CS 200 HW - 0 Priyanshu Sethi 12310528

Problem Size	**Insertion Sort**	**Merge Sort**	**std::sort**
10	0	0	0.001
100	0	0	0
1,000	0.002	0	0
10,000	0.171215	0.001007	0.000999
100,000	16.7284	0.017513	0.015991
1,000,000		0.182375	0.17409
10,000,000		2.03147	1.87112
100,000,000		23.5317	21.6409

a) We can observe that the time it takes to sort the problem increases as we increase the problem size. There is a direct correlation between problem size and the time it takes. This is in line with what I expected since it normally takes more time to solve a bigger problem.

b) Overall Merge Sort seems to be a faster algorithm with a better time complexity since it only took about 2 seconds to sort 10 million values whereas insertion sort took over 16 seconds just to sort 100,000 values. Looking at the performance for small values, the difference is negligible and both algorithms take almost no time to execute. Merge sort uses the divide and conquer method and hence is able to execute in a time complexity of O(n log n) whereas insertion sort performs linear operations and takes O (n^2) to execute.

c) Out of all three, std::sort performed the best by taking the least amount of time to execute all problem sizes. It is significantly faster than insertion sort and has a slightly better performance than merge sort.