Introduction:

For PA5 different sorting methods are being investigated. The main objective is to gain an understanding of the runtimes of different sorting methods. Speed is greatly influenced by the total number of comparisons so keeping that low should be one of the main goals.

Theoretical Analysis:

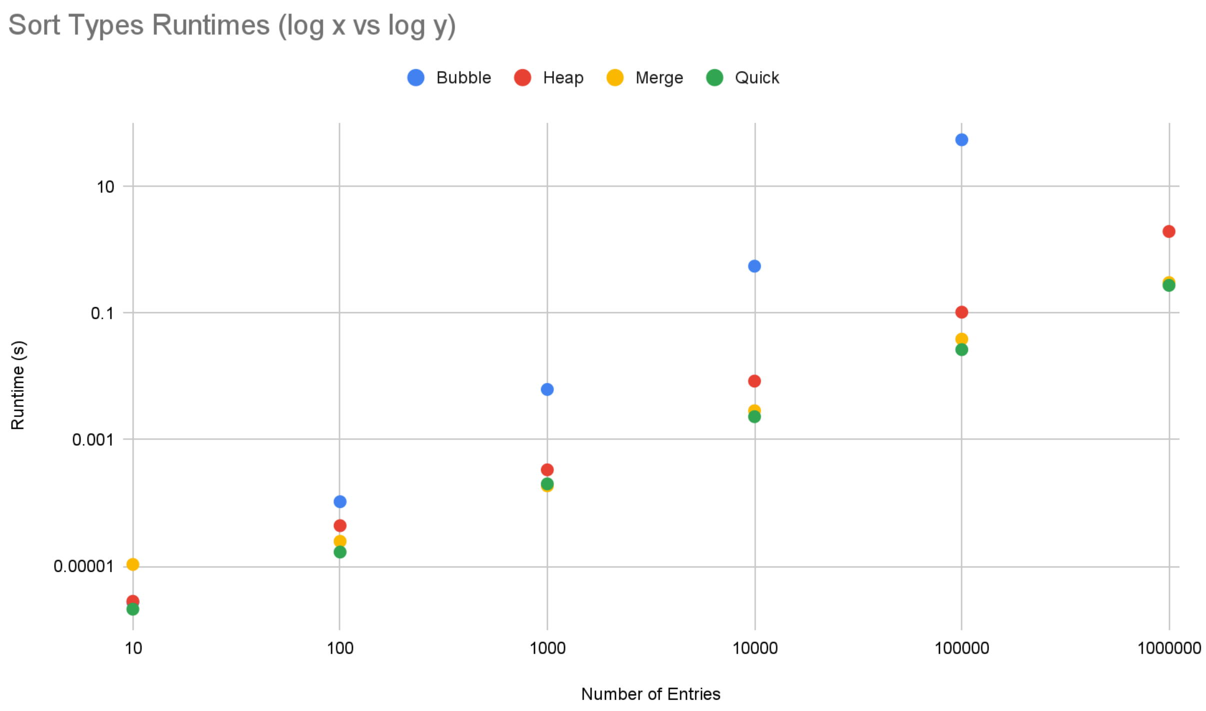
Right away bubble sort is going to be the worst as little consideration is put into decreasing the number of comparisons, thus the run time is going to be O(n^2) which is caused by the possibility of every element needing to be compared to every other. The rest of the three take different approaches to decreasing the number of comparisons. This will mean that each is roughly O(n log(n)) as each element still needs to be passed over however they don't need to be compared to every single element.

Experimental Setup:

For this experiment, a single pass was taken to organize the same list with a different number of input elements. So 100,000 random elements between 1 and 10^6 are added to a vector that will be preserved. Then these elements are copied into an int array that is sorted with the different algorithms. The array is then reset to the vector values and this is repeated at 10, 100, …, up to 1,000,000 elements. The exemption of this is bubble sort where it wasn't run at 1,000,000 as the O(n^2) runtime would result in a runtime upwards of 10 minutes.

Experimental Results:

Below is the included figure and table of the runtimes of each algorithm. It was confirmed as expected that in every number of entries, the bubble sort performs the worst with the exemption of 10 whereas the merge is slightly worse due to the upfront cost of the more complex algorithm. After this point, it pulls away from the pack quite quickly. For the rest, each of the 3 are within the O(n log(n)) runtime with merge and quick being more or less the same within the margin of error. Heap pulls consistently slightly slower with the gap growing as the number of entries increases. This was unexpected as it would have been expected to stay more with the runtime of the other algorithms. However, due to the provided in the PA5 documentation runtime deviating from merge and quick sort to the same degree this is seen as correct and not an error



	Bubble	Heap	Merge	Quick
10	0.0000027	0.0000028	0.0000107	0.0000021
100	0.000105	0.0000439	0.0000248	0.0000168
1000	0.006251	0.000335	0.000187	0.000201
10000	0.555398	0.008468	0.002878	0.00232
100000	55.2785	0.104085	0.039019	0.026631
1000000		1.96462	0.305526	0.27644