 UNIVERSIDAD SIMÓN BOLÍVAR

LABORATORIO DE ALGORITMOS II

PROF: GUILLERMO PALMA

ALUMNOS: **SABÁS GONZÁLEZ CARNET No: 15-10625**

**MIGUEL PÉREZ CARNET No: 15-11126**

**INFORME PROYECTO 1**

***Introducción***

El objetivo del proyecto es el de hacer un estudio experimental de algoritmos de ordenamientos, simples y de alto rendimiento, sobre diferentes tipos de secuencias. La idea es comparar el rendimiento de los algoritmos de ordenamiento vistos en los libros de texto, junto con otras versiones que son el estado del arte y que son ampliamente usados en la práctica.

**Los Algoritmos a estudiar**

* Mergesort
* Quicksort Iterativo
* Quicksort simple
* Median-of-3 Quicksort
* Introsort
* Quicksort with 3-way partitioning
* Dual pivot Quicksort
* Timsort

**Las Pruebas a realizar**

* Arreglo Punto Flotante
* Arreglo de números enteros ordenados
* Arreglo de números enteros ordenados descendentemente
* Arreglos de ceros y unos
* Arreglo cuyos elementos van de 1 a N/2 y luego de N/2 a 1
* Arreglo casi ordenado donde se intercambian entre 16 pares de elementos al azar
* Arreglo casi ordenado donde se intercambian entre si N/4 pares de elementos al azar

**La Máquina donde se correrán las mismas**

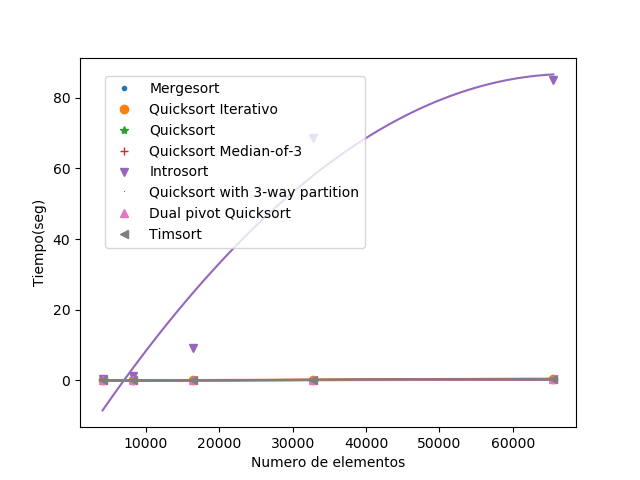
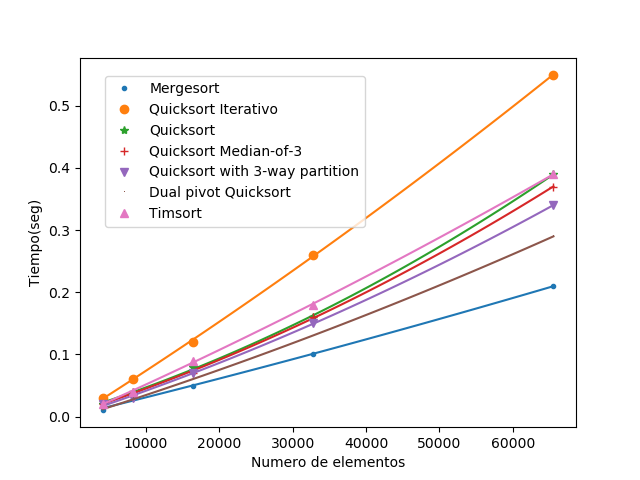
* SO = **Ubuntu 19.10**
* CPU = **Intel i3**
* RAM = **8GB**

***Resultados***

A continuación, se presentan los resultados obtenidos. Cada prueba fue aplicada un número de 3 veces sobre los 8 algoritmos, luego se calculó el promedio de tiempo que tardaron en cada corrida y se creó una gráfica y tabla con estos datos.

