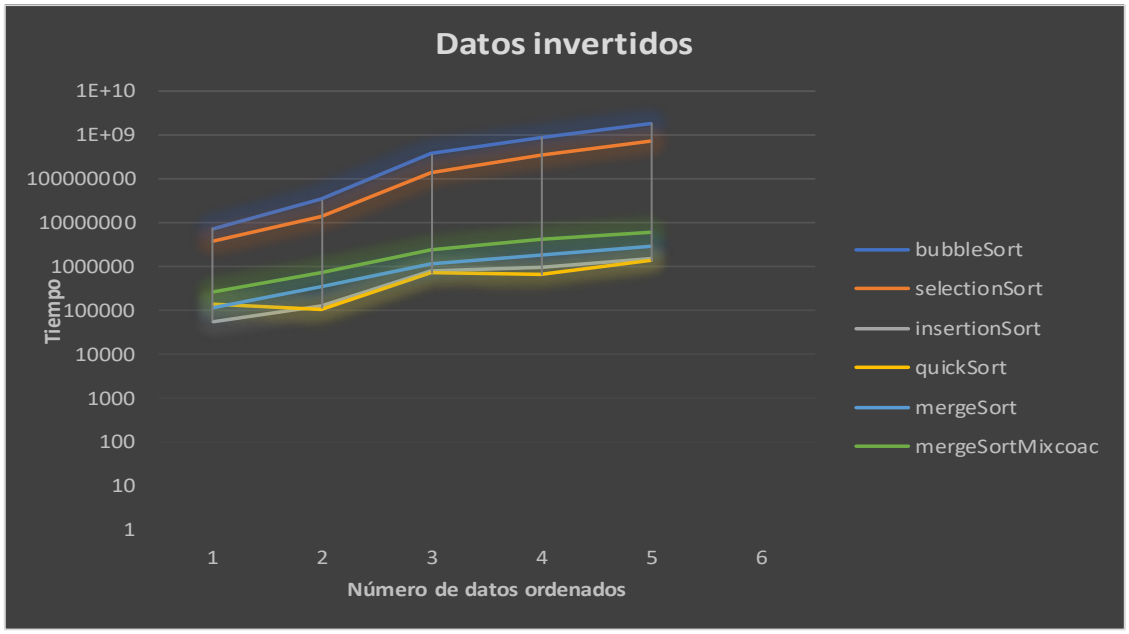
Relación de tiempo y cantidad de datos ordenados

I. Datos ordenados



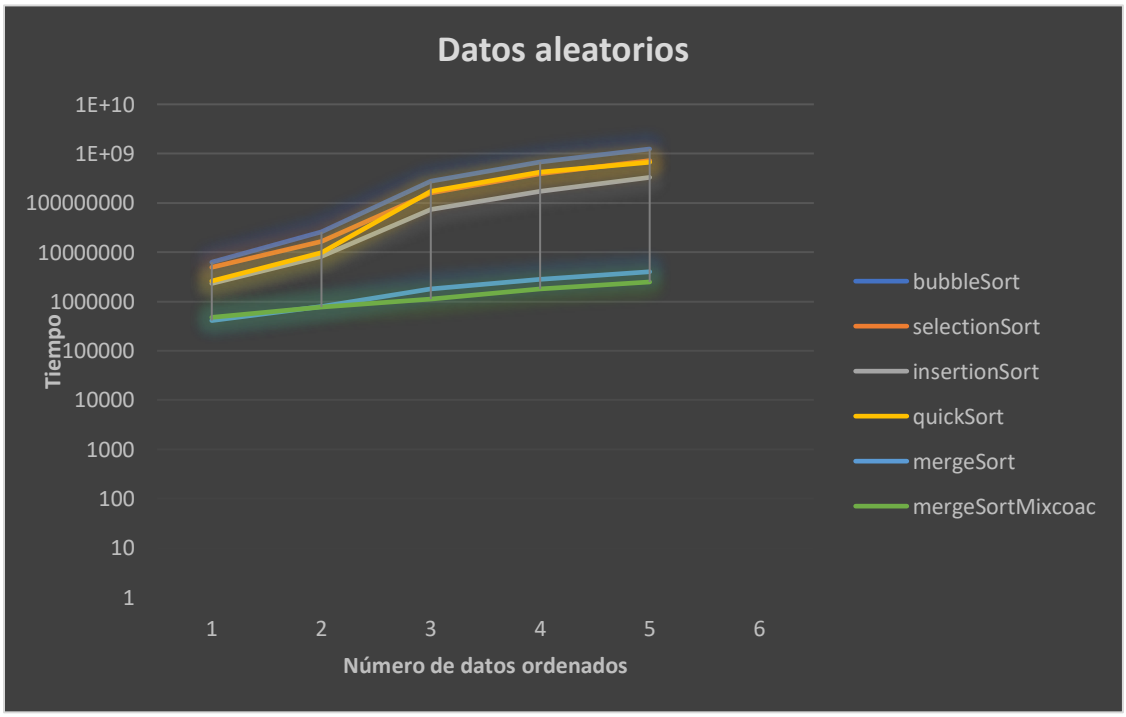
Como se puede apreciar en la siguiente gráfica, los algoritmos que resultaron ser menos eficientes fueron el Bubble sort y el Selection sort mientras que los algoritmos más eficientes son Insertion sort y Merge sort.

