

## Informe – Proyecto no. 1

El tipo de equipo en que se realizaron las pruebas fue: Laptop Sony Vaio, AMD E-350, disco 101,4gb, 1,6gb RAM con ubuntu 14.04 lts.

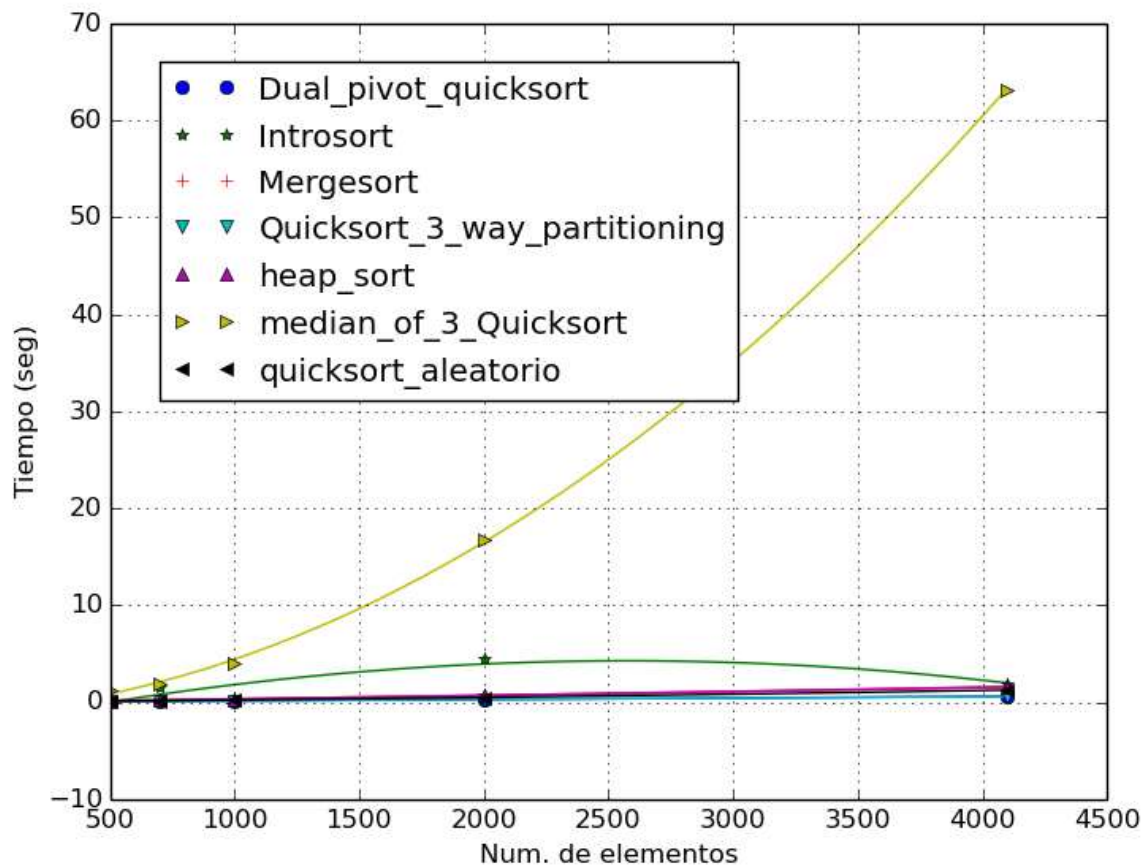
### Prueba con conjunto de datos 1: (Enteros aleatorios)

La prueba sobre la cantidad de elementos que se han pedido, en el equipo que utilizamos para realizar las pruebas, se ejecutó durante alrededor de 8 horas sin arrojar el resultado a tiempo para obtener una gráfica. A continuación agregamos únicamente los tiempos que logramos obtener desde las 2:30 pm hasta las 10:00 pm del 25 de febrero. El algoritmo Med-of-3 QS se tardó un número de segundos que no se pudo comprobar en el arreglo de 65.536 elementos, pues en el primer intento necesitó de 8314.385 segundos que no tuvimos disponibles para probar los otros dos intentos.

N	Mergesort	Heapsort	QS Random	Med-of-3 QS	Introsort	Dual pivot QS	QS 3-way
4096	0.686	0.774	0.584	26.661	0.914	0.264	0.254
8192	1.490	1.664	1.498	117.720	2.050	0.527	0.510
16384	3.191	3.649	4.376	566.724	4.348	1.145	1.019
32768	6.899	7.840	14.308	1657.639	9.460	2.639	2.050
65536	14.652	16.715			25.364	5.714	4.081

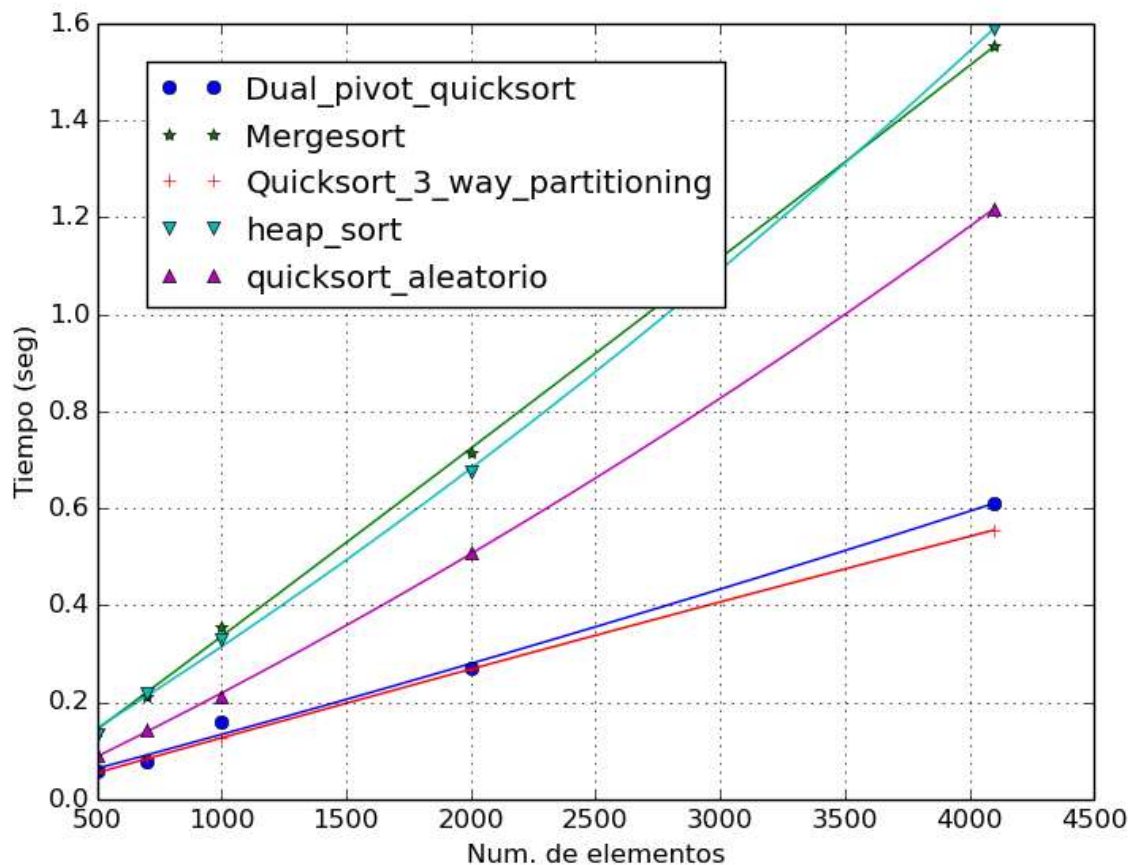
Se realizó de nuevo la prueba con arreglos más pequeños que no requieren de tanto tiempo para ejecutarse y de todas maneras, reflejan en la gráfica el resultado que queremos obtener a pequeña escala:

