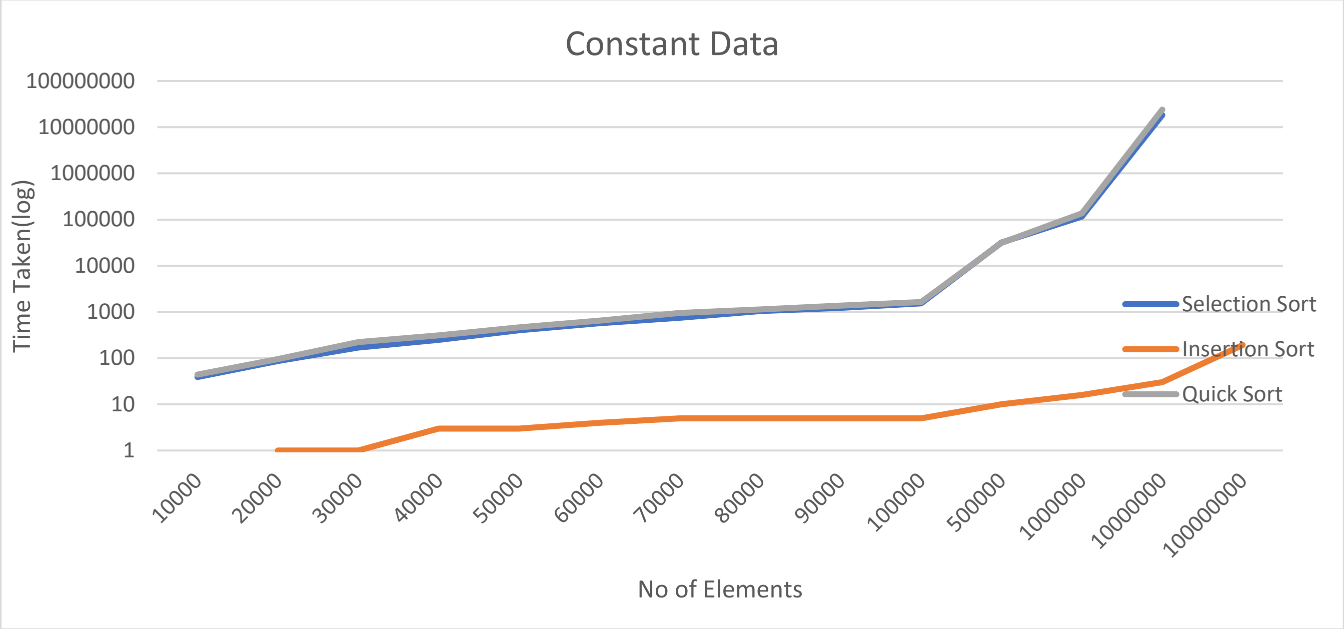
Report

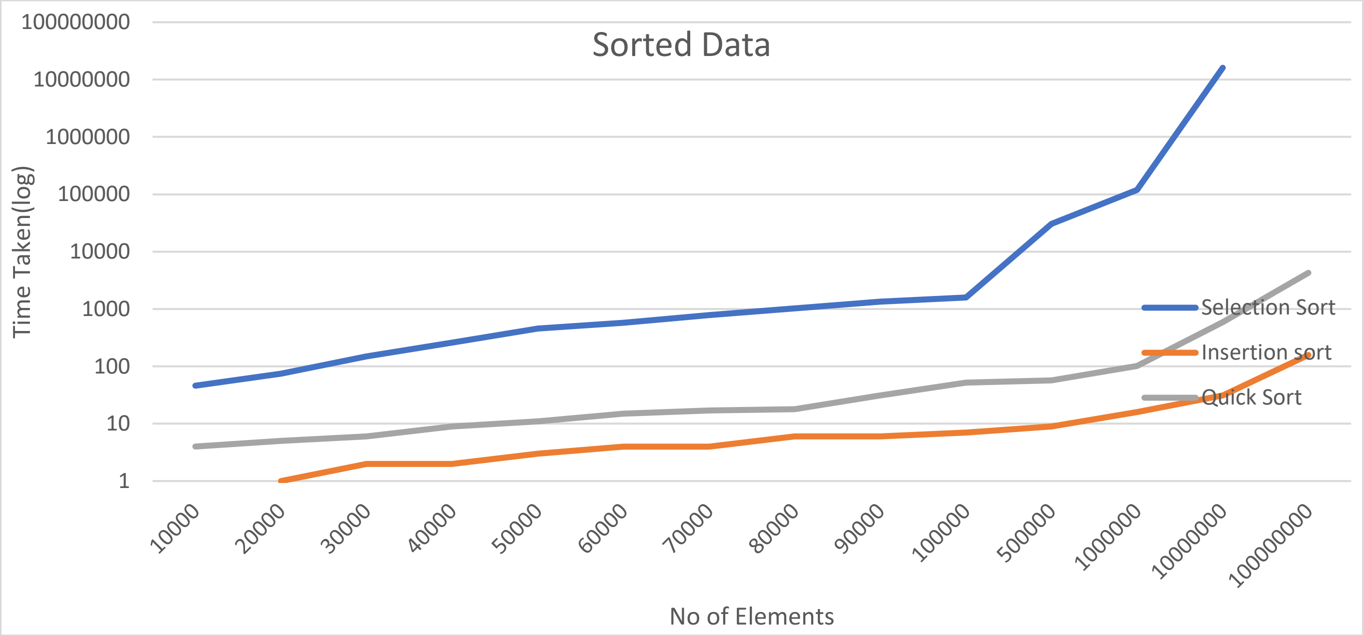
Name: Sai Tharun Bodduluru

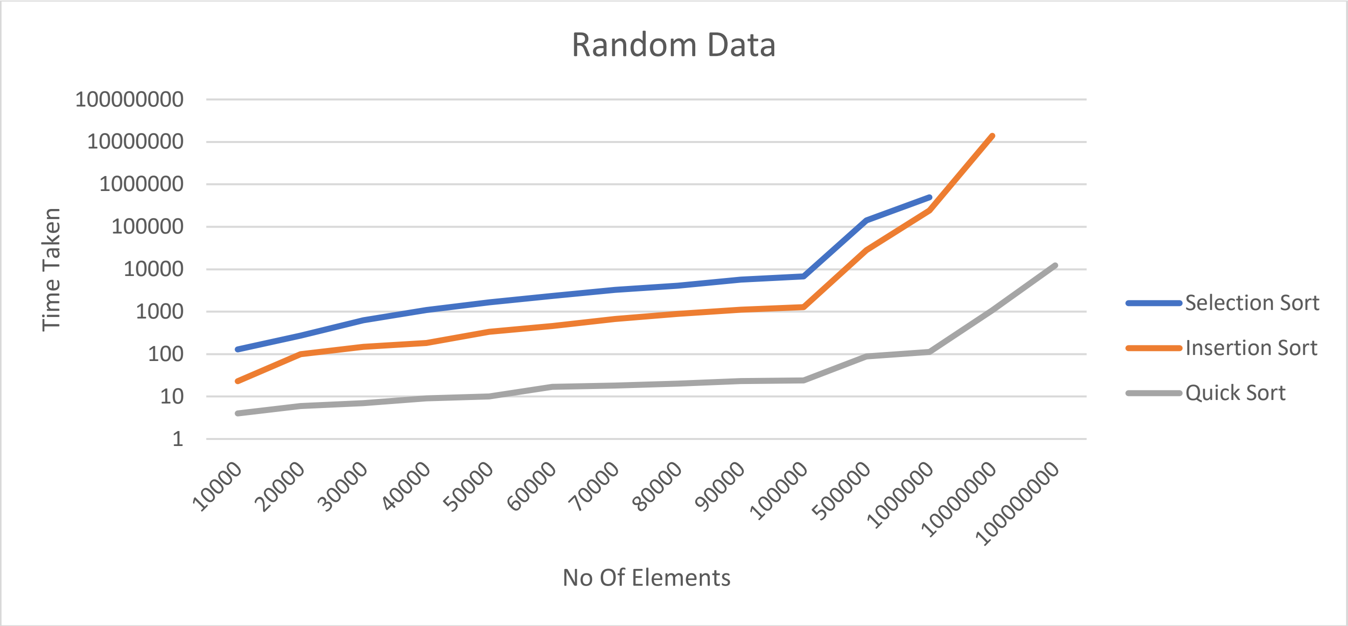
1) For the data <= 1,000,000
   If the data is sorted and constant insertion sort performed the best when compared to other sorting algorithms.
   If the data is random then quick sort performed the best
2) For the data >1,000,000 and <1,000,000,000
   For the random data quick sort is the best sorting algorithm and for sorted and constant insertion sort is best
3) No, I was not able to run any algorithm with elements size of 1,000,000,000. I have developed these sorts in java, which uses heap and stack as part of java architecture to run the programs. So, for 100,000,000 elements the generally allocated heap size will not be sufficient and we will need to increase the heap size by using flag -Xmx4g. This flag will increase the heap size to 4GB and it works fine. But for 1,000,000,000 elements 4GB flag size is also not sufficient and it requires at least 8GB. Where my system is having only ram of 8GB. When I tried to increase the heap size to 8GB system was not responding and hung.
   The error log was as below
   ```
   #There is insufficient memory for the Java Runtime Environment to continue.
   # Native memory allocation (mmap) failed to map 1610612736 bytes for G1 virtual space
   # Possible reasons:
   #   The system is out of physical RAM or swap space
   ```