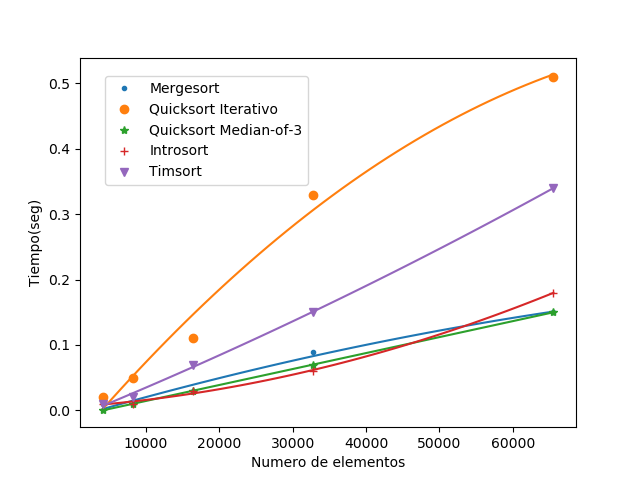
*Figura 1.1 – Tabla 1*

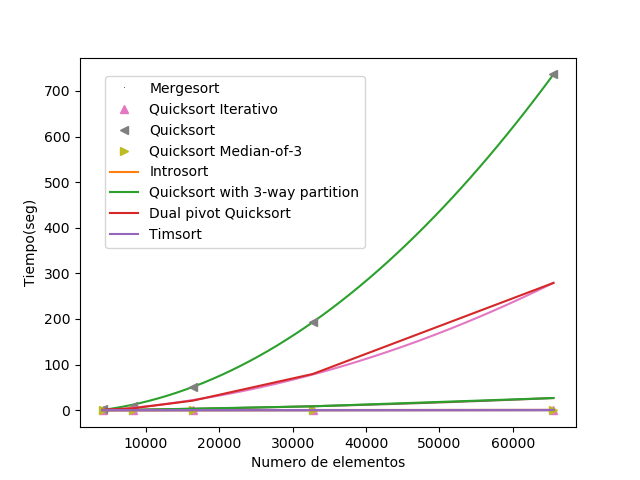


* Figura 1.2 – Grafico Cuadráticas Figura 1.3 – Grafico Lineales*

*Figura 2.1 – Tabla 2*

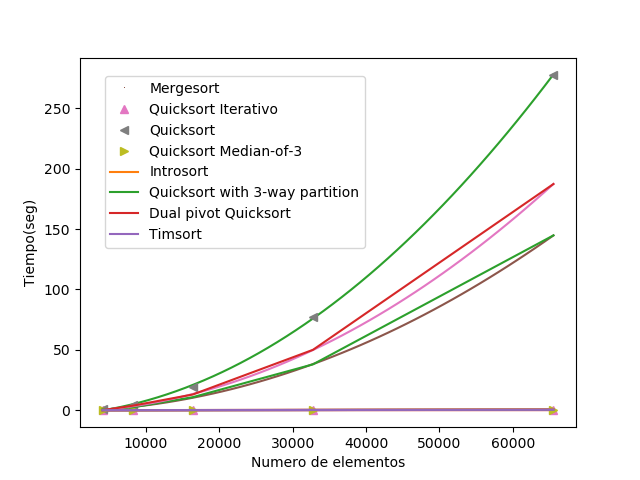


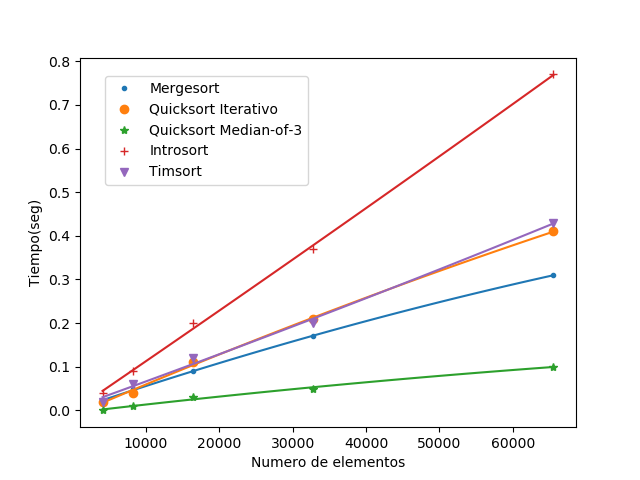
* Figura 2.2 – Grafico Cuadráticas Figura 2.3 – Grafico Lineales*

