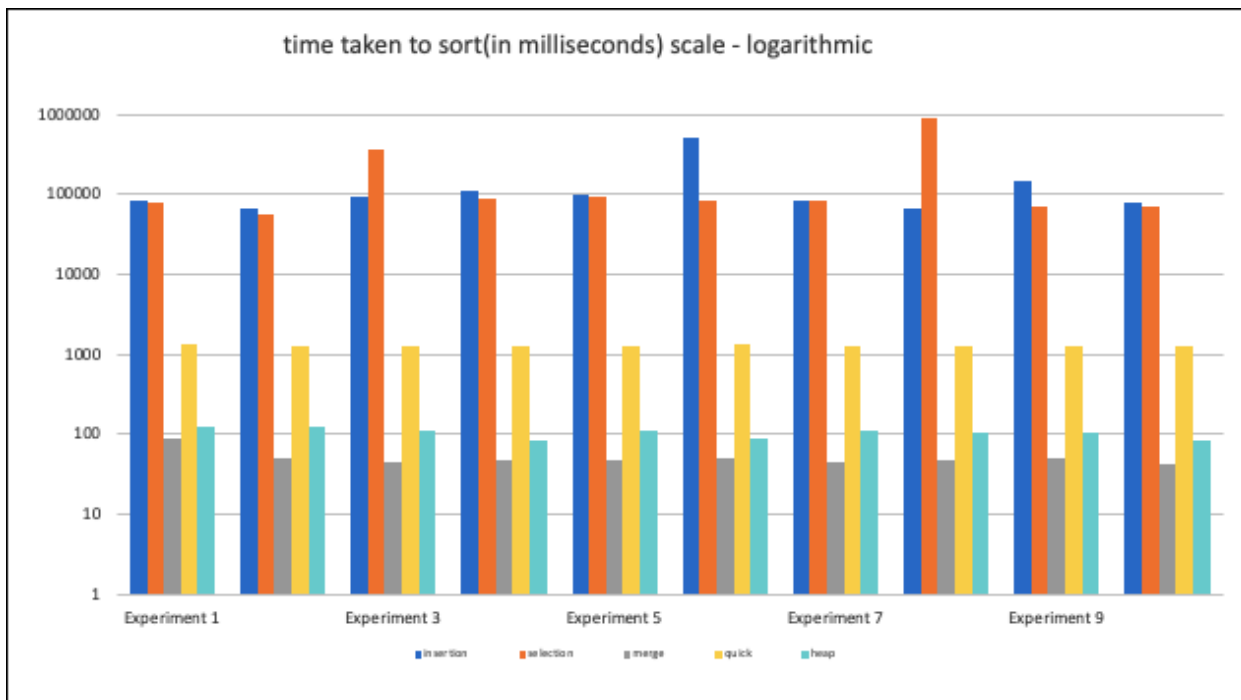


RUN TIME PERFORMANCE ANALYSIS OF SORTING ALGORITHMS

The following are some observations drawn from the runtime experiments conducted on sorting performance of various algos.

- Insertion and Selection sort take the most time to sort n words. The runtime complexity in the worst case for both these algorithms is $O(n^2)$. Insertion sort on average took 133562.1 milliseconds and selection sort took 185929.3 milliseconds on average.
- Merge sort takes the least amount of time, since it follows a divide and conquer paradigm, the runtime for merge sort is $O(n \log n)$. The only disadvantage with merge sort is that it does not sort the input in place and thus takes $O(n)$ memory as well. Merge sort took 51.1 milliseconds on average to sort.
- Quicksort takes more time than merge sort but it sorts in place. The runtime for quicksort is $O(n \log n)$ on average, it takes $O(n^2)$ in the worst case, but this behavior is rare. Quicksort took 1268.8 milliseconds to sort.
- HeapSort performs better than quicksort but not as fast as merge sort. Heap sort also takes $O(n \log n)$ time to sort and sorts in place. Heap sort took 103.8 milliseconds to sort on average



The following numerical runtimes(milliseconds) were observed across 10 experiments

Algo	Average	Max	Min	Std Dev
insertion	133562.1	515394	65580	160752.87
selection	185929.3	870844	54850	257454.184
merge	51.1	89	42	13.5355499
quick	1268.8	1298	1231	27.2102922
heap	103.4	125	83	14.0886085