Assignment 4

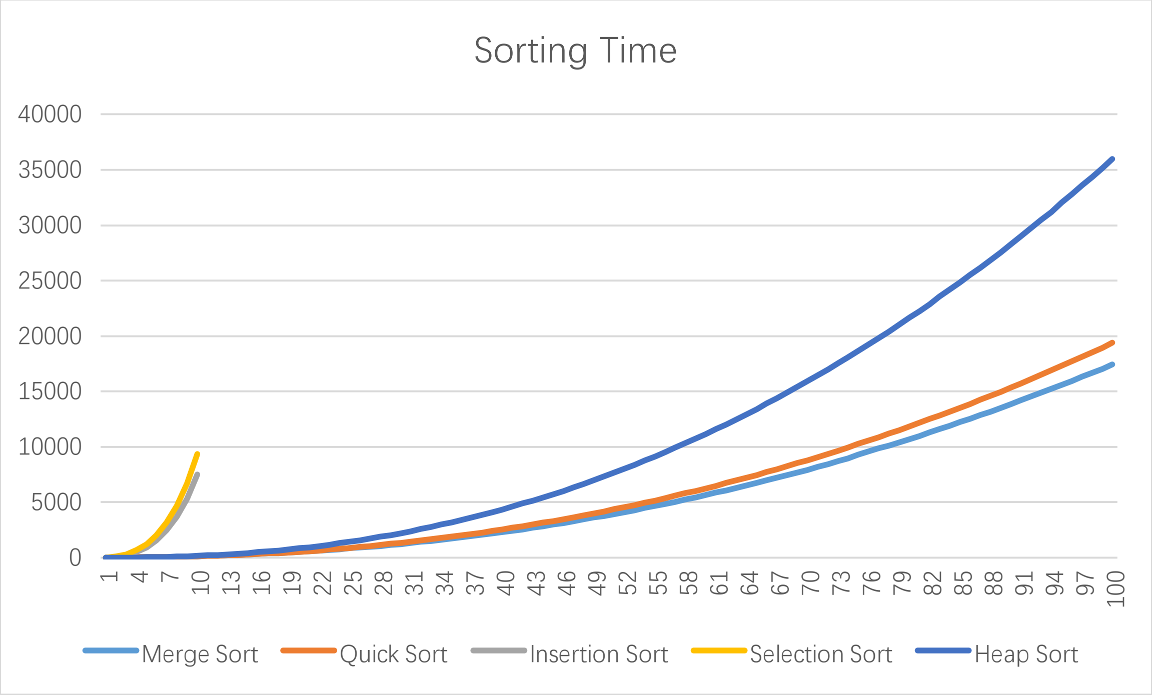
Zihan Zhang

001280965

Results and Conclusions:

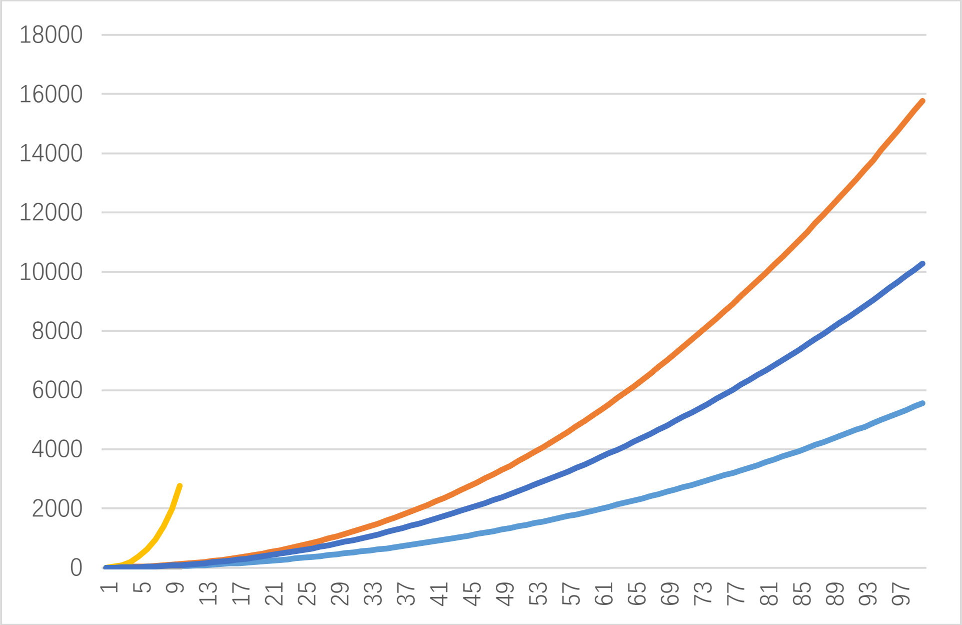
Source data are in the xlsx files in the package. Here is their graph.