4) Yes, the results were as expected. I have noticed that selection sort has the worst performance compared to other sorts. I have seen the other sorts performed good on one or the other type data, like insertion sort was good at sorted data and constant data and quick sort was good at sorted and random data but selection sort was worst in all types of data. As we have studied selection has time complexity of n^2 all the times whether it is worst case or best case. Where as insertion sort has best time complexity of $\Omega(n)$ and worst-case complexity of $O(n^2)$. We have found that insertion sort worked with time complexity of n with sorted and constant data. Where as in quick sort Constant data time complexity seems to be like worst case like $O(n^2)$ and the sorted and random was good.

N/A – For data more than 10000000 it took more than 12hrs but sort is not completed.
For data of size 1000000000 my system is not compatible and the error was explained in answer 3
For selection sort with data 10000000 it took more than 12hrs but sort is not completed.

5)


Constant Data
Time Taken(log)
100000000
10000000
1000000
100000
10000
1000
100
10
1
10000
20000
30000
40000
50000
60000
70000
80000
90000
100000
500000
1000000
10000000
100000000
No of Elements
Selection Sort
Insertion Sort
Quick Sort


Sorted Data
Time Taken(log)
100000000
10000000
1000000
100000
10000
1000
100
10
1
10000
20000
30000
40000
50000
60000
70000
80000
90000
100000
500000
1000000
10000000
100000000
No of Elements
Selection Sort
Insertion sort
Quick Sort


Random Data
Time Taken
100000000
10000000
1000000
100000
10000
1000
100
10
1
10000
20000
30000
40000
50000
60000
70000
80000
90000
100000
500000
1000000
10000000
100000000
No Of Elements
Selection Sort
Insertion Sort
Quick Sort