II. Datos invertidos



La gráfica muestra que al invertir los datos, el algoritmo que permanece como el más eficiente es el merge sort. En cambio, el algoritmo de Bubble sort y el algoritmo de Selection sort son los que cuentan con menos eficiencia.

III. Datos aleatorios



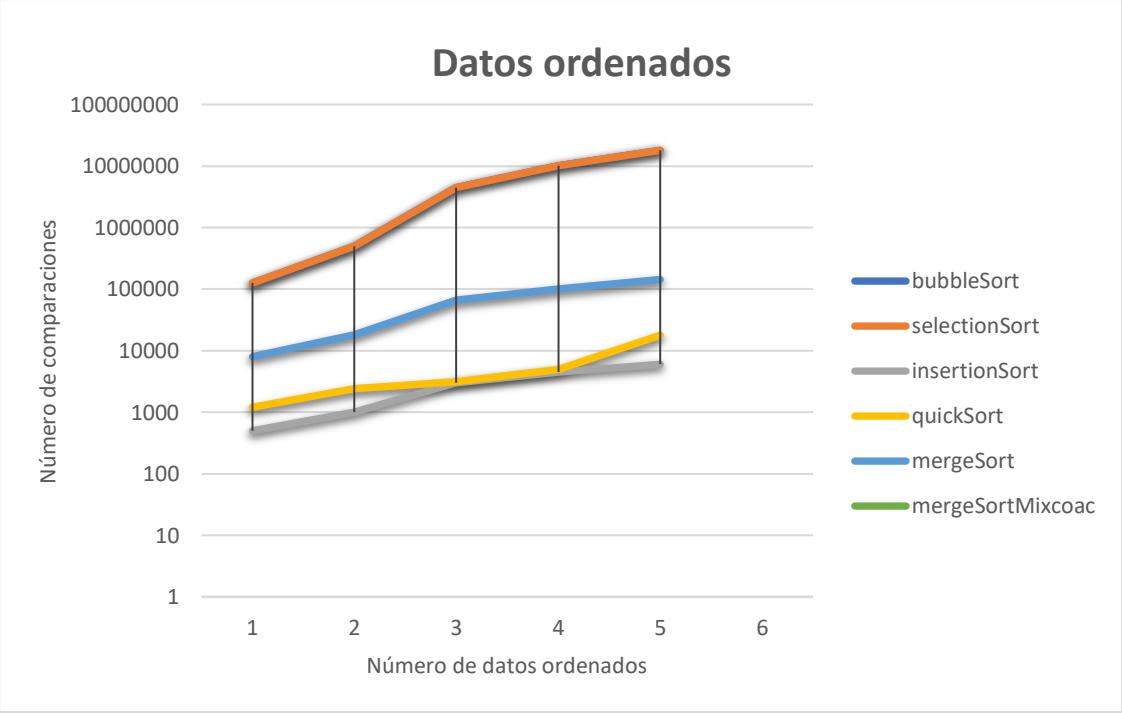
La gráfica presentada demuestra que los algoritmos más eficientes para ordenar datos dispuestos aleatoriamente son tanto el Merge sort como el Quick sort. Por otra parte, los algoritmos menos eficientes son el Insertion sort y Bubble sort.