N	Mergesort	Heapsort	QS Random	Med-of-3 QS	Introsort	Dual pivot QS	QS 3-way
500	0.140	0.137	0.087	1.203	0.172	0.059	0.055
700	0.214	0.0206	0.147	1.913	1.553	0.076	0.086
1000	0.306	0.307	0.199	3.990	0.373	0.118	0.128
2000	0.700	0.682	0.484	16.725	4.456	0.268	0.265
4096	1.540	1.580	1.242	63.155	1.907	0.588	0.562



Prueba sobre los algoritmos lineales en el conjunto de datos 1:

N	Mergesort	Heapsort	QS Random	Dual pivot QS	QS 3-way
500	0.139	0.133	0.090	0.059	0.055
700	0.211	0.219	0.141	0.078	0.083
1000	0.356	0.328	0.210	0.356	0.125
2000	0.715	0.674	0.509	0.270	0.268
4096	1.552	1.589	1.216	0.611	0.554

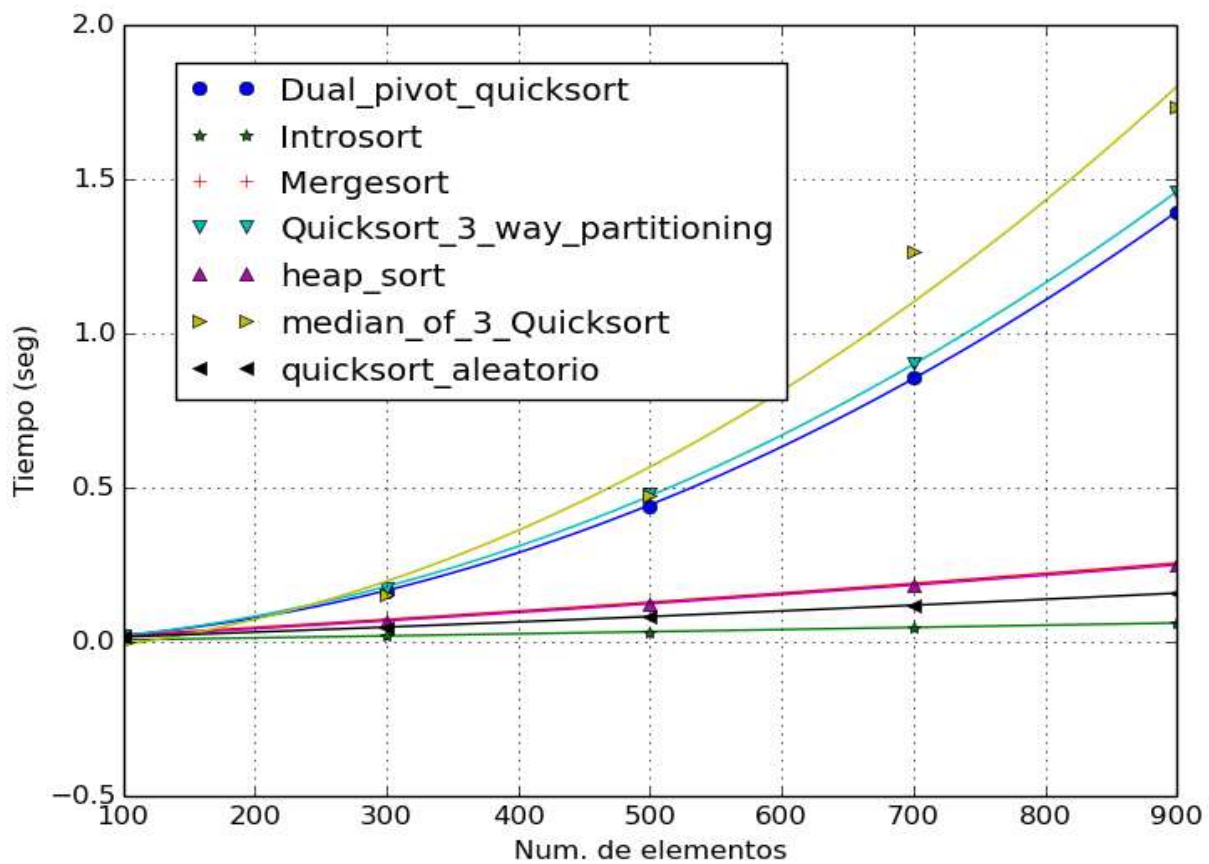


Es de esta manera como se observa la eficiencia de algoritmos como Quicksort (three way partitioning) y Dual Pivot Quicksort, sobre los demás algoritmos implementados; en este caso con aquellos con los que fue posible hacer comparaciones con un número muy grande de elementos en los arreglos, generados con números enteros grandes aleatorios.

