



UNIVERSIDAD SIMÓN BOLÍVAR
Verónica Machado 10-10407
Roberto Romero 10-10642

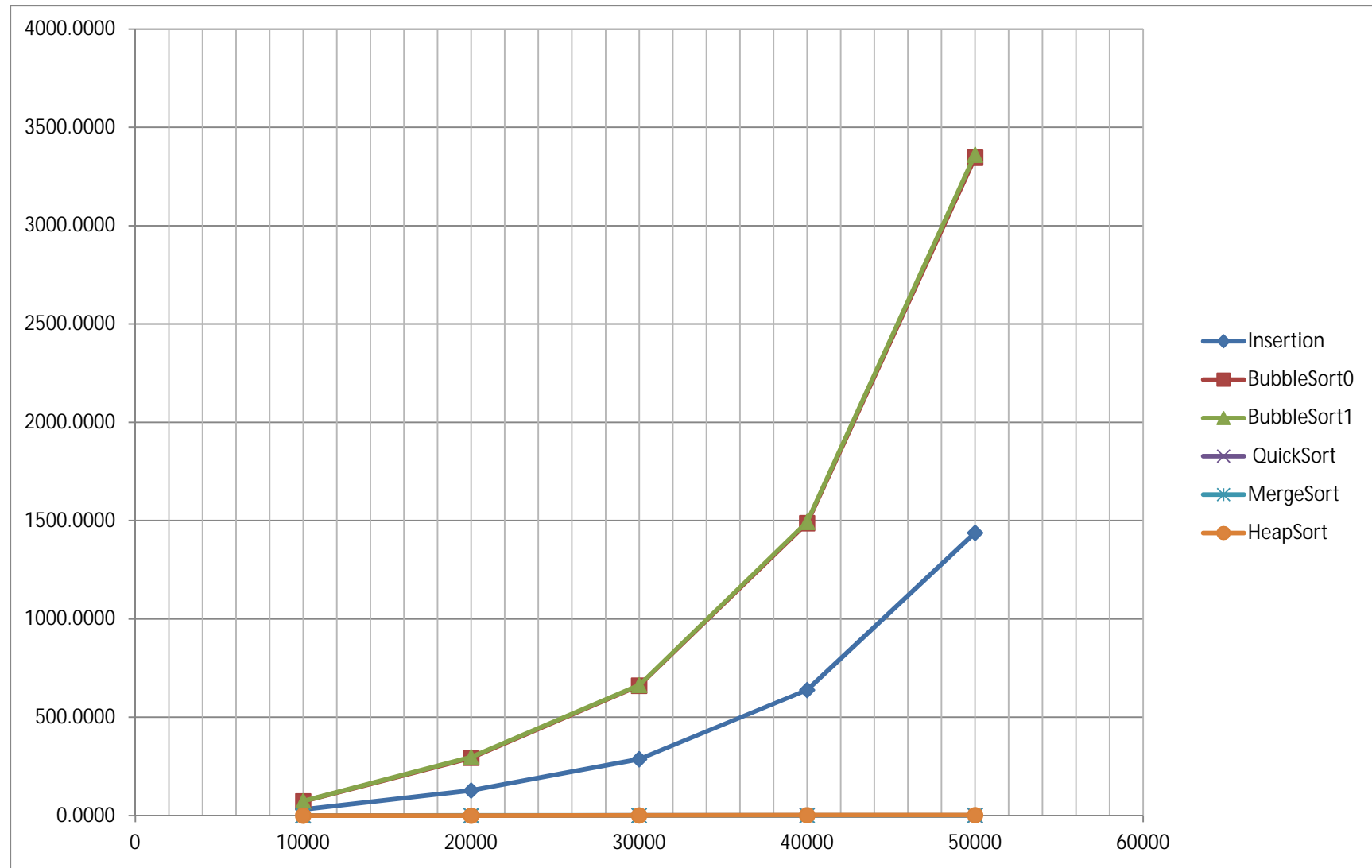
Informe Proyecto 3

TABLA 1

Resultados del tiempo de ejecución promedio, en milisegundos, de cinco (5) corridas de los algoritmos de ordenamiento con variable número de elementos.

Número de Elementos	Insertion	BubbleSort0	BubbleSort1	QuickSort	MergeSort	HeapSort
10000	31,4266	72,9078	73,4537	0,2080	0,3220	0,4700
20000	128,1773	294,1525	296,9551	0,4520	0,6850	0,1025
30000	287,3336	661,0505	664,0422	0,6890	1,0670	1,5980
40000	639,3172	1487,3636	1494,0949	1,0320	1,5000	2,2301
50000	1438,4637	3346,5681	3361,7135	1,3572	1,9188	2,8496