Conclusiones

El algoritmo Merge sort resulta uno de los más eficientes en los tres casos presentados para el ordenamiento de datos (ordenados, invertidos, aleatorios); además, Quick sort, Selection sort e Insertion sort también resultan eficientes para algunos casos en específico, no para todos como el algoritmo de Merge sort que presenta una complejidad de O (n log n). También se puede observar que otra constante que se presenta es que el algoritmo de Bubble sort de complejidad O (n^2) se encuentra entre los algoritmos menos eficientes para intentar organizar los datos. Con los datos arrojados se puede comprobar que los algoritmos de menor complejidad son aquellos que resultan más eficientes.

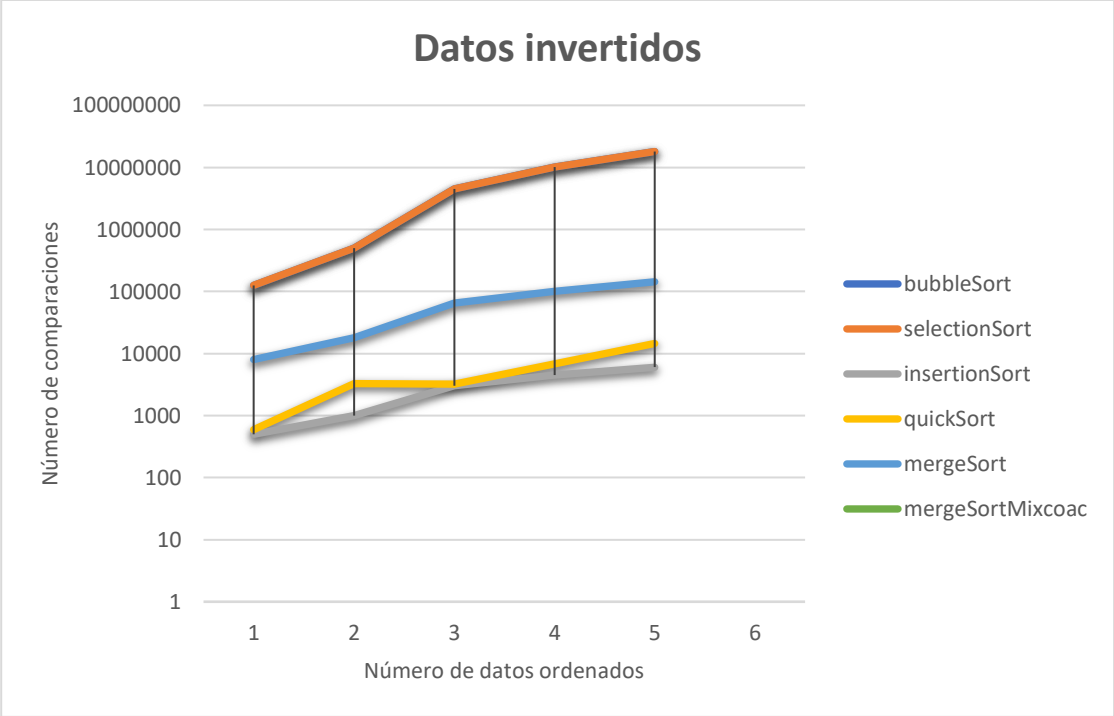
Relación de cantidad de datos ordenados y número de comparaciones

