Universidad del Valle de Guatemala
Departamento de ingeniería
Algoritmos y estructura de datos
Kevin Macario 17369
Rodrigo Urrutia 16139

Hoja de Trabajo 3 - Sorts

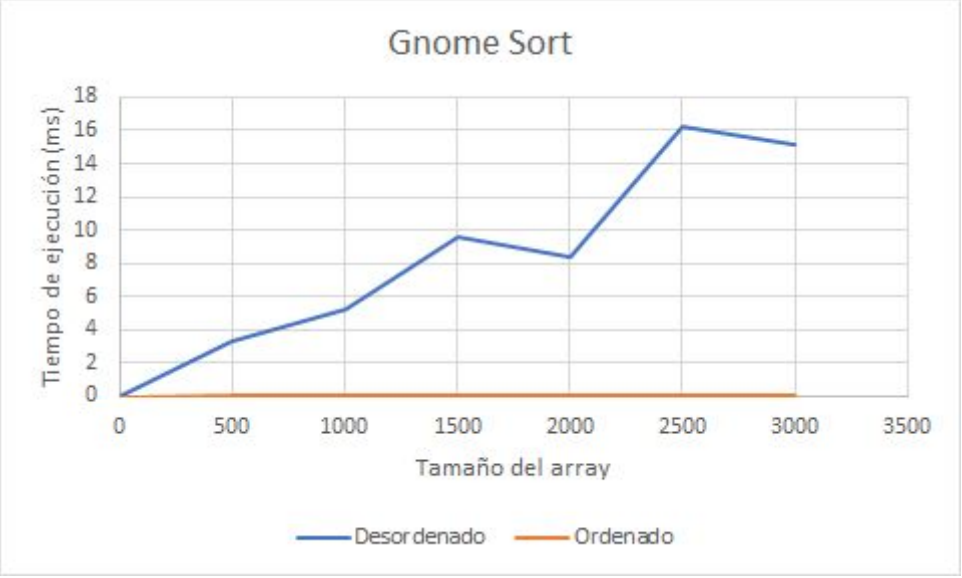
El Profile utilizado fue el ya implementado en NetBeans. Para poder obtener el tiempo de corrida de un método, se seleccionó la opción de clases para poder acceder a los tiempos respectivos de los 5 métodos implementados en la clase Sort. Estos tiempos se copiaron en excel y se ordenaron en tablas para realizar las gráficas.

GnomeSort:

Mejor caso: $O(n)$

Peor caso: $O(n^2)$

Promedio: $O(n^2)$



MergeSort:

