

Universidad del Valle de Guatemala

Algoritmos y Estructura de Datos

Douglas Barrios

Eric Mendoza 15002

Jonnathan Juárez 15377

Javier Jo 14343

Hoja de Trabajo 3

Para la presente hoja de trabajo, se analizaron 4 algoritmos de *Sort*, siendo estos los siguientes:

Gnome Sort

Tiene una complejidad de $O(n)$ en el mejor caso y $O(n^2)$ en el peor caso.

Merge Sort

Tiene una complejidad de $O(n \log(n))$ en el mejor caso y $O(n)$ en el peor caso.

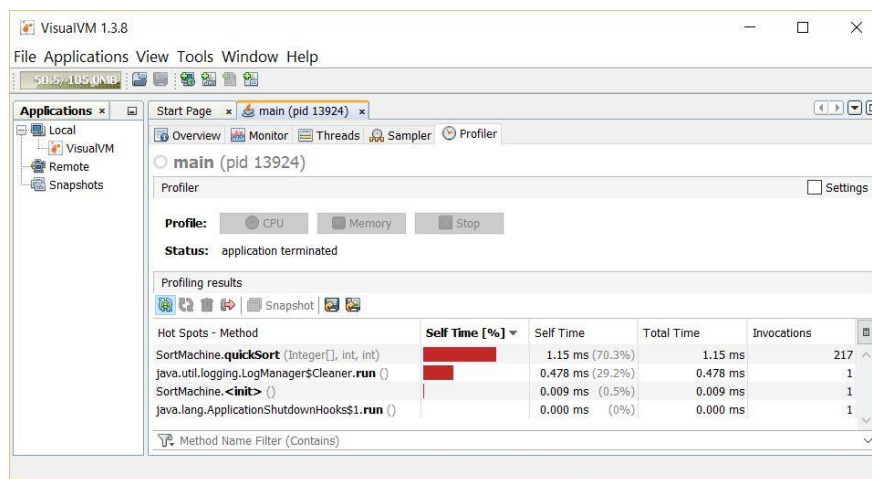
Radix Sort

Tiene una complejidad de $O(n)$ en el mejor caso y $O(n)$ en el peor caso.

Quick Sort

Tiene una complejidad de $O(n \log(n))$ en el mejor caso y $O(n^2)$ en el peor caso.

Profiler

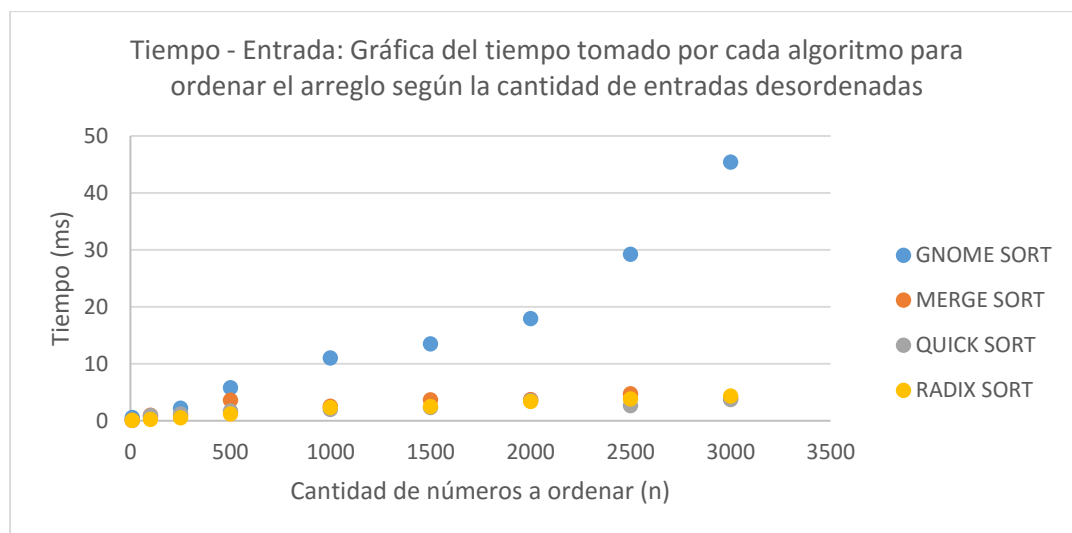


Para el análisis de los diferentes algoritmos se utilizó el profiler VisualVM en su versión 1.3.8 (disponible en: <https://visualvm.java.net/>).

Para la medición de los tiempos de ejecución de cada algoritmo se procedió a correr el programa desde la consola de comandos de Windows. Una vez se estaba ejecutando el programa se abría VisualVM y se seleccionaba el programa. Dentro de las opciones y datos que proveía VisualVM se encontraba la pestaña de profiler, una vez en este apartado se procedía a indicar el método del cual se deseaba obtener los tiempos de ejecución. Finalmente, se daba la instrucción para que el programa corriese y se observaban los tiempos proveídos por el profiler.