## Prueba con conjunto de datos 2: (Orden inverso)

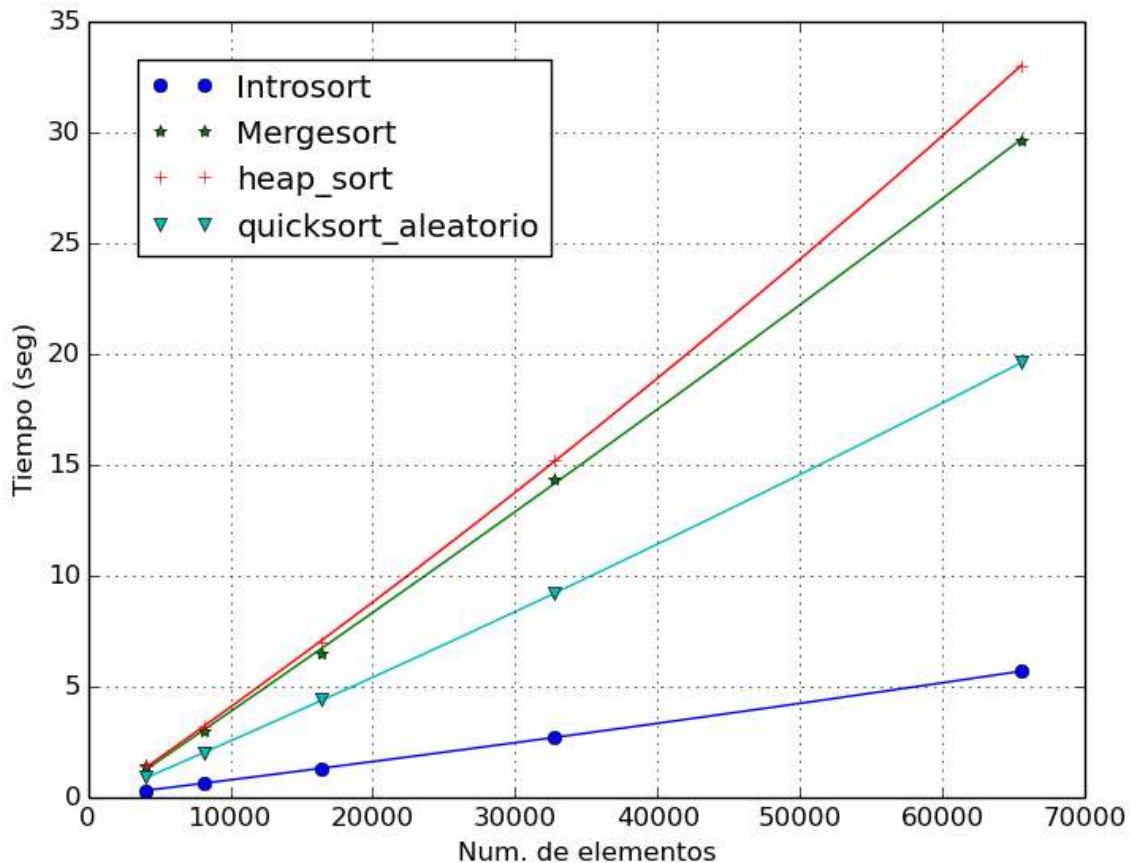
A pesar de haber cambiado el límite de recursiones y el ulimit al cuádruple para realizar esta prueba, siguió indicando que se excedió el número de recursiones para los arreglos de números dados, por lo tanto, se redujo el número de elementos de los arreglos a los siguientes:

N	Mergesort	Heapsort	QS Random	Med-of-3 QS	Introsort	Dual pivot QS	QS 3-way
100	0.020	0.017	0.012	0.027	0.005	0.020	0.021
300	0.077	0.070	0.051	0.160	0.018	0.177	0.017
500	0.134	0.124	0.081	0.490	0.033	0.474	0.475
700	0.199	0.184	0.123	1.304	0.048	0.891	0.935
900	0.261	0.251	0.182	1.726	0.059	1.488	1.519



Prueba sobre los algoritmos lineales en el conjunto de datos 2:

N	Introsort	Mergesort	Heapsort	Quicksort Aleatorio
4096	0.302	1.404	1.437	0.905
8192	0.622	2.989	3.165	1.964
16384	1.295	6.464	6.960	4.376
32768	2.693	14.312	15.227	9.198
65536	5.678	29.657	33.002	19.603

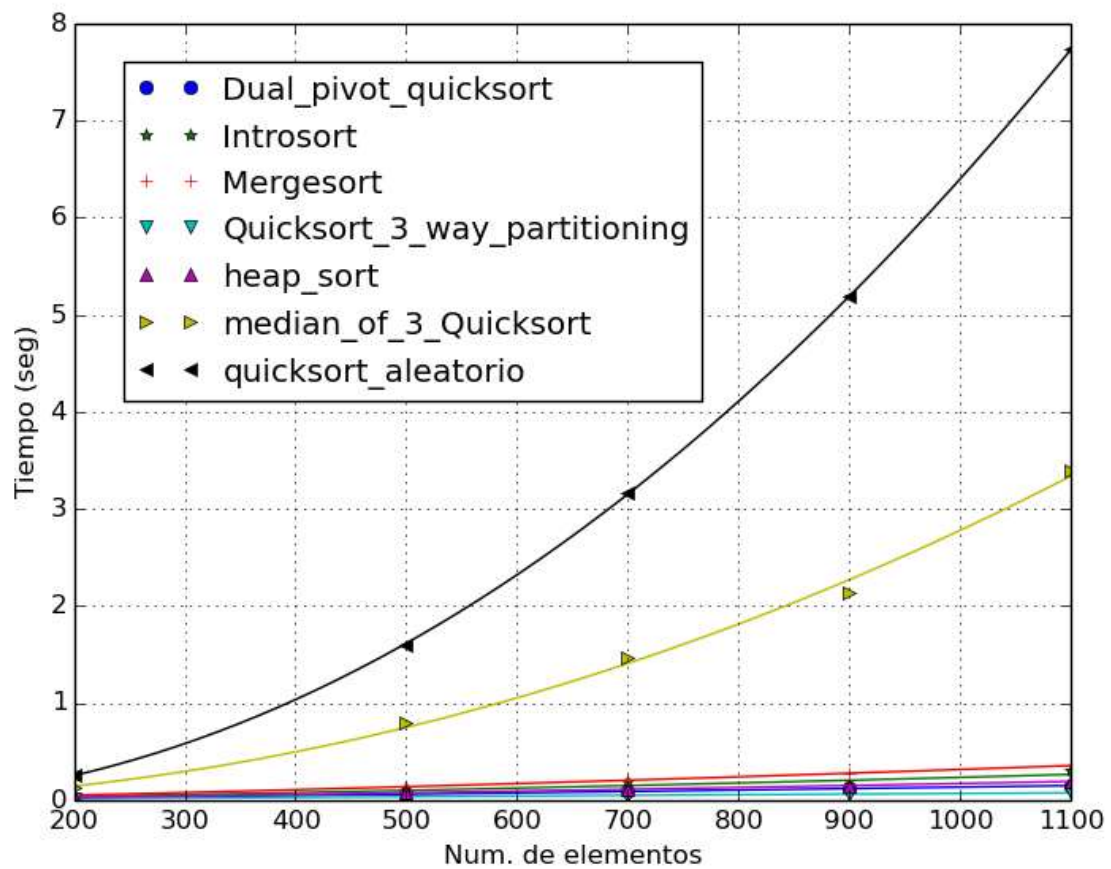


Este caso muestra la eficiencia del algoritmo Quicksort (three way partitioning), sobre los demás. Al menos con arreglos muy grandes. Lo cual permite entender la definición dada en el enunciado del proyecto, del mismo algoritmo; de por qué compañías de reconocido prestigio usan este algoritmo para sus desarrollos. Este comportamiento y preferencia se verá reflejado también en los próximos conjuntos de datos.

