Insertion Sort			
	Input of Size 10	Input of Size 100	Input of Size 1000
	6400	123400	4004500
	7400	140300	4983200
	4400	127500	4327200
	7100	151500	3997100
	6700	150600	4162000
	7500	132400	4078400
	5100	153700	4311600
	7000	149700	4067100
Average:	6,450	141,137	4,241,387

Merge Sort			
	Input of Size 10	Input of Size 100	Input of Size 1000
	17000	144400	631400
	13100	127200	789200
	16100	150200	704300
	16900	115300	1014900
	16500	120800	971200
	12900	132700	1248600
	17000	114700	694400
	16700	121800	671500
Average:	15,775	128,387	840,687

Analysis of Insertion Sort and Merge Sort:

The order of growth of running time of Insertion Sort Algorithm on average case is theta of n^2
Theory: when input grows 10 times, the running time increases by 100 times
Experiment: ignoring lower order terms, 6,450 is the same as 1,000 and 141,137 is the same as 100,000. This shows that as the input increases 10 times, the running time increases by 100 times
Experiment: ignoring lower order terms, 6,450 is the same as 1,000 and 4,241,387 is the same as 1,000,000. This shows that as the input increases 100 times (from 10 to 1000), the running time increases by 10,000 times. This proves that the order of growth for average case insertion sort is theta of n^2.

The Merge Sort is faster than Insertion Sort when the array is of size 1000 and above. However, when the size is 100 and above, the 2 algorithms have roughly the same running times, but the insertion sort is still a little faster on average as seen from the data above.
The order of growth of running time of Merge Sort Algorithm on average case is theta of nlogn
Theory: when input grows 10 times, the running time increases by 30 (10log 2 base 10) times
Experiment: ignoring lower order terms, 15,775 is the same as 10,000 and 128,387 is the same as 100,000. This shows that as the input increases 10 times, the running time increases by 30 times because 15,775 x 30 = 473,250; 471,250 is the same as 100,000 ignoring the lower order terms, which is the same as 128,387. This proves that the order of growth for average case merge sort is theta of nlogn.
In conclusion, the Insertion Sort is faster than Merge Sort when the array size is 10 because its lower order terms are less in that case. However, as the input size increases, we can see that the Merge Sort's (lgn) grows much slower than Insertion Sort's extra (n).