I. Datos ordenados



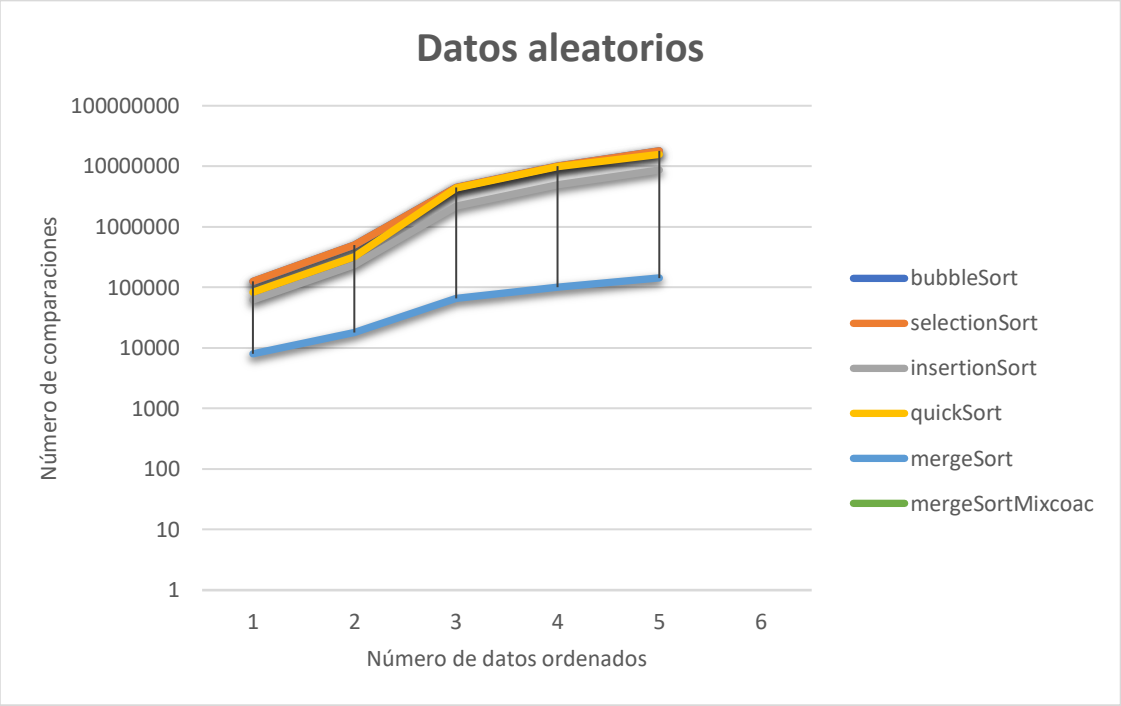
Como puede ser observado en la gráfica, al tomar en cuenta el número de comparaciones y los datos ordenados, tanto Selection sort como Bubble sort son los algoritmos menos convenientes ya que son los más ineficientes. Por otra parte, se tiene que el MergeSortMixcoac y el Insertion sort son los algoritmos que hacen menos comparaciones.

II. Datos invertidos



Al observar la gráfica resulta claro que tanto Bubble sort como Selection sort continúan siendo los algoritmos más ineficientes al realizar el mayor número de comparaciones. En contraste, Insertion sort resulta ser uno de los algoritmos más eficientes junto con el Merge sort Mixcoac.

III. Datos aleatorios



Finalmente, en la última gráfica se puede apreciar a simple vista que Merge sort resulta nuevamente ser el algoritmo que menos comparaciones hace mientras que Quick sort y Selection sort demuestran ser los que mayor número de comparaciones realizan.

Conclusiones

Al igual que en los casos anteriormente presentados, algoritmo Merge sort resulta uno de los más eficientes en los tres casos para el ordenamiento de datos (ordenados, invertidos, aleatorios); además, Quick sort, Selection sort e Insertion sort también resultan eficientes para algunos casos en específico, no para todos como el algoritmo de Merge sort que presenta una complejidad de O (n log n). También se puede observar que otra constante que se presenta es que el algoritmo de Bubble sort de complejidad O (n^2) se encuentra entre los algoritmos menos eficientes para intentar organizar los datos. Una vez más, se puede comprobar que con los datos arrojados se puede comprobar que los algoritmos de menor complejidad son aquellos que resultan más eficientes.