### Prueba con conjunto de datos 3: (Cero-uno)

A pesar de haber cambiado el límite de recursiones y el ulimit al cuádruple para realizar esta prueba, siguió indicando que se excedió el número de recursiones para los arreglos de números dados, por lo tanto, se redujo el número de elementos de los arreglos a los siguientes:

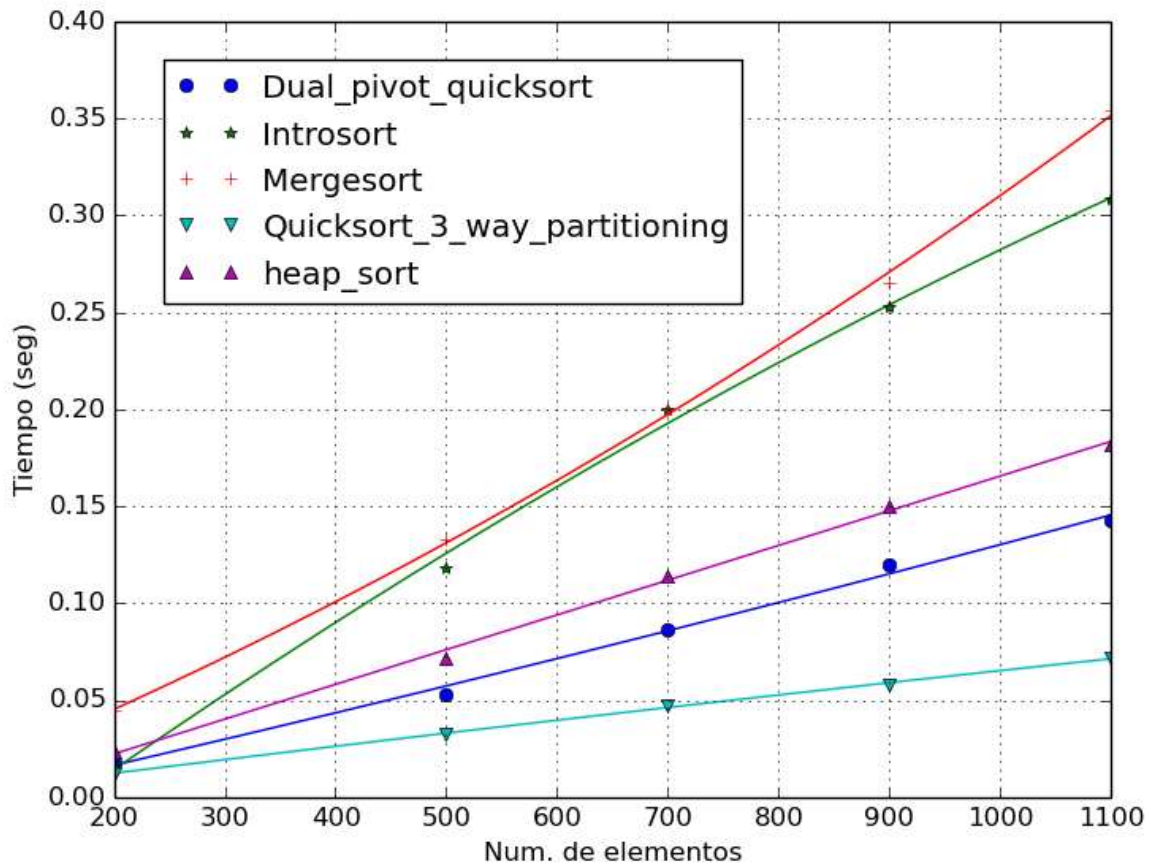
N	Mergesort	Heapsort	QS Random	Med-of-3 QS	Introsort	Dual pivot QS	QS 3-way
200	0.048	0.027	0.253	0.120	0.018	0.018	0.013
500	0.136	0.074	1.602	0.788	0.128	0.055	0.034
700	0.206	0.109	3.167	1.458	0.192	0.089	0.047
900	0.270	0.149	5.181	2.143	0.102	0.119	0.060
1100	0.356	0.191	7.734	3.393	0.307	0.146	0.073



Prueba sobre los algoritmos lineales en el conjunto de datos 3:

N	Dual Pivot Quicksort	Introsort	Mergesort	Quicksort three-way	Heapsort
200	0.018	0.016	0.044	0.012	0.023
500	0.053	0.118	0.132	0.032	0.072
700	0.086	0.200	0.200	0.047	0.114
900	0.119	0.253	0.265	0.058	0.150
1100	0.143	0.308	0.354	0.072	0.182

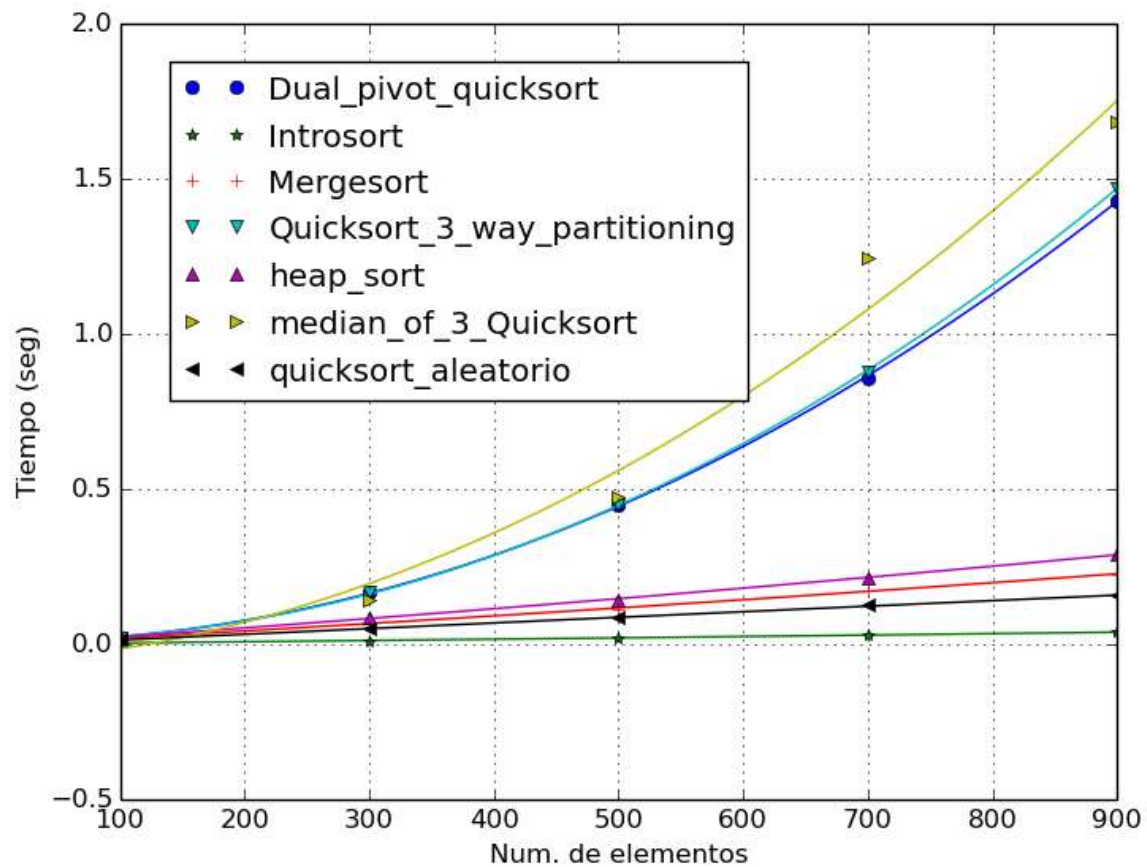




#### Prueba con conjunto de datos 4: (Ordenado)

A pesar de haber cambiado el límite de recursiones y el ulimit al cuádruple para realizar esta prueba, siguió indicando que se excedió el número de recursiones para los arreglos de números dados, por lo tanto, se redujo el número de elementos de los arreglos a los siguientes:

