



REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD SIMÓN BOLÍVAR
ENERO – MARZO 2020
LAB ALGORITMOS 2
CI2692

AUTORES

EROS CEDEÑO

USBID: 16-10216

JESÚS BANDEZ

USBID: 17-10046

PROFESOR

GUILLERMO PALMA

INFORME DE RESULTADOS

PROYECTO 1

Introducción:

En cumplimiento de las instrucciones planteadas por el profesor Guillermo Palma, se programaron, implementaron y probaron los algoritmos de ordenación: Mergesort, Quicksort iterativo, Quicksort simple, Median-of-3 quicksort, Introsort, Quicksort with 3-way partitioning, Dual pivot Quicksort y además Timsort.

Los algoritmos fueron comparados sometiendo a secuencias de 4.096, 8.192, 16.384, 32.768 y 65.536 elementos, donde por cada entrada se les realizaron 3 pruebas sobre cada tipo de secuencia. Según las especificaciones dadas en el documento Proyecto1CI2692em20.pdf por el profesor Guillermo Palma. Dichas secuencias fueron generadas aleatoriamente. Los tiempos promedios de ejecución fueron almacenados y graficados para observar el desempeño de los algoritmos.

Descripción de la plataforma usada para correr los experimentos:

Sistema Operativo: Linux Ubuntu 18.04.4 LTS

CPU: Intel Core 2 Duo E 7600 3.06 GHz x64

RAM: 3 GB DDR2 600 MHz

Resultados obtenidos:

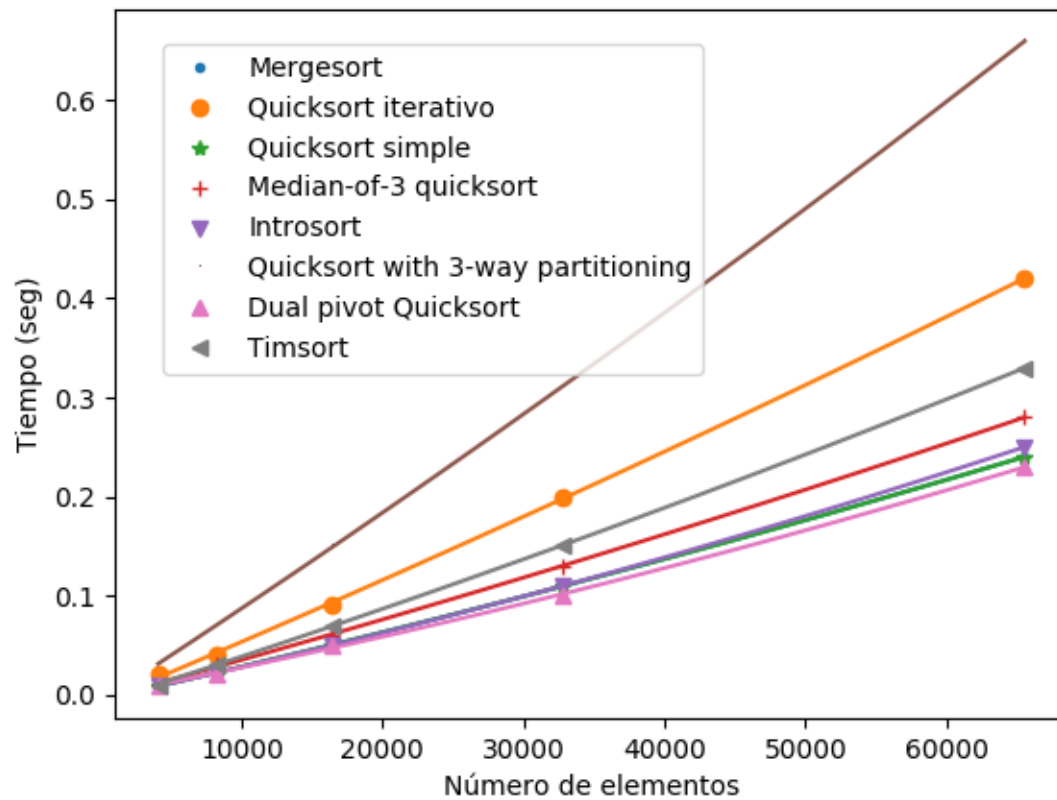
A continuación se presentarán 7 tablas con los tiempos promedio de ejecución de los algoritmos para cada tipo de secuencia.

Descripción de las etiquetas en las tablas:

- **N.º Elem:** Número de Elementos en la secuencia.
- **Mergesort:** Algoritmo Mergesort simple.
- **QS iter:** Algoritmo Quicksort iterativo
- **QS simple:** Algoritmo Quicksort Simple
- **Med-of-3:** Algoritmo Median-of-three quicksort
- **Introsort:** Algoritmo Introsort
- **3-way:** Algoritmo Quicksort with 3-way partitioning
- **DP:** Algoritmo Dual pivot Quicksort
- **Timsort:** Algoritmo Timsort

Secuencias de punto flotante o tipo 1: Números reales comprendidos en el intervalo [0, 1) generados aleatoriamente.