Mejor caso: $O(nLogn)$

Peor caso: $O(nLogn)$

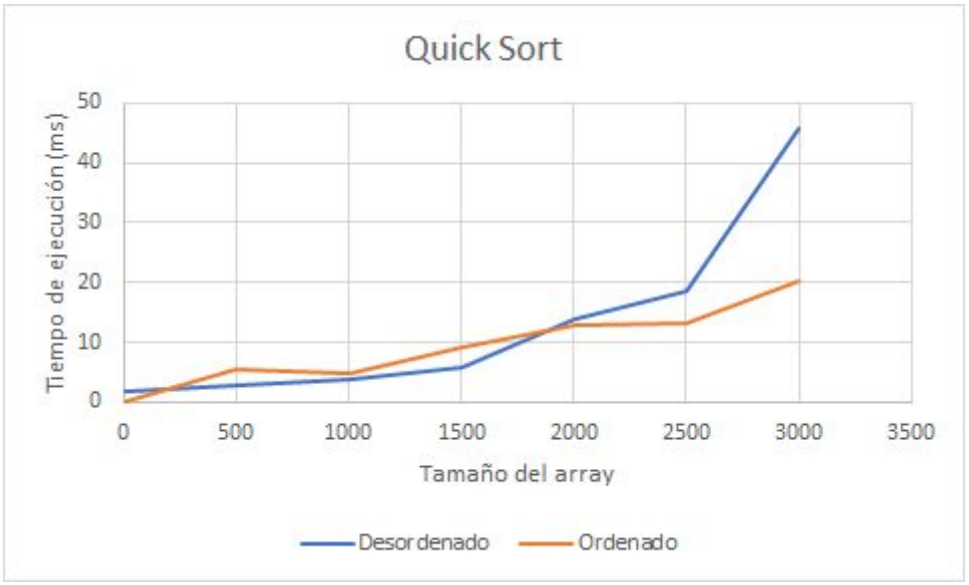
Promedio: $O(nLogn)$

QuickSort:

Mejor caso: $O(nLogn)$

Peor caso: $O(n^2)$

Promedio: $O(nLogn)$



InsertionSort:

Mejor caso: $O(n)$

Peor caso: $O(n^2)$

Promedio: $O(n^2)$

RadixSort:

Mejor caso: $O(nk)$

Peor caso: $O(nk)$

Promedio: $O(nk)$

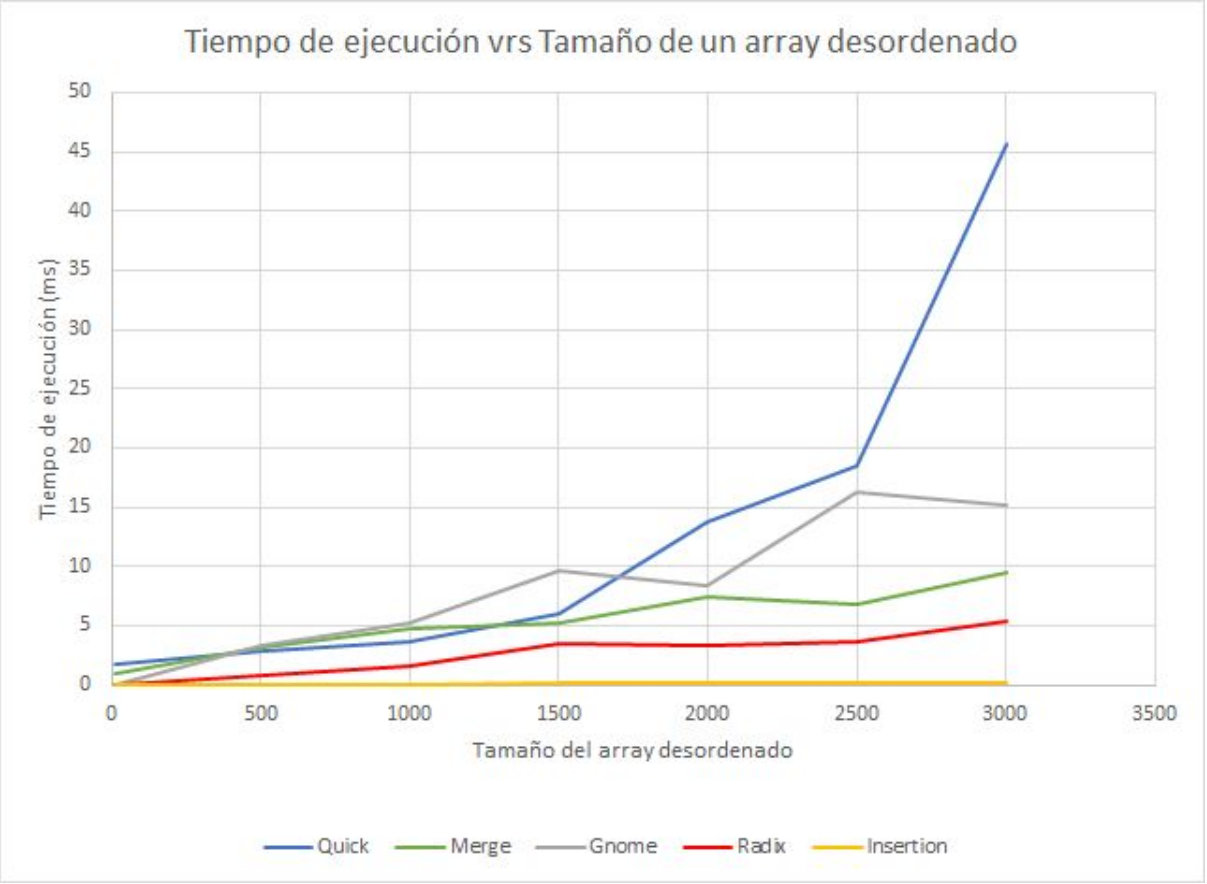
Tiempos de ejecución con array desordenado