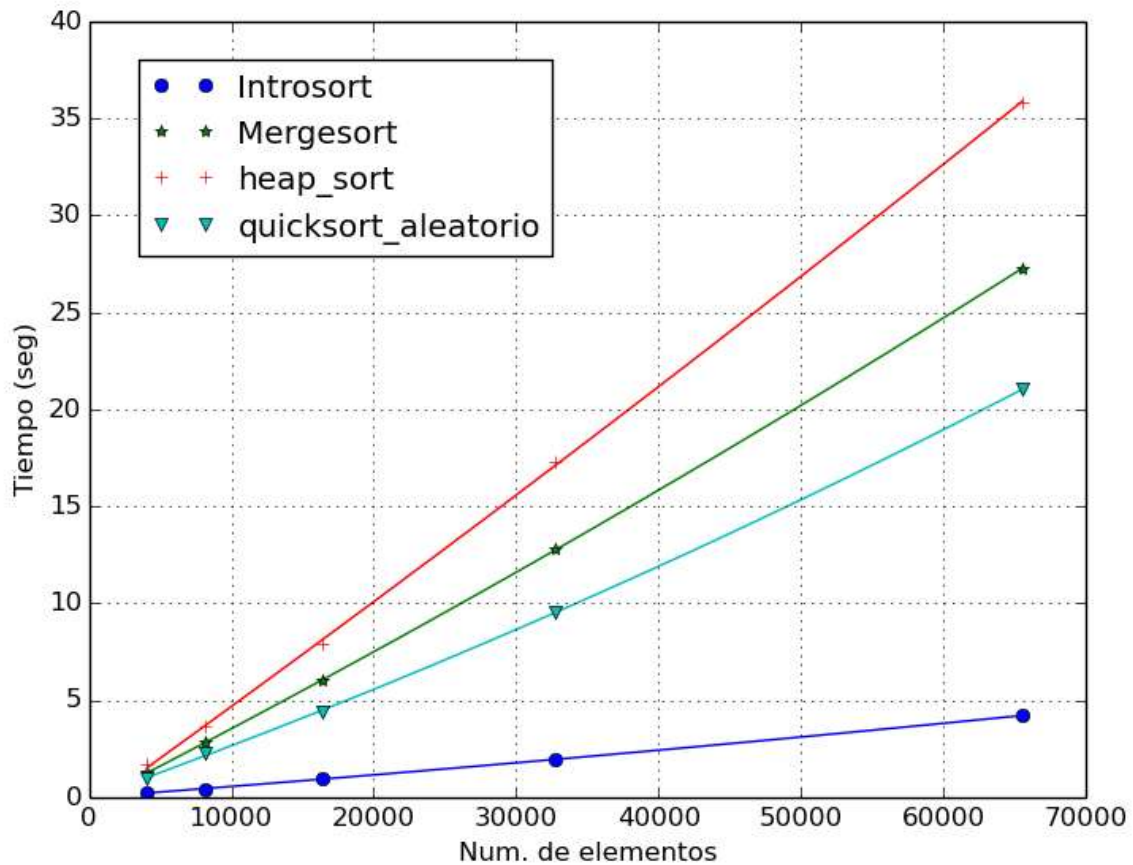
N	Mergesort	Heapsort	QS Random	Med-of-3 QS	Introsort	Dual pivot QS	QS 3-way
100	0.018	0.020	0.013	0.025	0.003	0.020	0.021
300	0.066	0.086	0.048	0.141	0.011	0.165	0.167
500	0.113	0.143	0.084	0.472	0.019	0.449	0.446
700	0.172	0.213	0.124	1.245	0.029	0.855	0.877
900	0.225	0.288	0.156	1.680	0.037	1.429	1.470



Prueba sobre los algoritmos lineales en el conjunto de datos 4:

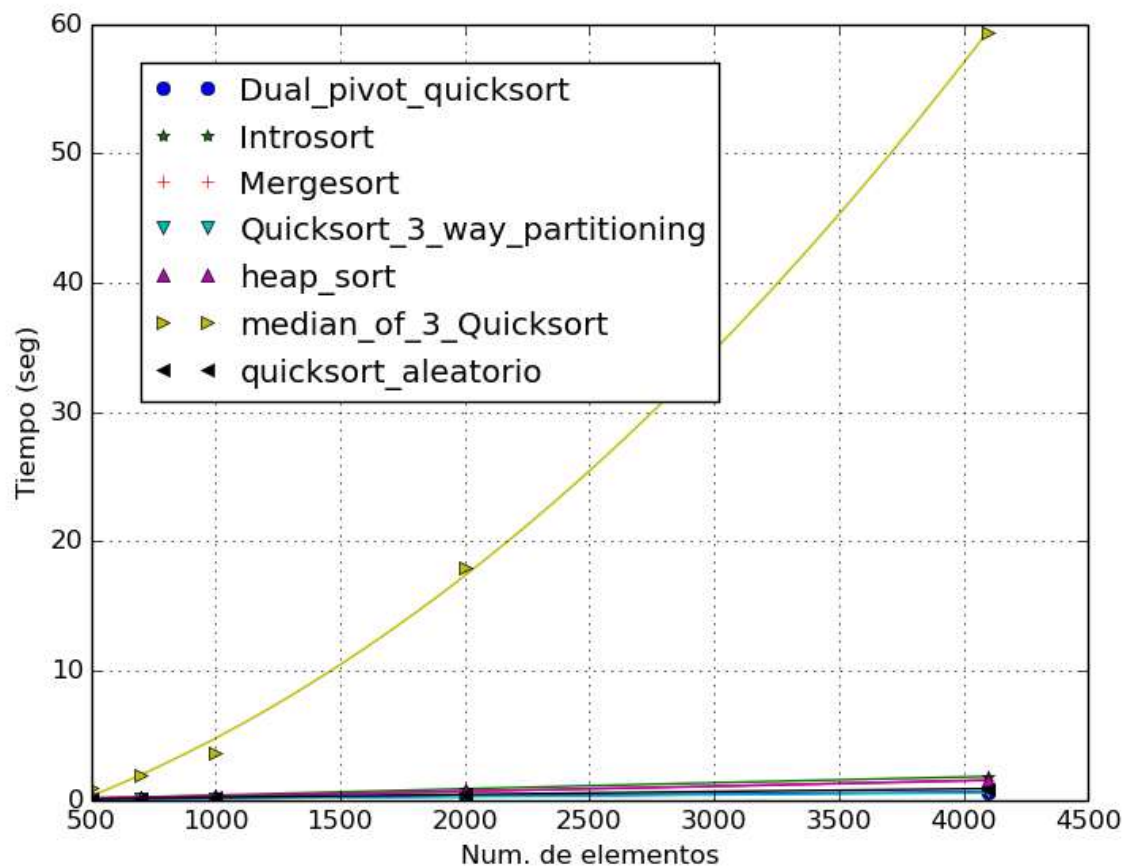
N	Introsort	Mergesort	Heapsort	Quicksort Aleatorio
4096	0.220	1.281	1.658	0.939
8192	0.424	2.797	3.662	2.274
16384	0.924	6.005	7.865	4.412
32768	1.947	12.776	17.272	9.532
65536	4.204	27.260	35.851	21.027