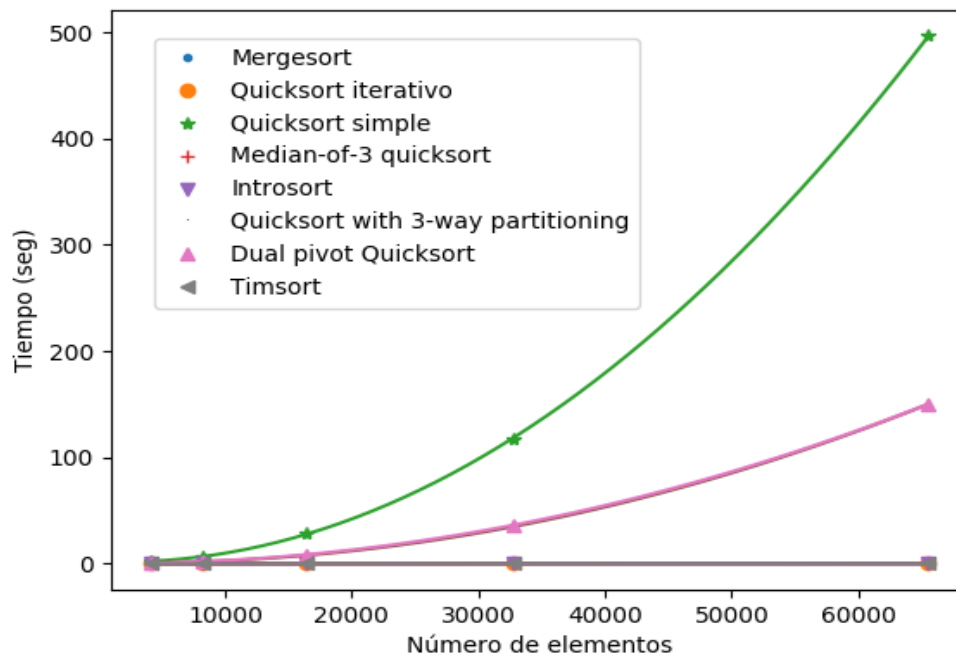
N.º Elem	Mergesort	QS iter	QS simple	Med-of-3	Introsort	3-way	DP	Timsort
4096	0.01 sg	0.02 sg	0.01 sg	0.01 sg	0.01 sg	0.03 sg	0.01 sg	0.01 sg
8192	0.02 sg	0.04 sg	0.02 sg	0.03 sg	0.02 sg	0.07 sg	0.02 sg	0.03 sg
16384	0.05 sg	0.09 sg	0.05 sg	0.06 sg	0.05 sg	0.15 sg	0.05 sg	0.07 sg
32768	0.11 sg	0.2 sg	0.11 sg	0.13 sg	0.11 sg	0.31 sg	0.1 sg	0.15 sg
65536	0.24 sg	0.42 sg	0.24 sg	0.28 sg	0.25 sg	0.66 sg	0.23 sg	0.33 sg



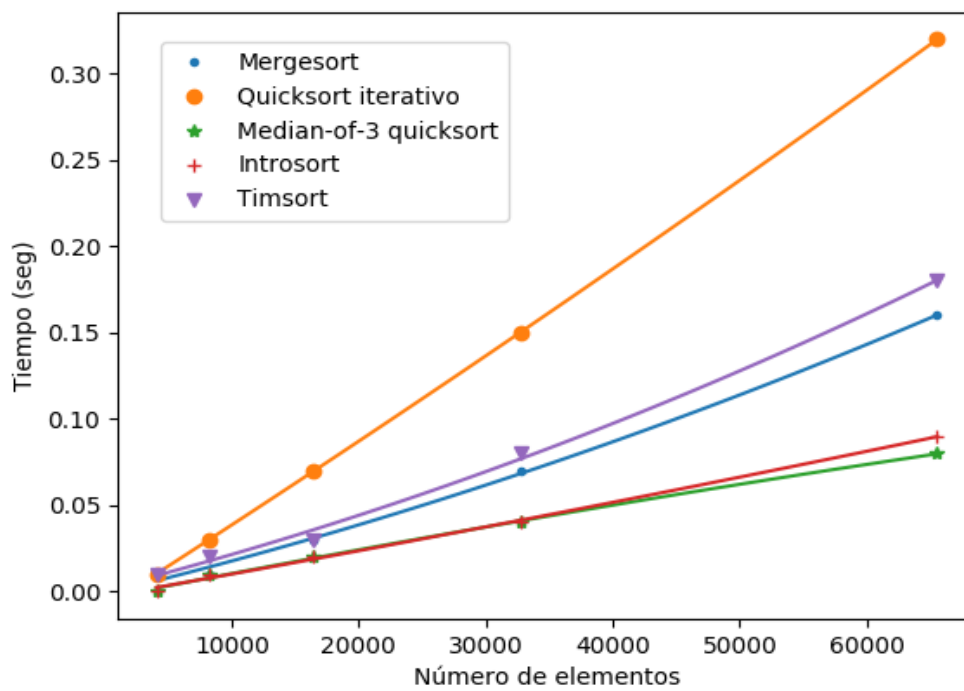
Gráfica Tipo 1

Secuencias Ordenadas o tipo 2: Números enteros ordenados en orden no decreciente.

N.º Elem	Mergesort	QS iter	QS simple	Med-of-3	Introsort	3-way	DP	Timsort
4096	0.01 sg	0.01 sg	1.73 sg	0.0 sg	0.0 sg	0.51 sg	0.52 sg	0.01 sg
8192	0.01 sg	0.03 sg	6.97 sg	0.01 sg	0.01 sg	2.02 sg	2.07 sg	0.02 sg
16384	0.03 sg	0.07 sg	28.27 sg	0.02 sg	0.02 sg	8.21 sg	8.43 sg	0.03 sg
32768	0.07 sg	0.15 sg	117.84 sg	0.04 sg	0.04 sg	34.58 sg	36.34 sg	0.08 sg
65536	0.16 sg	0.32 sg	496.54 sg	0.08 sg	0.09 sg	150.25 sg	150.31 sg	0.18 sg



