COMP 3270 PA1

In general, Insertion sort performed the best on data of 10 and 1,000 arrays. This is surprising because in theory, selection sort would be the fastest sorting algorithm since its performance is measured most of the time in O (N × log N). The slowest algorithm used to sort the arrays was quick sort. It's not surprising for quick sort to have a slow run time for small arrays but it is surprising to see that insertion sort outperformed quick sort when it came to organizing the large array of 1,000. Insertion sort also performed the best on randomized arrays because of its break condition in the inner loop. The behavior of each of the four algorithms is exactly what they're predicted to be.

	n_{min}	t_{min}	n_{max}	t_{max}
SC	10	1500	1000	1121200
SS	10	1500	1000	1124100
SR	10	1500	1000	1127400
IC	10	600	1000	31200
IS	10	500	1000	31500
IR	10	500	1000	8900
MC	10	3700	1000	257900
MS	10	3700	1000	169500
MR	10	5200	1000	125300
QC	10	2600	1000	1429000
QS	10	3200	1000	1369700
QR	10	2500	1000	1459300

	t_{max}/t_{min}	n ratio	n ln(n) ratio	n^2 ratio	Behavior
SC	747.5	100	300	10000	Quadratic
SS	749.4	100	300	10000	Quadratic
SR	751.6	100	300	10000	Quadratic
IC	52	100	300	10000	Linear
IS	63	100	300	10000	Linear
IR	17.8	100	300	10000	Linear
MC	69.7	100	300	10000	Linear
MS	45.8	100	300	10000	Linear
MR	24.1	100	300	10000	Linear
QC	549.6	100	300	10000	nlgn
QS	428	100	300	10000	nlgn
QR	583.72	100	300	10000	nlgn