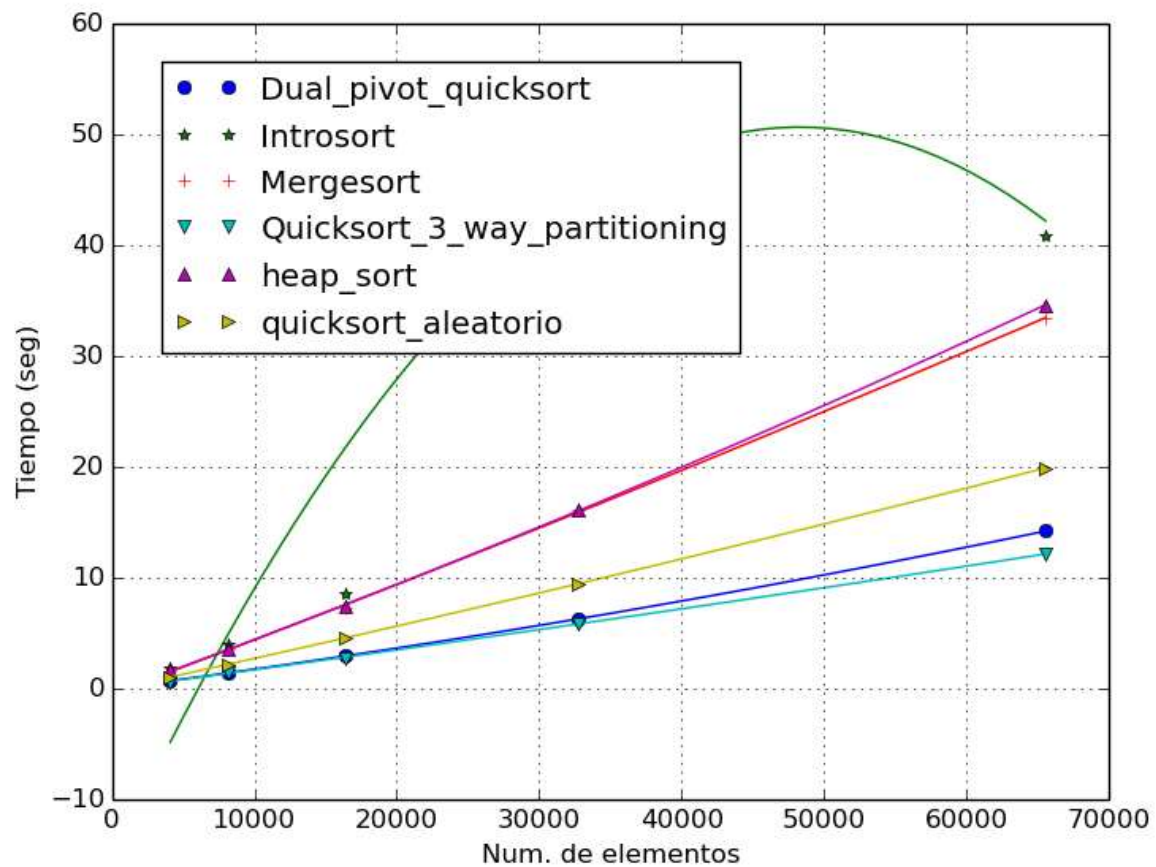
Prueba con conjunto de datos 5: (Reales aleatorios entre cero y uno)

N	Mergesort	Heapsort	QS Random	Med-of-3 QS	Introsort	Dual pivot QS	QS 3-way
500	0.141	0.140	0.085	0.957	0.066	0.057	0.053
700	0.213	0.211	0.127	1.959	0.263	0.084	0.078
1000	0.310	0.305	0.184	3.608	0.395	0.131	0.115
2000	0.670	0.677	0.437	17.930	0.843	0.291	0.250
4096	1.520	1.557	0.872	59.314	1.826	0.621	0.566



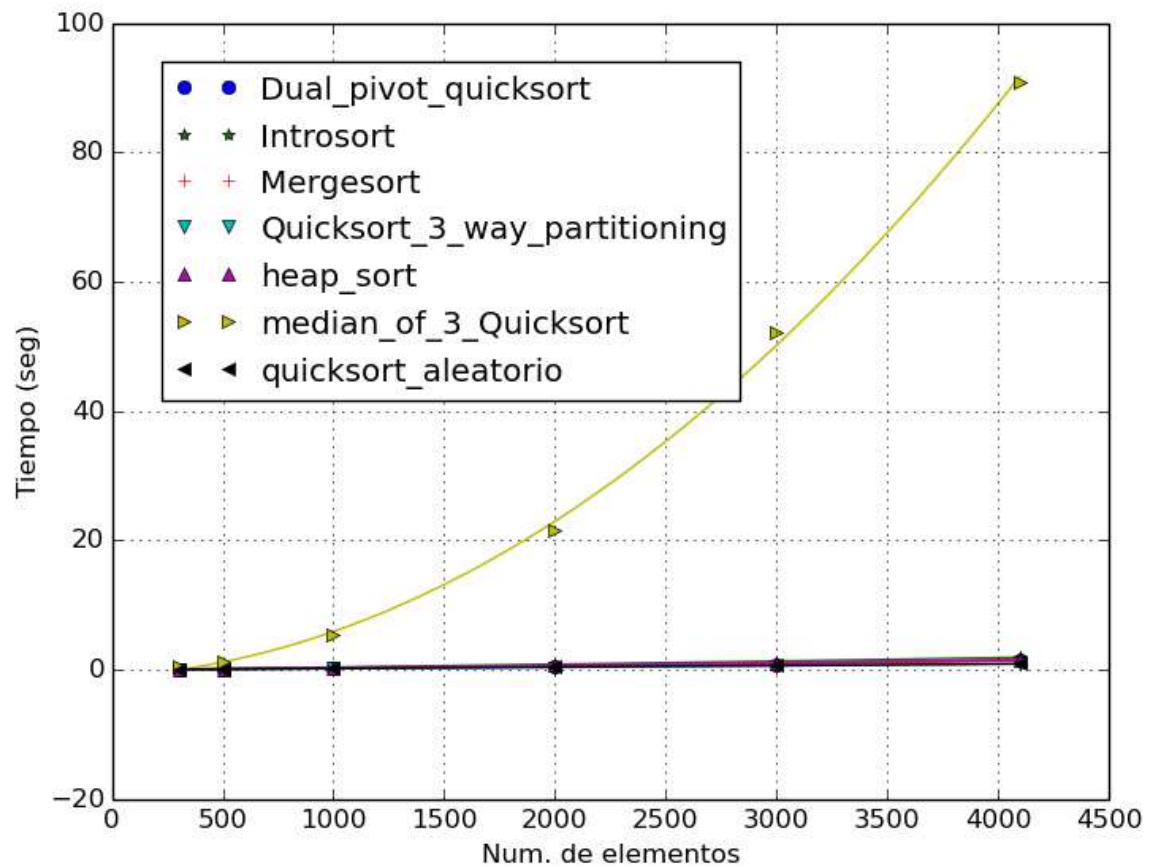
Prueba sobre los algoritmos lineales en el conjunto de datos 5:

N	Dual Pivot Quicksort	Introsort	Mergesort	Quicksort three-way	Heapsort	Quicksort Aleatorio
4096	0.650	1.869	1.555	0.584	1.613	0.991
8192	1.340	4.014	3.569	1.464	3.467	2.159
16384	3.029	8.576	7.222	2.691	7.392	4.570
32768	6.195	52.451	16.066	5.847	16.116	9.412
65536	14.186	40.807	33.404	12.103	34.575	19.879



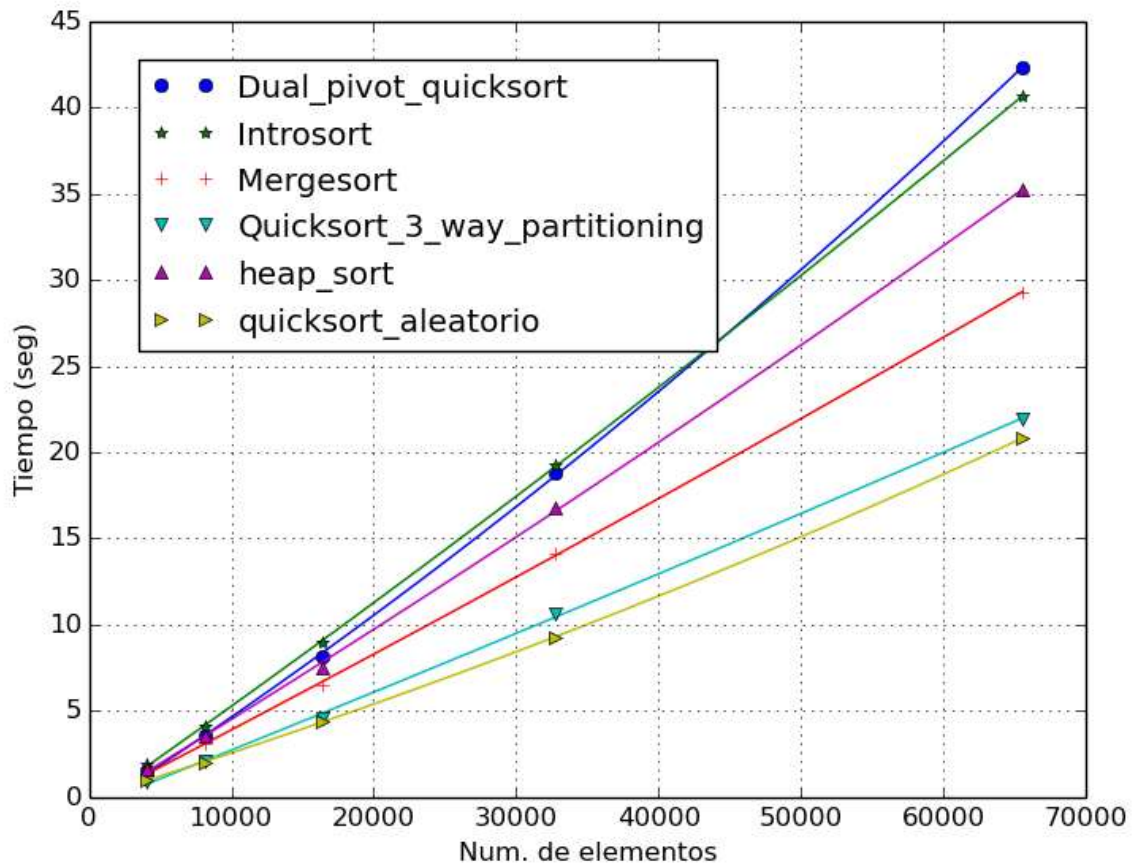
Prueba con conjunto de datos 6: (Mitad)

N	Mergesort	Heapsort	QS Random	Med-of-3 QS	Introsort	Dual pivot QS	QS 3-way
500	0.127	0.140	0.089	1.351	0.175	0.095	0.073
1000	0.276	0.315	0.206	5.376	0.371	0.233	0.167
2000	0.61	0.709	0.401	21.549	0.836	0.555	0.376
3000	0.962	1.092	0.636	52.269	1.341	0.841	0.599
4096	1.362	1.598	0.907	90.972	1.895	1.525	0.916