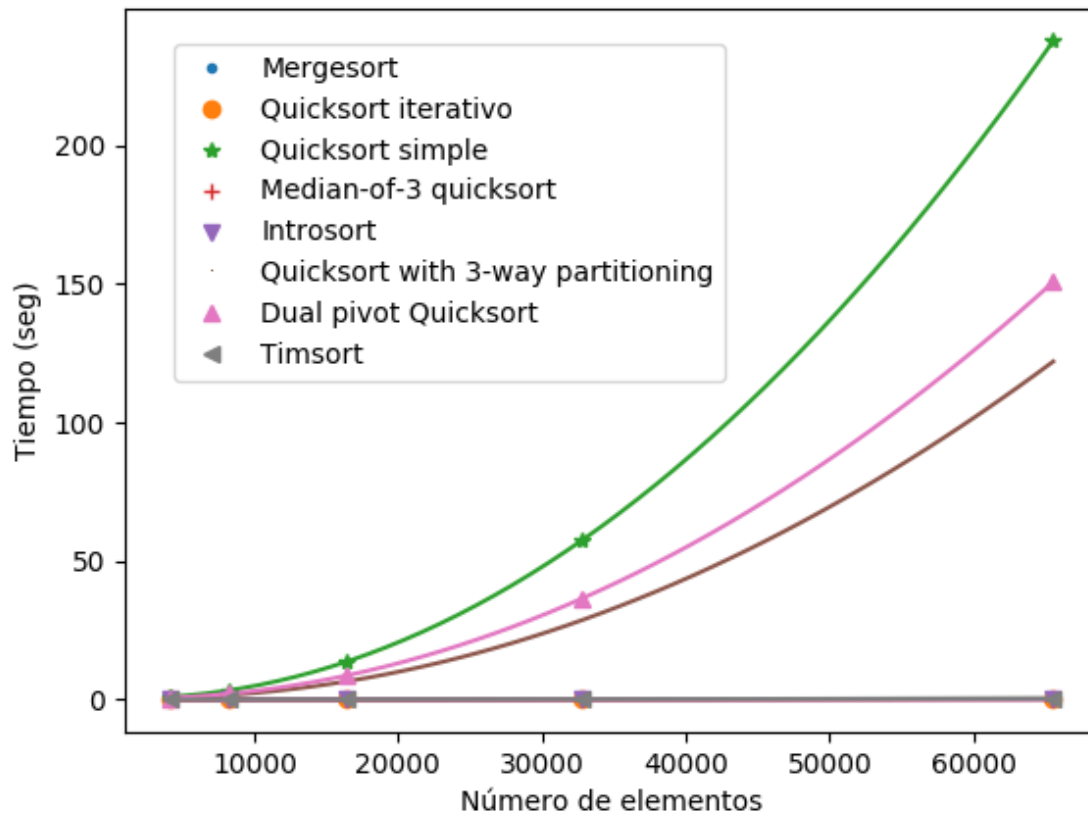
Gráfica Tipo 2



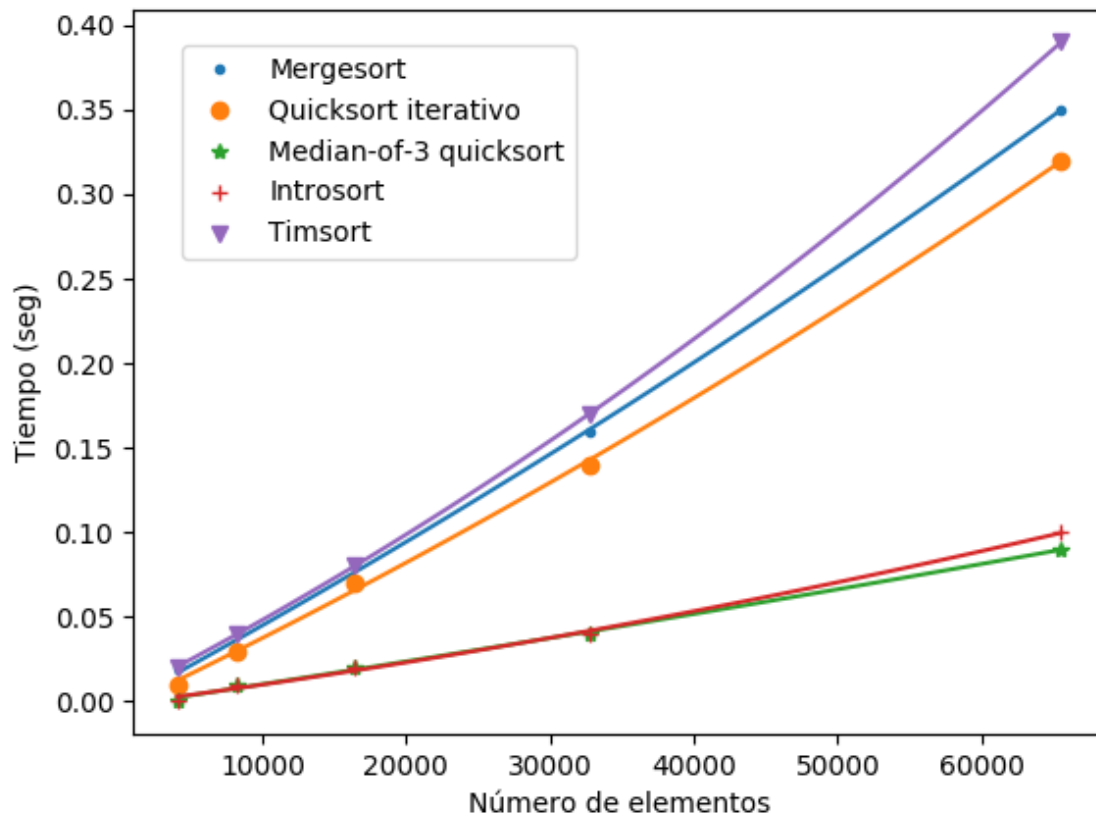
Gráfica Tipo 2 Algoritmos Cuasi-lineales

Secuencias en orden inverso o tipo 3: Números enteros que se encuentran ordenados descendentemente.

