Design & Analysis of Algorithms

Monsoon Semester III 2020-21

Lab - 1

**Topics: To Compute the performance of sorting algorithms**

# AIM

To evaluate the performance of sorting algorithm. Performance of bubble Sort, insertion sort, selection sort, merge sort & quick sort should be carried out.

# EXPERIMENT

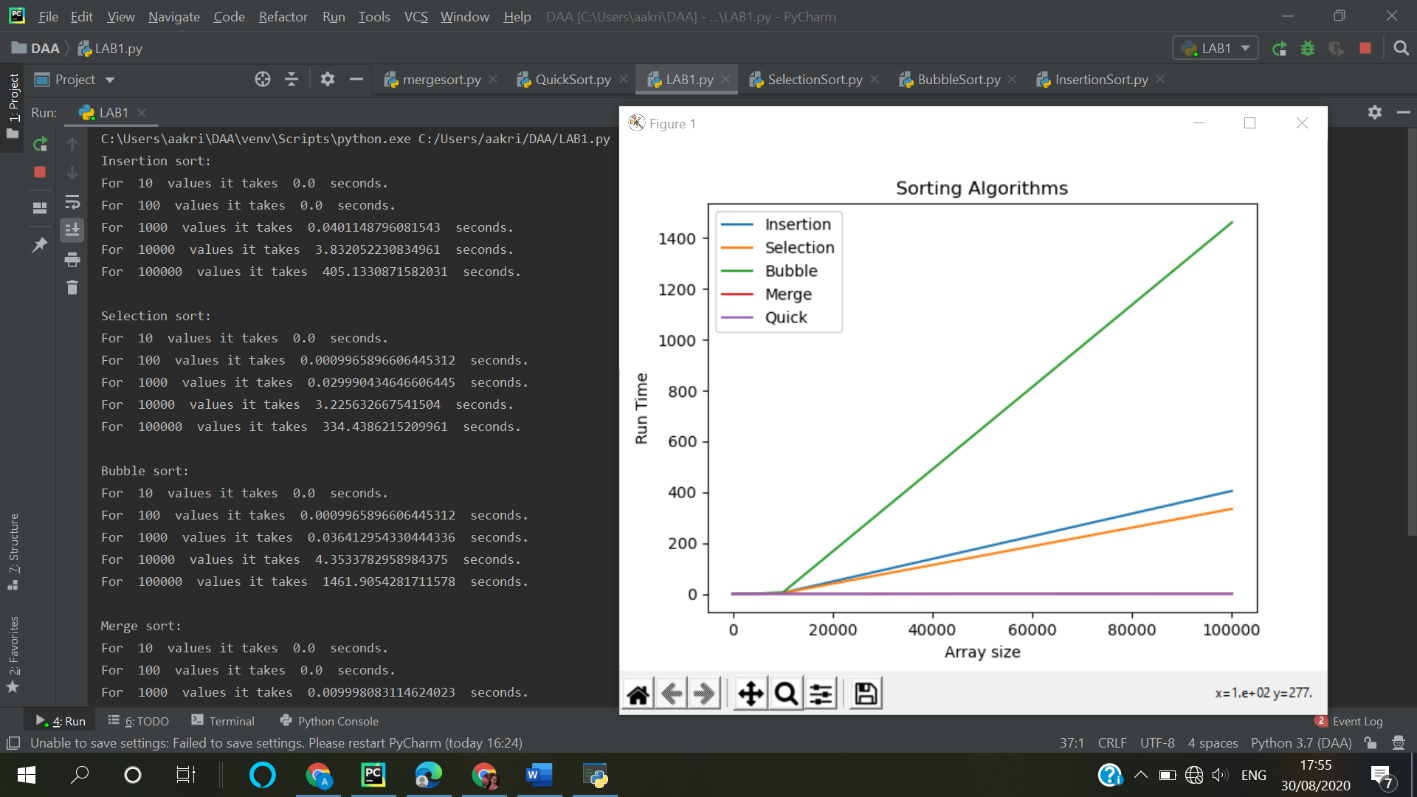
1. Insertion sort, selection sort bubble sort, merge sort and quick sort were implemented. The elements were generated randomly within the code.