Random:



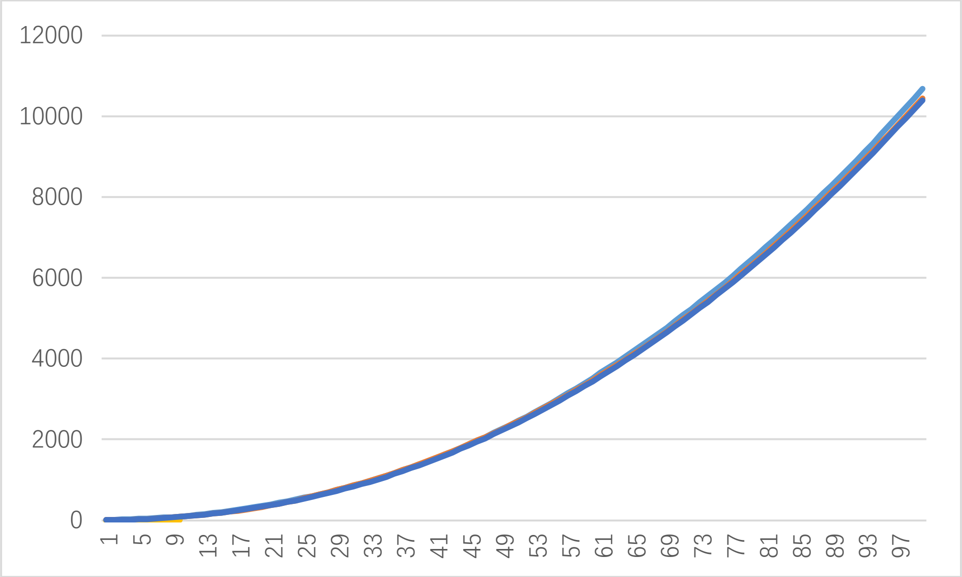
From the graph we can conclude that in terms of large number of random data, insertion and selection sort have extremely poor behavior. Merge sort behaves best which is due to the use of extra memory. Quick Sort is the second one which is a little slower than merge sort. And the Heap Sort is much slower than them.

Sorted:



From the graph, we can know that in terms of sorted arrays, Insertion Sort(which is always 0 so not shown on graph) is the champion which cost nearly no cost. Selection Sort is still the worst one. Merge Sort is the second one. Then heap sort and then quick sort. Heap sort wins quick sort because it reduce the cost of heaping the data.

Partially Sorted:



In terms of partially sorted data, all of them behave similarly. Maybe because I split them up in half. Half sorted and half unsorted which make them behave more evenly.

Conclusion:

If the data is highly random and the number is very large, besides there is no limit on the use of extra memory, then merge sort is definetly the NO. 1 choice. Like handling big-data problems and other no-limited-memory situation

Otherwise choosing quick sort or heap sort depends on how random the data is, more random, quick sort behaves better.

If the data is highly sorted, then insertion sort will be a very good tool. Like finding single bad data in a big orderly arranged data, picking up extreme data maybe.

Selection sort is not recommended in my research because there are other better choice.