N.º Elem	Mergesort	QS iter	QS simple	Med-of-3	Introsort	3-way	DP	Timsort
4096	0.02 sg	0.01 sg	0.86 sg	0.0 sg	0.0 sg	0.42 sg	0.53 sg	0.02 sg
8192	0.03 sg	0.03 sg	3.34 sg	0.01 sg	0.01 sg	1.66 sg	2.11 sg	0.04 sg
16384	0.08 sg	0.07 sg	13.83 sg	0.02 sg	0.02 sg	7.03 sg	8.76 sg	0.08 sg
32768	0.16 sg	0.14 sg	57.22 sg	0.04 sg	0.04 sg	28.27 sg	36.32 sg	0.17 sg
65536	0.35 sg	0.32 sg	237.64 sg	0.09 sg	0.1 sg	122.03 sg	150.68 sg	0.39 sg



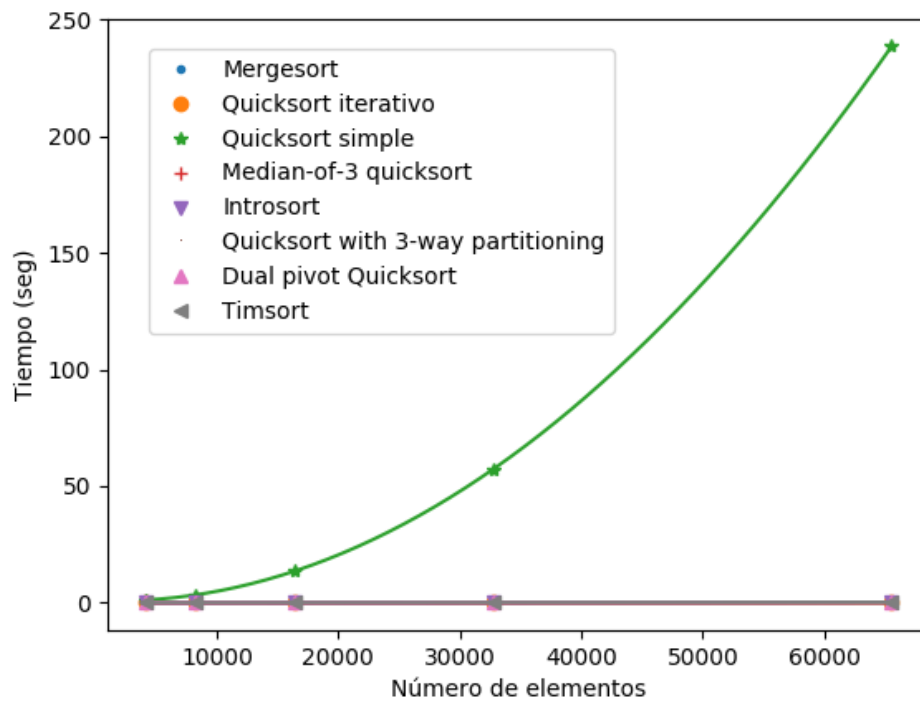
Gráfica Tipo 3



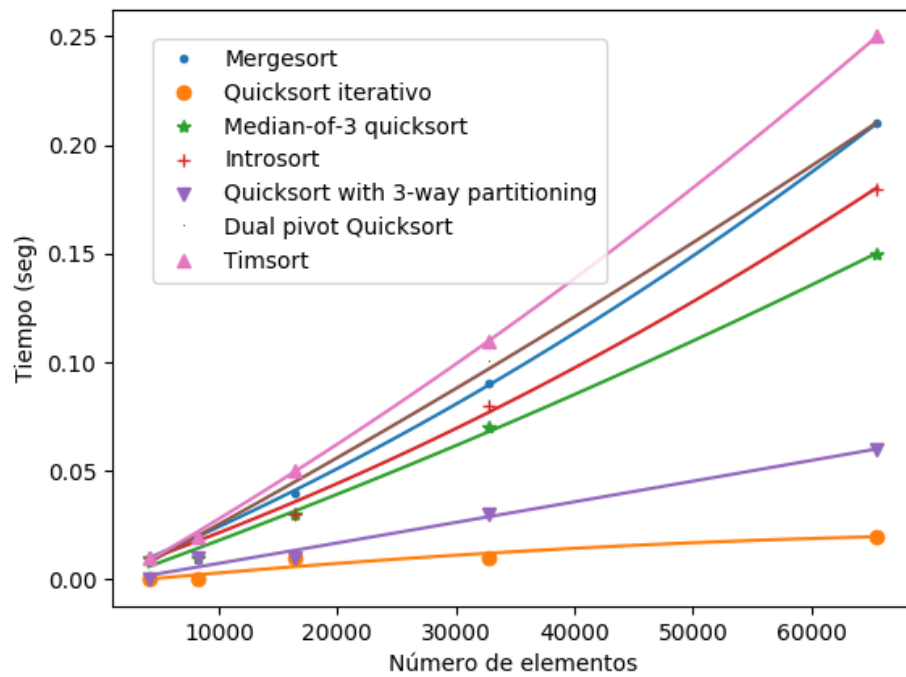
Gráfica Tipo 3 Algoritmos Cuasi-lineales

Secuencias Cero-uno o tipo 4: Ceros y unos generados aleatoriamente.