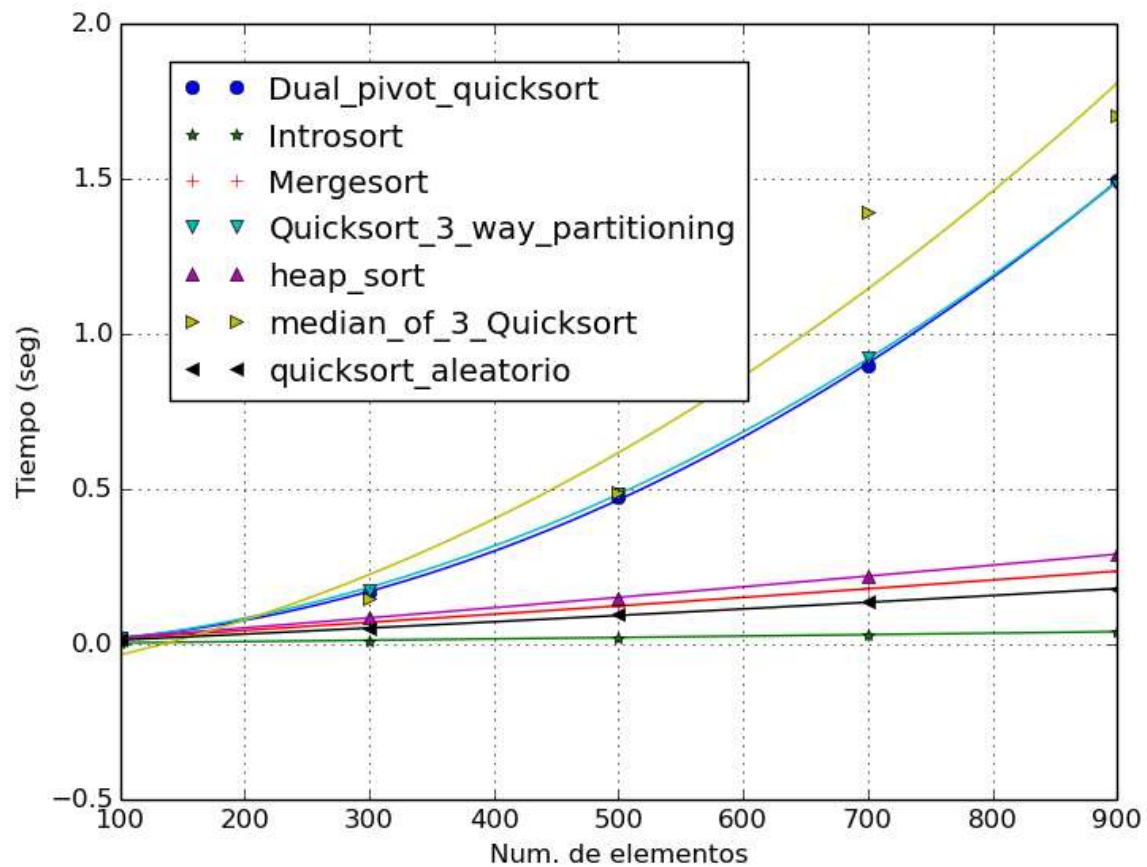
Prueba sobre los algoritmos lineales en el conjunto de datos 6:

N	Dual Pivot Quicksort	Introsort	Mergesort	Quicksort three-way	Heapsort	Quicksort Aleatorio
4096	1.499	1.917	1.442	0.912	1.602	0.989
8192	3.538	4.096	3.085	2.108	3.594	2.015
16384	8.110	8.977	6.518	4.550	7.549	4.420
32768	18.835	19.271	14.131	10.647	16.764	9.269
65536	42.314	40.646	29.297	21.951	32.216	20.826



Prueba con conjunto de datos 7: (Casi ordenado)

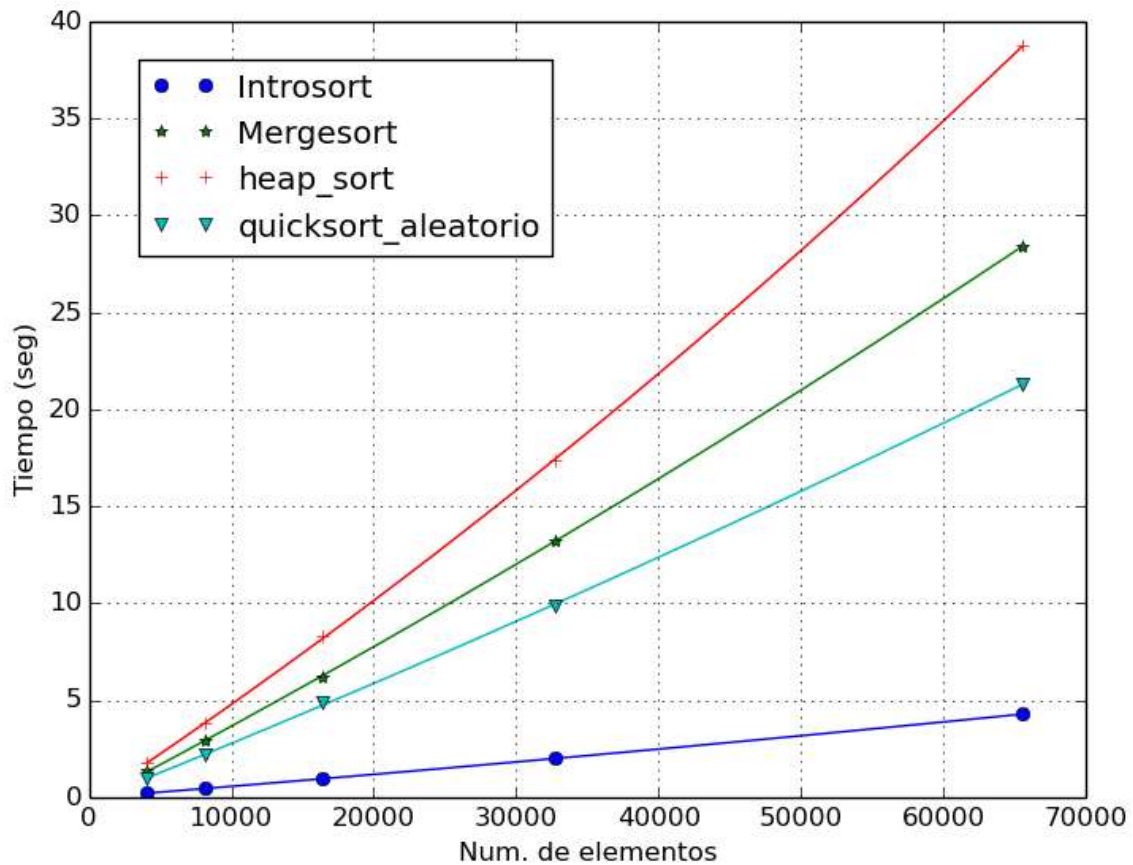
N	Mergesort	Heapsort	QS Random	Med-of-3 QS	Introsort	Dual pivot QS	QS 3-way
100	0.018	0.020	0.013	0.024	0.004	0.020	0.021
300	0.069	0.083	0.049	0.146	0.011	0.165	0.173
500	0.120	0.148	0.093	0.486	0.020	0.474	0.484
700	0.180	0.221	0.134	1.393	0.031	0.896	0.924
900	0.233	0.288	0.178	1.703	0.039	1.493	1.485



Prueba sobre los algoritmos lineales en el conjunto de datos 7:

N	Introsort	Mergesort	Heapsort	Quicksort Aleatorio
4096	0.205	1.338	1.740	0.977
8192	0.434	2.922	3.781	2.147
16384	0.938	6.206	8.324	4.858
32768	1.995	13.241	17.356	9.896
65536	4.275	28.393	38.728	21.288





Finalmente, se aprecian en las gráficas presentadas las distintas eficiencias de los algoritmos, cuyas preferencias se ven determinadas por el tipo de arreglo a ordenar. Se puede ver cómo el Introsort tiene algunas ventajas sobre el Quicksort Three Way Partitioning en algunos conjuntos de datos, así como también estos se encuentran en varias desventajas frente a otros como el Dual Pivot Quicksort o el Median of Three Quicksort; sin dejar atrás los algoritmos básicos estudiados anteriormente que también cumplen un papel fundamental para el estudio y las distintas comparaciones.