Desorden (ms)	Quick	Merge	Gnome	Radix	Insertion
10	1.8	0.955	0.008	0.037	0.009
500	2.9	3.22	3.26	0.82	0.024
1000	3.64	4.78	5.2	1.61	0.041
1500	5.93	5.28	9.58	3.54	0.097
2000	13.8	7.47	8.43	3.34	0.085
2500	18.5	6.85	16.2	3.7	0.104
3000	45.7	9.55	15.1	5.4	0.192

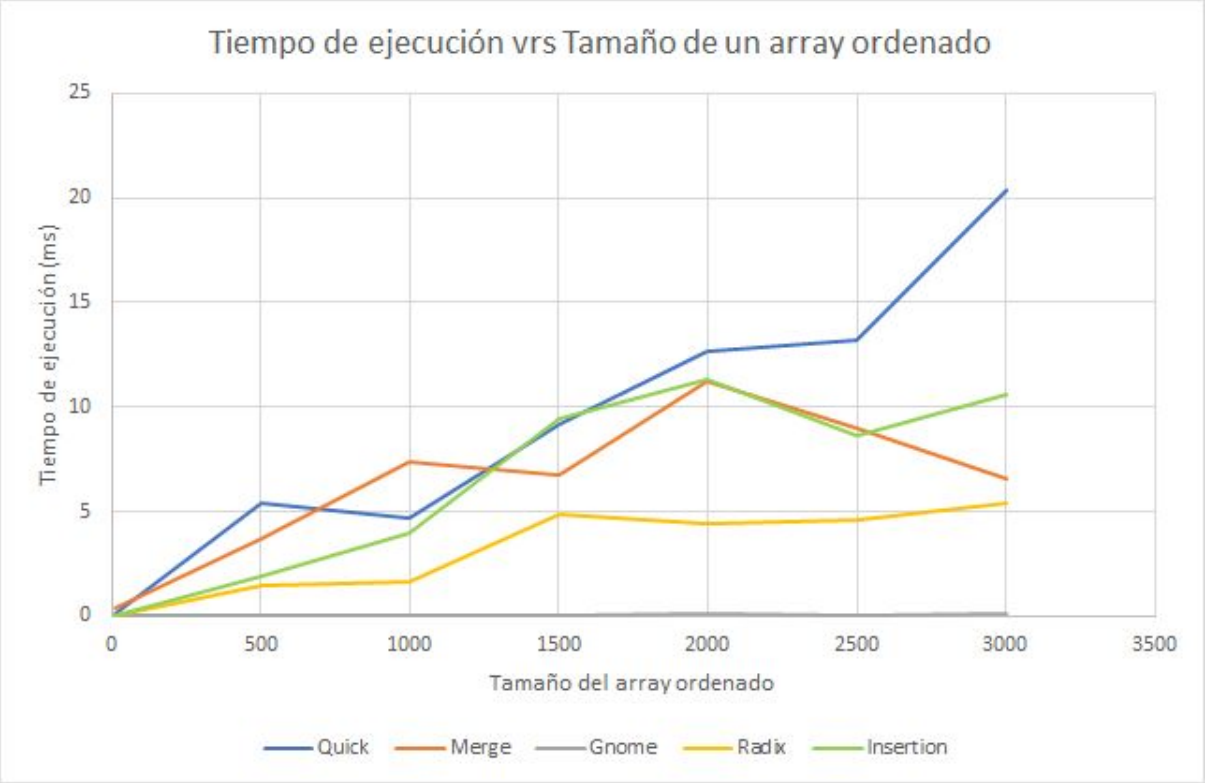
Tiempos de ejecución con array ordenado

Orden (ms)	Quick	Merge	Gnome	Radix	Insertion
10	0.077	0.406	0.01	0.038	0.012