N.º Elem	Mergesort	QS iter	QS simple	Med-of-3	Introsort	3-way	DP	Timsort
4096	0.01 sg	0.0 sg	0.86 sg	0.01 sg	0.01 sg	0.0 sg	0.01 sg	0.01 sg
8192	0.02 sg	0.0 sg	3.45 sg	0.01 sg	0.02 sg	0.01 sg	0.02 sg	0.02 sg
16384	0.04 sg	0.01 sg	13.83 sg	0.03 sg	0.03 sg	0.01 sg	0.04 sg	0.05 sg
32768	0.09 sg	0.01 sg	57.04 sg	0.07 sg	0.08 sg	0.03 sg	0.1 sg	0.11 sg
65536	0.21 sg	0.02 sg	238.58 sg	0.15 sg	0.18 sg	0.06 sg	0.21 sg	0.25 sg



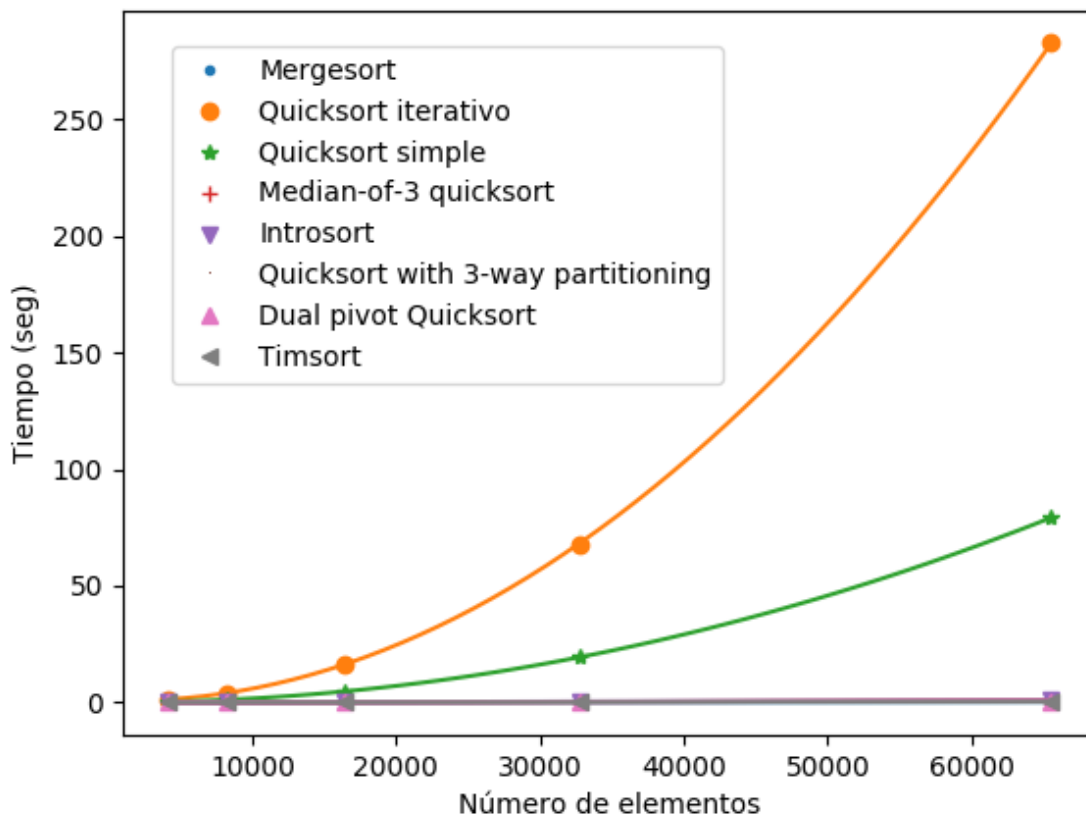
Gráfica Tipo 4



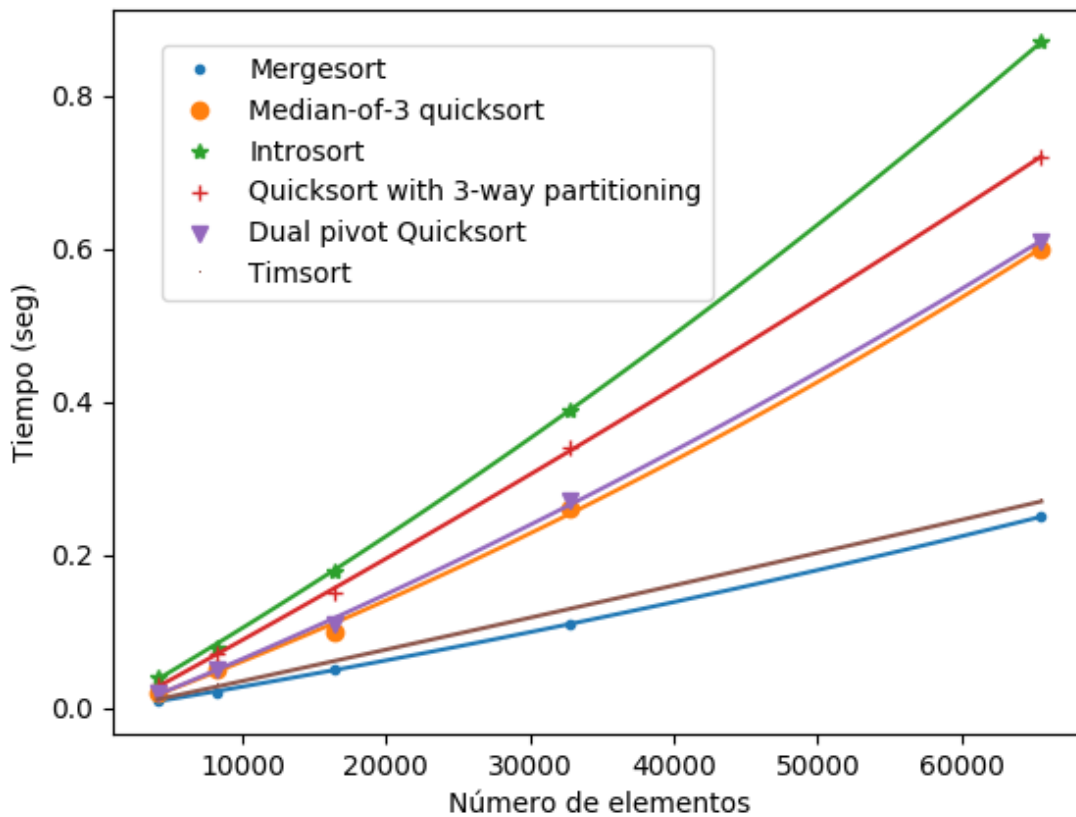
Gráfica Tipo 4 Algoritmos Cuasi-lineales