**

*Figura 3.1 – Tabla 3*

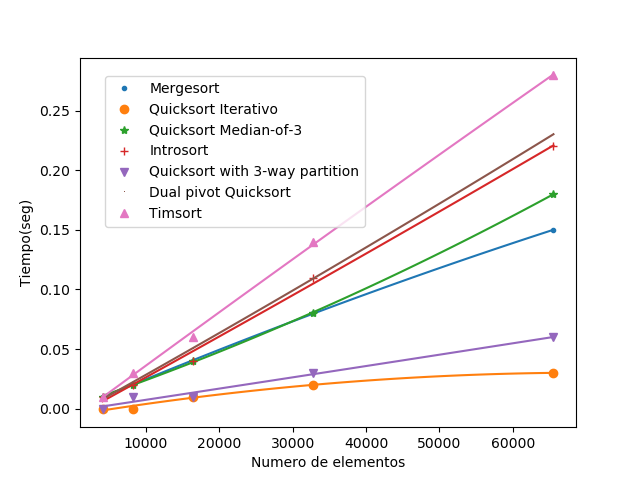


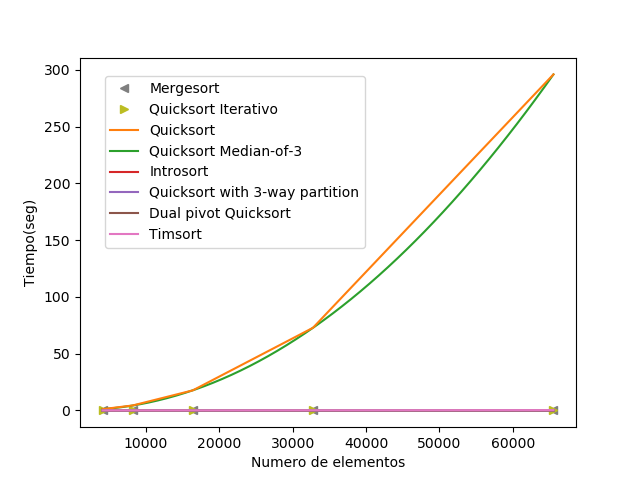
*Figura 3.2 – Grafico Cuadráticas Figura 3.3 – Grafico Lineales*

**

*Figura 4.1 – Tabla 4*

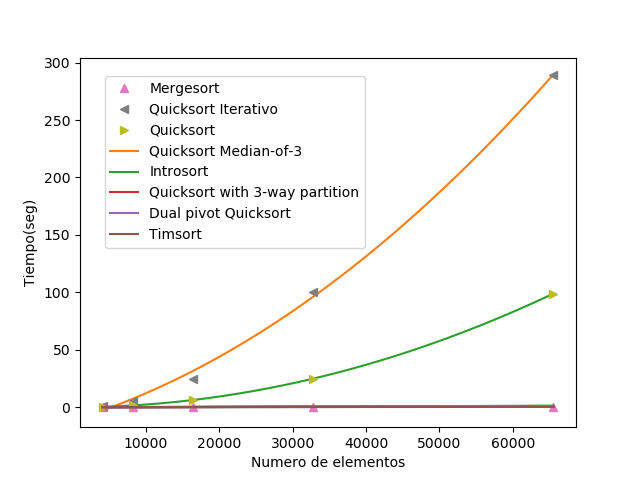
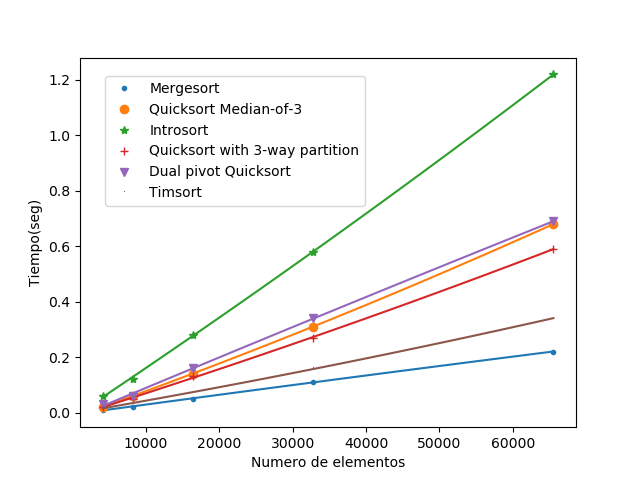