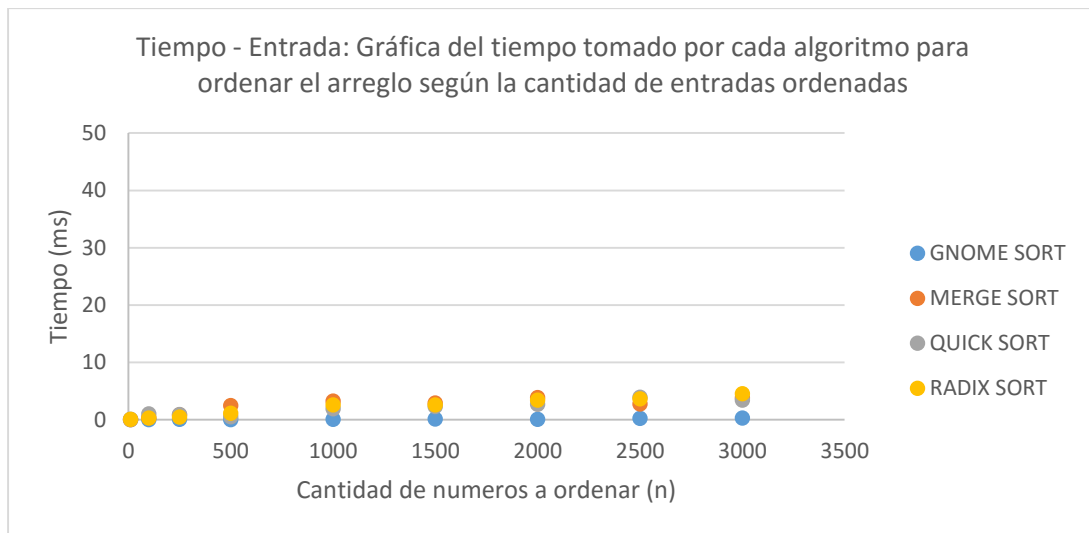
Este procedimiento se realizó para cada uno de los distintos algoritmos y para cada cantidad de datos distinta. Obteniendo los siguientes resultados.

Gráfica No.1: Prueba con listado de números desordenados



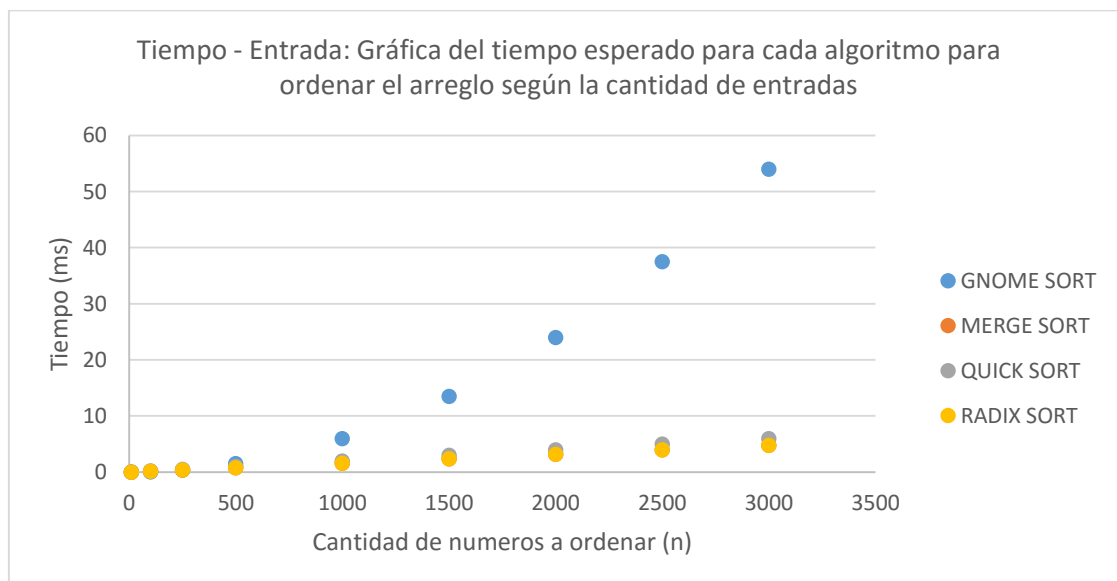
Como se observa en la gráfica, el algoritmo que más tiempo requirió fue el *Gnome Sort*, el cual se sabía que es de orden $O(n^2)$. Por otro lado, *merge* y *quick sort* presentaron un orden semejante al mejor de sus casos $O(n\log(n))$. Y por último, *radix sort* presentó un comportamiento aproximadamente lineal.

Gráfica No.2: Prueba con listado de números ordenados



Al utilizar todos los valores ordenados, el que presentó la mayor diferencia fue el *gnome sort*. Los demás se mantuvieron sus tiempos aproximadamente iguales.

Gráfica No.3: Tiempos estimados para cada uno de los algoritmos basados en su orden.



Como se observa, la estimación para cada uno de los algoritmos fue de acuerdo a los resultados obtenidos. En el caso de los datos ordenados para el *gnome sort*, en el mejor de los casos sí se esperaba ese cambio.