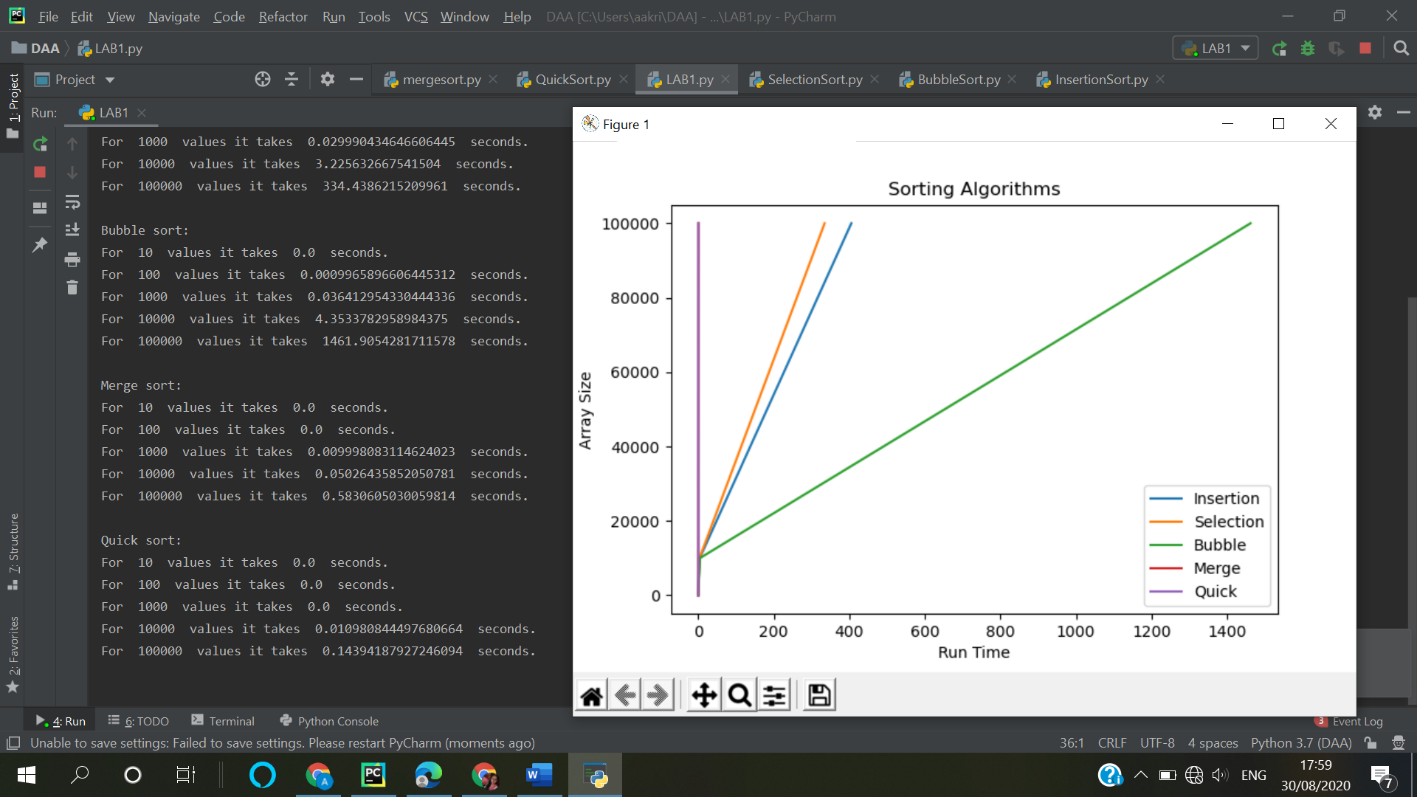
1. The performance of program by varying the number of elements was noted.
2. The time taken by each case (for particular number of inputs) n = 10, 100, 1000, 10000, 1000000 was computed.
3. A graph with number of inputs Vs time taken in seconds was plotted
4. The graphical plots for each sorting algorithms with its theoretical time complexity was compared

e. Also the time taken for sorted array (worst case) was computed and compared with different number of elements.

# OBSERVATIONS

The output received on running the program is shown in the below figure:





The theoretical values for the runtime of the algorithms are:

| **Algorithm** | **Time Complexity** | | |
| --- | --- | --- | --- |
|  | **Best** | **Average** | **Worst** |
| [Selection Sort](http://geeksquiz.com/selection-sort/) | Ω(n^2) | θ(n^2) | O(n^2) |
| [Bubble Sort](http://geeksquiz.com/bubble-sort/) | Ω(n) | θ(n^2) | O(n^2) |
| [Insertion Sort](http://geeksquiz.com/insertion-sort/) | Ω(n) | θ(n^2) | O(n^2) |
| [Quick Sort](http://geeksquiz.com/quick-sort/) | Ω(n log(n)) | θ(n log(n)) | O(n^2) |
| [Merge Sort](http://geeksquiz.com/merge-sort/) | Ω(n log(n)) | θ(n log(n)) | O(n log(n)) |

On comparing we see that, Quick sort and Merge sort the most efficient sorting algoritms.

# CONCLUSION

Hence we have successfully evaluated the performance of bubble Sort, insertion sort, selection sort, merge sort & quick sort algorithms.