*Figura 4.2 – Grafico Cuadráticas Figura 4.3 – Grafico Lineales*

**

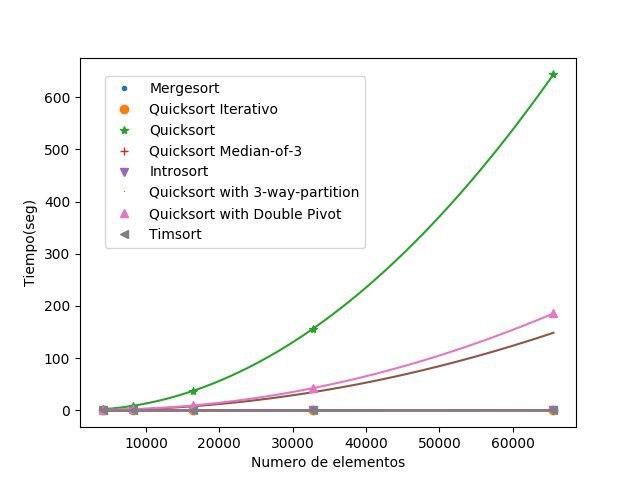
*Figura 5.1 – Tabla 5*

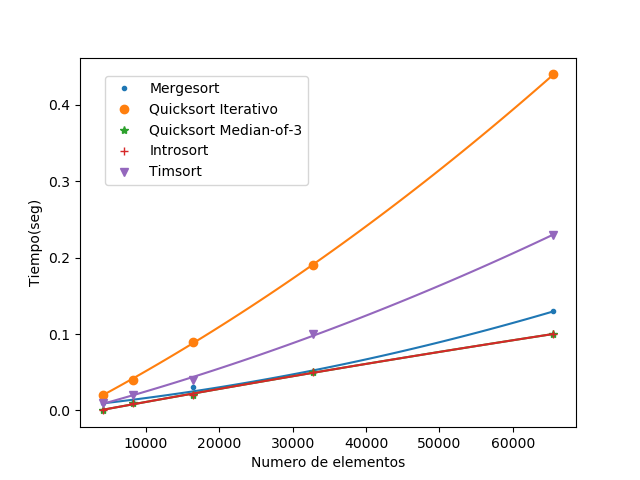


*Figura 5.2 – Grafico Cuadráticas Figura 5.3 – Grafico Lineales*

*Figura 6.1 – Tabla 6*



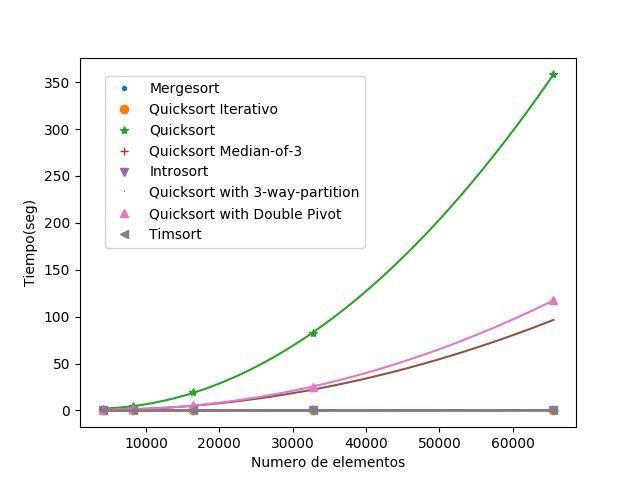
*Figura 6.2 – Grafico Cuadráticas Figura 6.3 – Grafico Lineales*

**

*Figura 7.1 – Tabla 7*



*Figura 7.2 – Grafico Cuadráticas Figura 7.3 – Grafico Lineales*

**

***Análisis de resultados***

* **Prueba 1***:* En esta prueba el algoritmo con el mejor desempeño fue Mergesort y el de peor desempeño fue Introsort. El rendimiento obtenido se corresponde con el esperado para estos algoritmos excepto para el caso del Introsort el cual a pesar de que su peor caso debería ser de tiempo nlog(n), adquiere una curva de crecimiento mucho mayor.
* **Prueba 2***:* En esta prueba el algoritmo con el mejor desempeño fue Mergesort y el de peor desempeño fue Quicksort Simple. El rendimiento obtenido es el esperado para estos algoritmos y tal como su peor caso, el Quicksort simple es el que tarda mucho más debido a que debe atravesar cada elemento del arreglo hasta el final.
* **Prueba 3***:* En esta prueba el algoritmo con el mejor desempeño fue Quicksort Median-of-3 y el de peor desempeño fue Quicksort Simple. Como en el anterior se esperaba que Quicksort Simple sufriera una gran ralentización en su rendimiento y además esta prueba es perfecta para las otras implementaciones de QS las cuales realizan cambios a la forma de pivotear o comparar elementos para evitar los atascos que sufre la versión simple, permitiendo que en este caso Quicksort Median-of-3 sea el más veloz de todos.
* **Prueba 4***:* En esta prueba el algoritmo con el mejor desempeño fue Quicksort Iterativo y el de peor desempeño fue Quicksort Simple. De igual forma que en la prueba anterior, el hecho de que solo haya dos elementos posibles en estos arreglos es un obstáculo inmensio en la vía del Quicksort Simple pero que es sorteado por sus otras implementaciones más avanzadas.
* **Prueba 5***:* En esta prueba el algoritmo con el mejor desempeño fue Mergesort y el de peor desempeño fue Quicksort Iterativo. Tanto Quicksort Simple como iterativo demstraron sus casos de peor tiempo en este tipo de prueba.
* **Prueba 6***:* En esta prueba los algoritmos con mejor desempeño fueron Quicksort Median-of-3 e Introsort y el de peor desempeño fue Quicksort Simple. Salvo por el Quicksort Iterativo y Median-of-3, todas sus implementaciones fallan en esta prueba que a pesar de ser manejada de forma eficiente de manera ordenada en la prueba 2, logra plantar los peores casos de las implementaciones de QS al realizar los intercambios de pares que realiza.
* **Prueba 7***:* En esta prueba los algoritmos con mejor desempeño fueron Quicksort Median-of-3 e Introsort y el de peor desempeño fue Quicksort Simple. Mismo planteamiento y razonamiento que la prueba anterior.