GRAFICA DE LA TABLA 1



ANALISIS DE LOS RESULTADOS DE LA TABLA 1

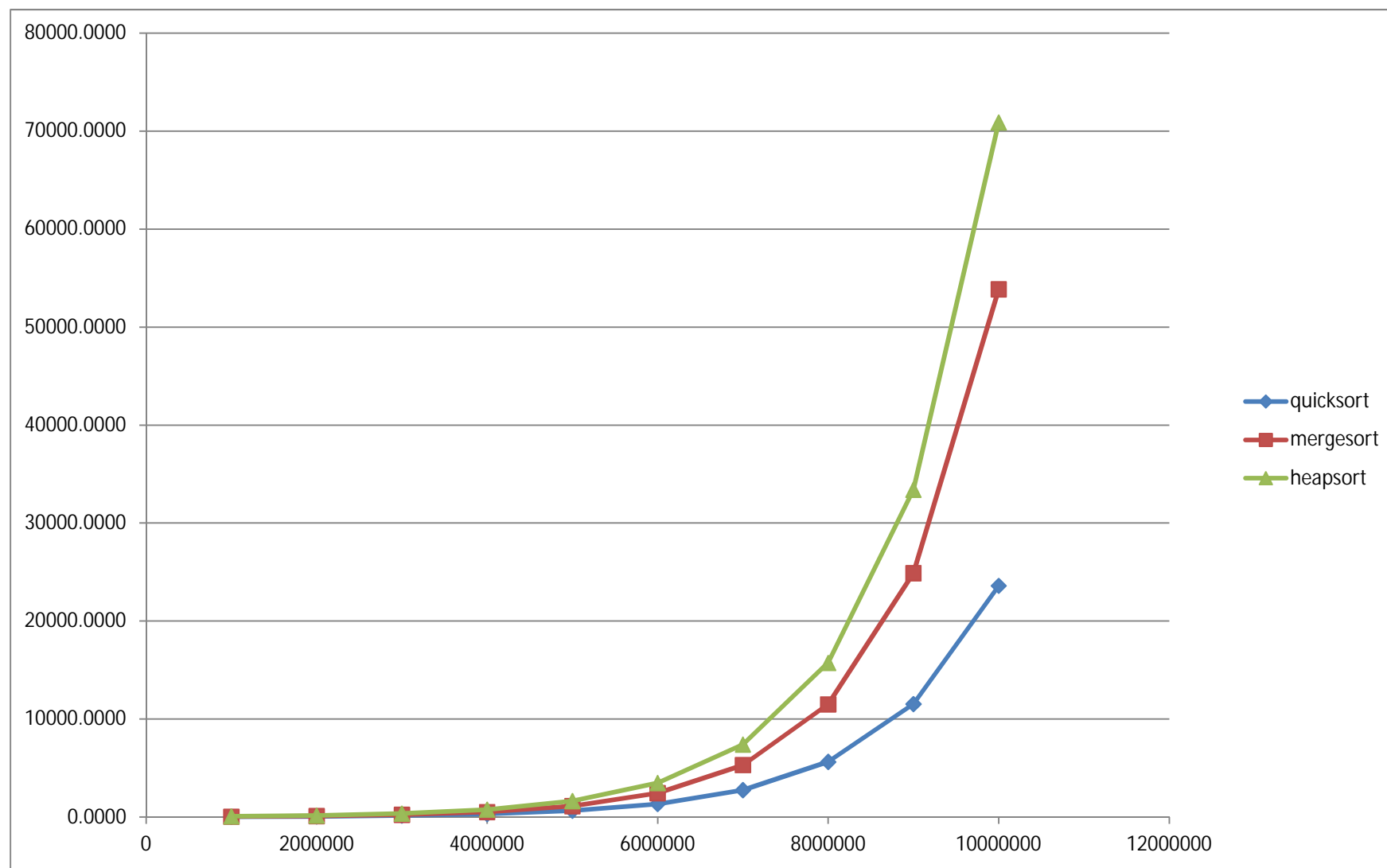
Podemos observar de manera evidente que los algoritmos BubbleSort0, BubbleSort1 e Insertion son aquellos que son menos óptimos en tiempo de corrida, debido a que se tardaron en ordenar los arreglos aleatorios con número variable de elementos mucho más que los otros tres algoritmos que se utilizaron durante el experimento, especialmente en los casos con arreglos con más de 20000 elementos. Esto es debido a que son de orden $\Theta(n^2)$, es decir entre mayor es el número de elementos a ordenar, su tiempo de corrida aumenta proporcionalmente, mientras que en el caso de los algoritmos de orden $O(n \log n)$, es decir, QuickSort, HeapSort y MergeSort, mantienen sus tiempos de corrida cercanos o menores a los 3 milisegundos durante todo el experimento.

TABLA 2

Resultados del tiempo de ejecución promedio, en milisegundos, de cinco (5) corridas de los algoritmos de ordenamiento de orden $O(n \log n)$ con variable número de elementos.

Número de Elementos	QuickSort	MergeSort	HeapSort
1000000	37,7399	52,9828	79,7353
2000000	77,1792	112,4258	172,4642
3000000	157,8337	243,1724	365,9570
4000000	322,7746	525,9720	776,5352
5000000	660,0835	1137,6560	1647,7533
6000000	1349,8900	2460,7035	3496,4175
7000000	2760,5646	5322,4013	7419,1536
8000000	5645,4353	11512,1368	15742,9256
9000000	11545,0805	24900,2820	33405,3882
10000000	23610,0275	53858,2939	70883,9000

GRAFICA DE LA TABLA 2



ANALISIS DE LOS RESULTADOS DE LA TABLA 2

Al analizar los algoritmos de orden $O(n \log n)$ con arreglos con un número mayor de elementos, observamos que se empiezan a presentar diferencias respecto al tiempo de corrida. Podemos observar que QuickSort es el algoritmo más rápido, ordenando arreglos de 10000000 (diez millones) de elementos en una fracción del tiempo de los demás y que HeapSort es el más lento.