Secuencias mitad o tipo 5: Dada una secuencia de tamaño N , la secuencia de entrada contiene elementos con la forma $1, 2, \dots, N/2, N/2, \dots, 2, 1$.

N.º Elem	Mergesort	QS iter	QS simple	Med-of-3	Introsort	3-way	DP	Timsort
4096	0.01 sg	1.0 sg	0.3 sg	0.02 sg	0.04 sg	0.03 sg	0.02 sg	0.01 sg
8192	0.02 sg	4.13 sg	1.19 sg	0.05 sg	0.08 sg	0.07 sg	0.05 sg	0.03 sg
16384	0.05 sg	16.6 sg	4.71 sg	0.1 sg	0.18 sg	0.15 sg	0.11 sg	0.06 sg
32768	0.11 sg	68.0 sg	19.4 sg	0.26 sg	0.39 sg	0.34 sg	0.27 sg	0.13 sg
65536	0.25 sg	282.71 sg	79.24 sg	0.6 sg	0.87 sg	0.72 sg	0.61 sg	0.27 sg



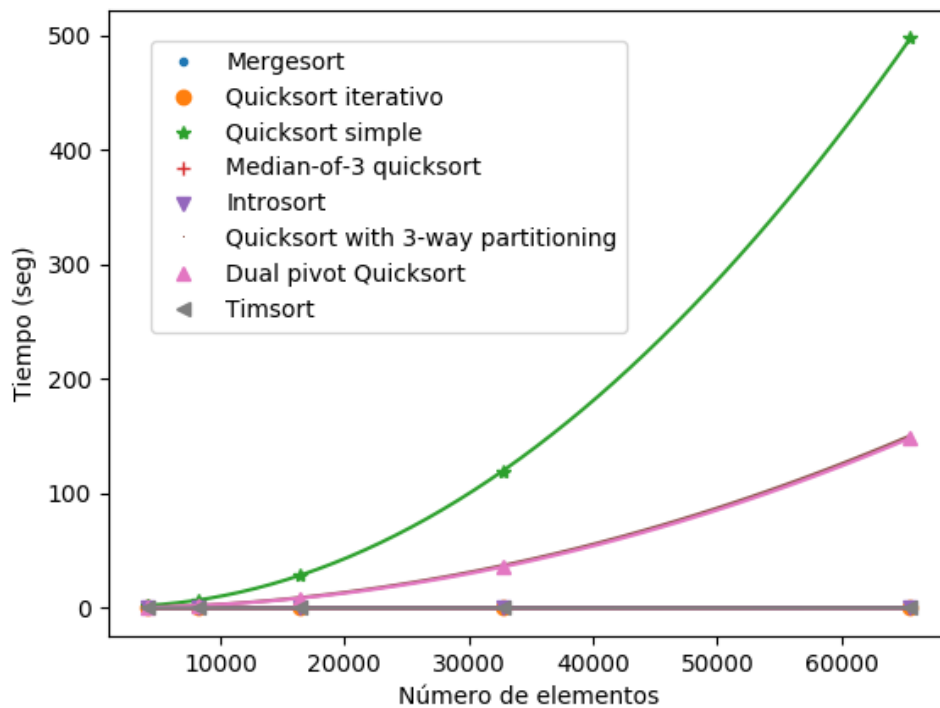
Gráfica Tipo 5



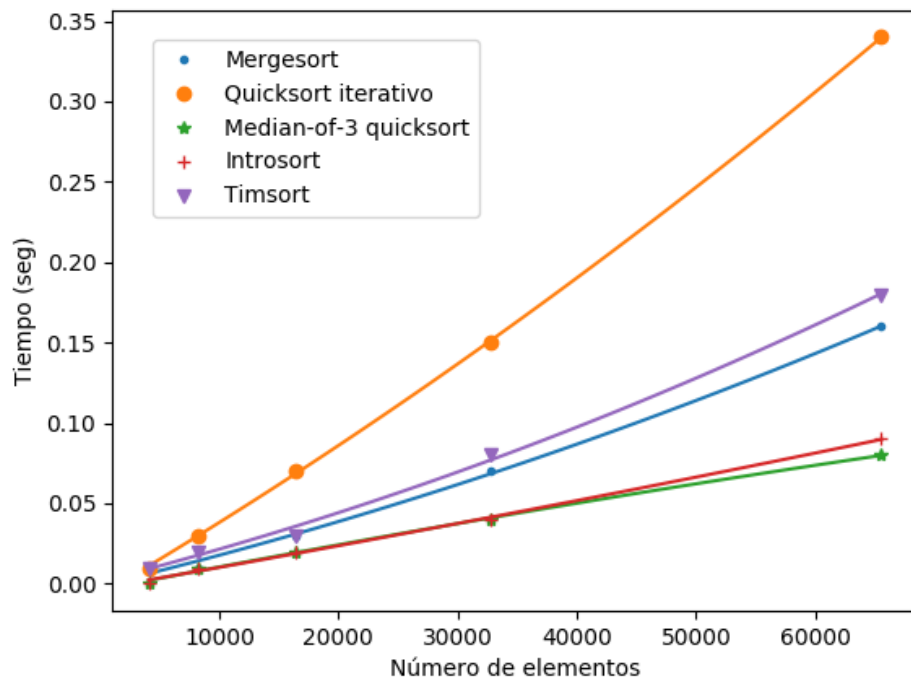
Gráfica Tipo 5 Algoritmos Cuasi-lineales

Secuencias casi-ordenadas 1, o tipo 6: Dado un conjunto ordenado de elementos de tipo entero, se escogen al azar 16 pares de elementos que se encuentran separados 8 lugares, entonces se intercambian los pares.