500	5.43	3.71	0.024	1.5	1.91
1000	4.66	7.4	0.041	1.61	3.95
1500	9.16	6.73	0.06	4.88	9.44
2000	12.7	11.2	0.081	4.46	11.3
2500	13.2	8.99	0.066	4.59	8.66
3000	20.4	6.56	0.084	5.42	10.6

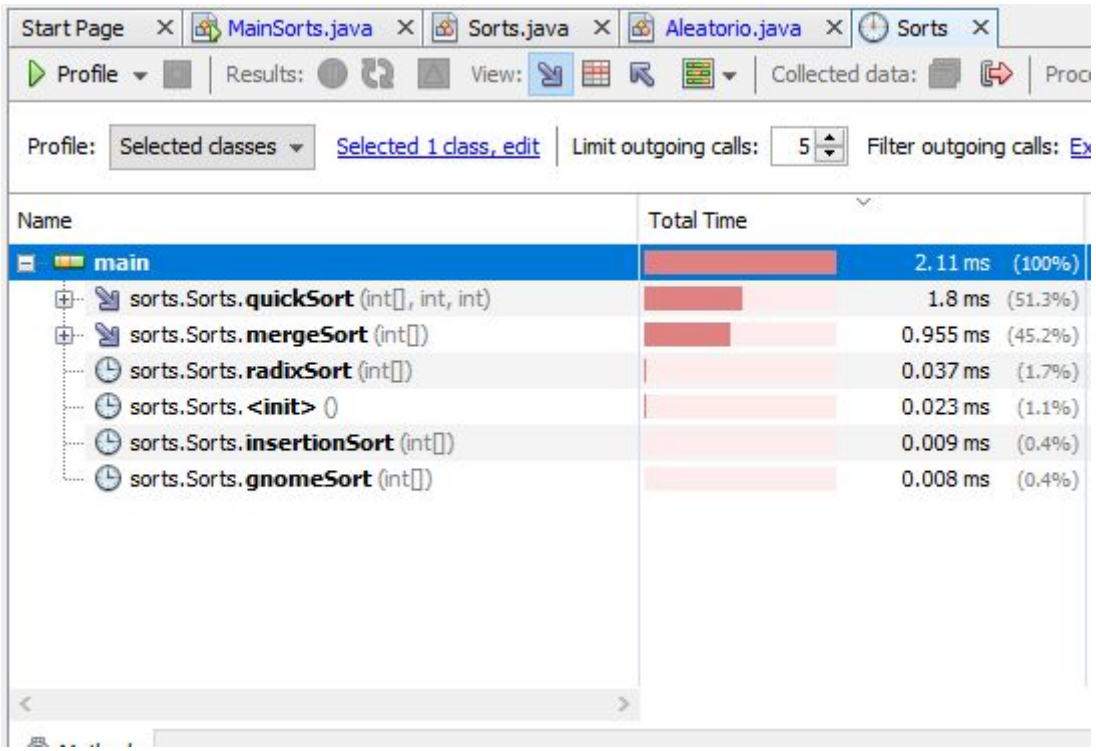
Gráfica de los sorts implementados con un array desordenado



Gráfica de los sorts implementados con un array ordenado



Profile con array de tamaño 10



Profile con array de tamaño 3000