	t_{max}/t_{min}	n ratio	ln(n) rati	n^2 ratio	Behavior
SC	2975.64	55	78.92999	3025	n^2
SS	2764.346	55	78.92999	3025	n^2
SR	2733.058	55	78.92999	3025	n^2
IC	105.661	100	128.5714	10000	n
IS	108.9016	100	128.5714	10000	n
IR	5470.25	200	315.0515	40000	n^2
MC	2912.643	2000	3158.465	4000000	nln(n)
MS	3146	2000	3158.465	4000000	nln(n)
MR	2529.556	2000	3158.465	4000000	nln(n)
QC	3237.649	55	78.92999	3025	n^2
QS	6819.351	150	207.2759	22500	n^2
QR	3553.693	260	370.1766	67600	nln(n)

My results show a correlation with the ones in the pdf. The results show that for Selection sort the behavior is n^2 which correlates with the best average and worst case in the given table. For insertion sort shows as n and n^2 behavior which also goes with the given table. Merge sort is nln(n) across which is the same as the given table. For quick sort it follows n^2 behavior except for the random data values which were a bit unclear what behavior it was following, it was between n^2 and nln(n). The unusual one was insertion sort but that makes sense, the program ran the max number of n in less than 5 minutes for a sorted group. The algorithm itself works so that this would not take all that long in comparison to some of the others like merge sort which would need to break down for every element before grouping again. The expected matched closely with what was calculated. After being told over and over again what the time complexities are this project helped demonstrate first hand the time change, giving me a more grounded idea of the different complexities and how they are extremely important to time saving.

	t_{max}/t_{min}	n ratio	nln(n) ratio	n^2 ratio	Behavior
SC	2975.64	55	78.92999	3025	n^2
SS	2764.346	55	78.92999	3025	n^2
SR	2733.058	55	78.92999	3025	n^2
IC	105.661	100	128.5714	10000	n
IS	108.9016	100	128.5714	10000	n
IR	5470.25	200	315.0515	40000	n^2
MC	2912.643	2000	3158.465	4000000	nln(n)

MS	3146	2000	3158.465	4000000	nln(n)
MR	2529.556	2000	3158.465	4000000	nln(n)
QC	3237.649	55	78.92999	3025	n^2
QS	6819.351	150	207.2759	22500	n^2
QR	3553.693	260	370.1766	67600	nln(n)