N.º Elem	Mergesort	QS iter	QS simple	Med-of-3	Introsort	3-way	DP	Timsort
4096	0.01 sg	0.01 sg	1.7 sg	0.0 sg	0.0 sg	0.5 sg	0.51 sg	0.01 sg
8192	0.01 sg	0.03 sg	6.96 sg	0.01 sg	0.01 sg	2.02 sg	2.06 sg	0.02 sg
16384	0.03 sg	0.07 sg	29.13 sg	0.02 sg	0.02 sg	8.43 sg	8.56 sg	0.03 sg
32768	0.07 sg	0.15 sg	119.36 sg	0.04 sg	0.04 sg	37.43 sg	35.9 sg	0.08 sg
65536	0.16 sg	0.34 sg	497.66 sg	0.08 sg	0.09 sg	150.31 sg	148.5 sg	0.18 sg



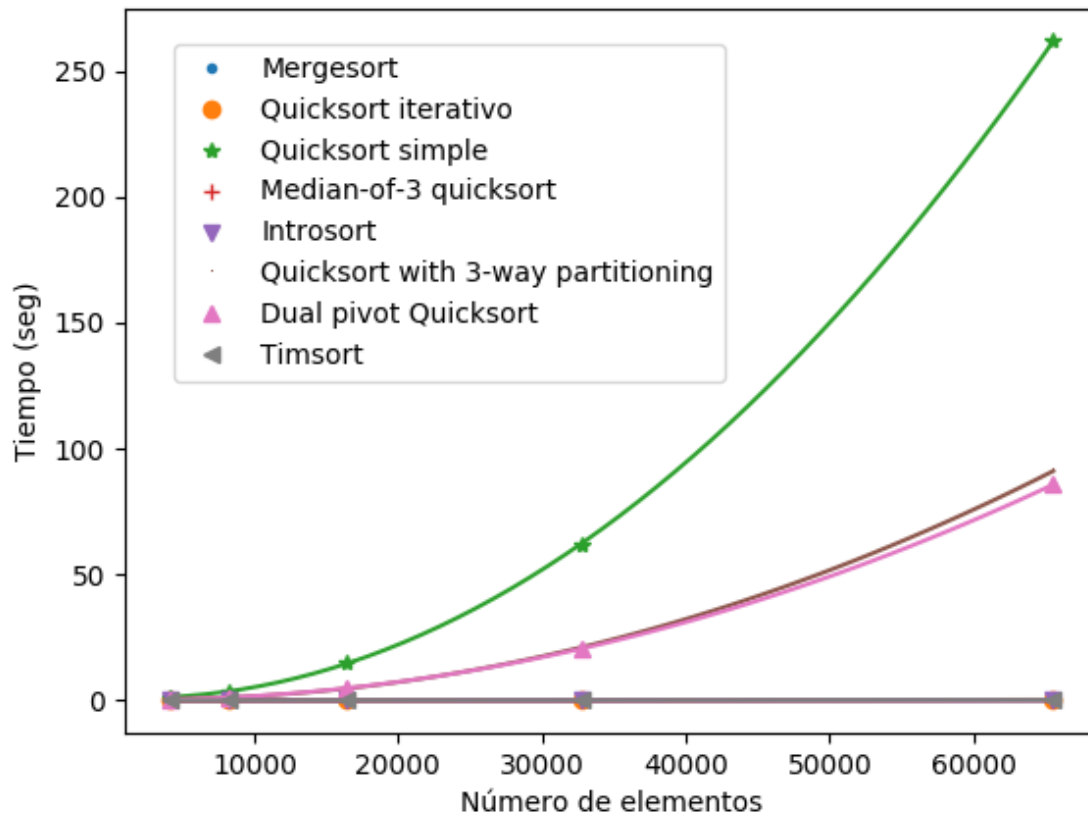
Gráfica Tipo 6



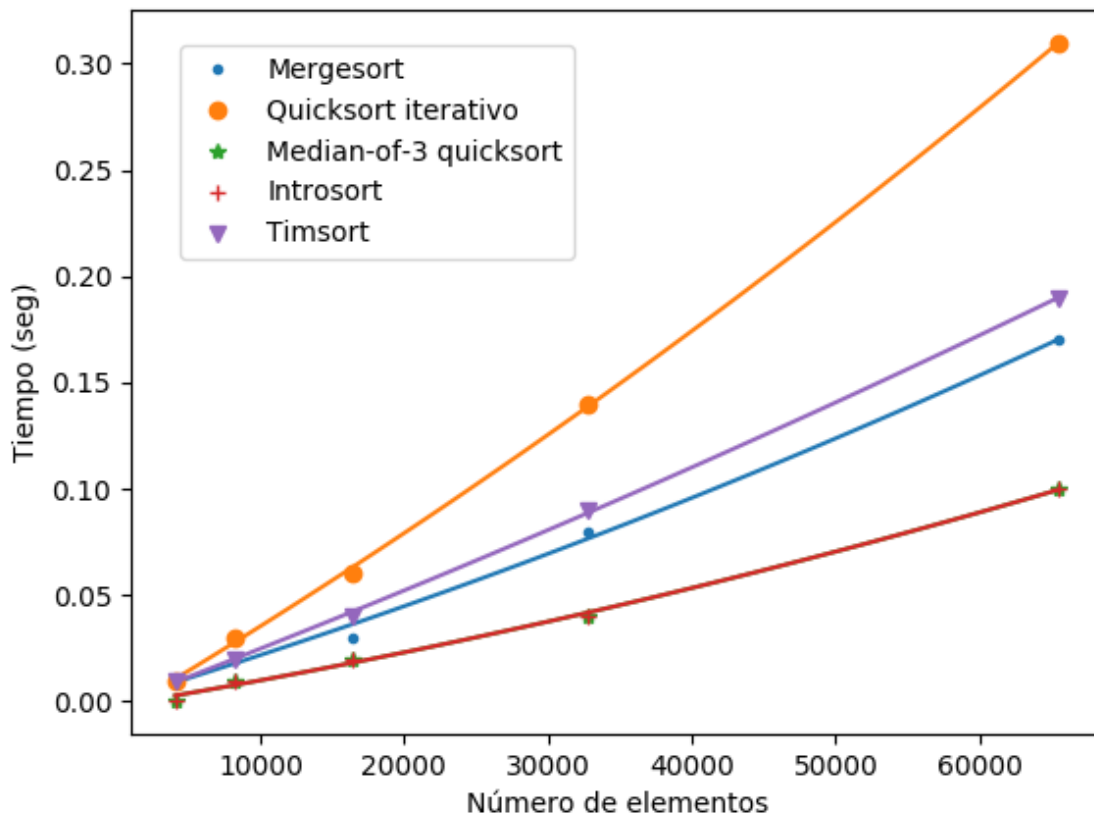
Gráfica Tipo 6 Algoritmos Cuasi-lineales

Secuencias casi-ordenadas 2, o tipo 7: Dado un conjunto ordenado de N elementos de tipo entero, se escogen al azar $n/4$ pares de elementos que se encuentran separados 4 lugares, entonces se intercambian los pares.

N.º Elem	Mergesort	QS iter	QS simple	Med-of-3	Introsort	3-way	DP	Timsort
4096	0.01 sg	0.01 sg	0.93 sg	0.0 sg	0.0 sg	0.32 sg	0.31 sg	0.01 sg
8192	0.02 sg	0.03 sg	3.76 sg	0.01 sg	0.01 sg	1.25 sg	1.22 sg	0.02 sg
16384	0.03 sg	0.06 sg	15.13 sg	0.02 sg	0.02 sg	5.01 sg	4.91 sg	0.04 sg
32768	0.08 sg	0.14 sg	61.94 sg	0.04 sg	0.04 sg	20.9 sg	20.49 sg	0.09 sg
65536	0.17 sg	0.31 sg	261.88 sg	0.1 sg	0.1 sg	91.14 sg	85.75 sg	0.19 sg



Gráfica Tipo 7



Gráfica Tipo 7 Algoritmos Cuasi-lineales

Conclusiones

- ✓ Se observó que todos los algoritmos tienen comportamiento cuasi-lineal en el caso promedio (secuencias tipo 1), cuando los elementos de la secuencia a ordenar están desordenados aleatoriamente.
- ✓ En secuencias de tipo 1, el algoritmo de ordenación más rápido es dual-pivot quicksort, seguido por quicksort simple.
- ✓ Como era de esperarse por lo visto en teoría, para secuencias de tipo 2 de N elementos, el algoritmo quicksort simple y el dual pivot quicksort están en su peor caso y son muy ineficientes, realizando alrededor de $O(N^2)$ operaciones. Es decir son de orden cuadrático para este caso.

- ✓ Para secuencias de tipo 3, se tiene que quicksort simple, Dual-pivot quicksort y quicksort with 3 way partitioning tendieron al orden cuadrático. Éstos algoritmos están en su peor caso, por lo que dejan de ser $O(n \cdot \log(n))$ y se convierten en $O(N^2)$.
- ✓ Por otro lado para secuencias del tipo 3, el algoritmo con mejor desempeño fue median of 3 quicksort, dado que la técnica de pivoteo usando la media entre los extremos y el medio de cada sub arreglo permitió evitar el peor caso del quicksort: caer en un extremo del subarreglo.
- ✓ Cuando se busca ordenar secuencias cero-uno (tipo 4) en tiempo cuasi-lineal, el quicksort iterativo es el más veloz, seguido por el quicksort with 3 way partitioning. Quicksort simple sigue estando en su peor caso, presentando orden cuadrático. Para secuencias de éste tipo todos los algoritmos a excepción del quicksort tuvieron buen desempeño.
- ✓ En secuencias de tipo 5 quicksort iterativo estuvo en su peor caso por primera vez; éste mostró un crecimiento asintótico de orden $O(N^2)$. Mergesort fue el más veloz entre todos en éste caso.
- ✓ De igual manera para las secuencias tipo 6 y 7, Quicksort fue el más lento de todos.
- ✓ Podemos concluir que los algoritmos más volátiles a sufrir crecimiento asintótico cuadrático son aquellos basados en quicksort que no toman en consideración el posicionamiento del pivote a la hora de realizar la partición de la secuencia a ordenar.
- ✓ Los algoritmos menos volátiles son: Timsort, Mergesort, Introsort y Median of 3 quicksort. Ésto no es sorpresa debido a que éstos algoritmos para su peor caso, son de crecimiento asintótico del orden $O(n \cdot \log(n))$.