

Bubble Sort :

Bubble sort is a sorting algorithm, It works by comparing each pair of adjacent elements and switching their positions if necessary. It repeats this process until all the elements are sorted.

The average and worst-case time complexity of bubble sort is – $O(n^2)$

Merge Sort:

Merge Sort is a **Divide and Conquer** algorithm. It divides input array in two halves, calls itself for the two halves and then merges the two sorted halves

Time complexity of Merge Sort is $O(n \log n)$ in all 3 cases (worst, average and best) as merge sort always divides the array into two halves and take linear time to merge two halves.

Insertion Sort :

Insertion sort is a simple sorting algorithm that builds the final sorted array (or list) one item at a time. It is much less efficient on large lists than more advanced algorithms such as quicksort, heapsort, or merge sort.

Time complexity of Insertion Sort in Average and Worst case is $O(n^2)$ and best case being $O(n)$

Heap Sort:

Heap sort is a comparison based sorting technique based on Binary Heap data structure. It is similar to selection sort where we first find the maximum element and place the maximum element at the end. We repeat the same process for remaining element.

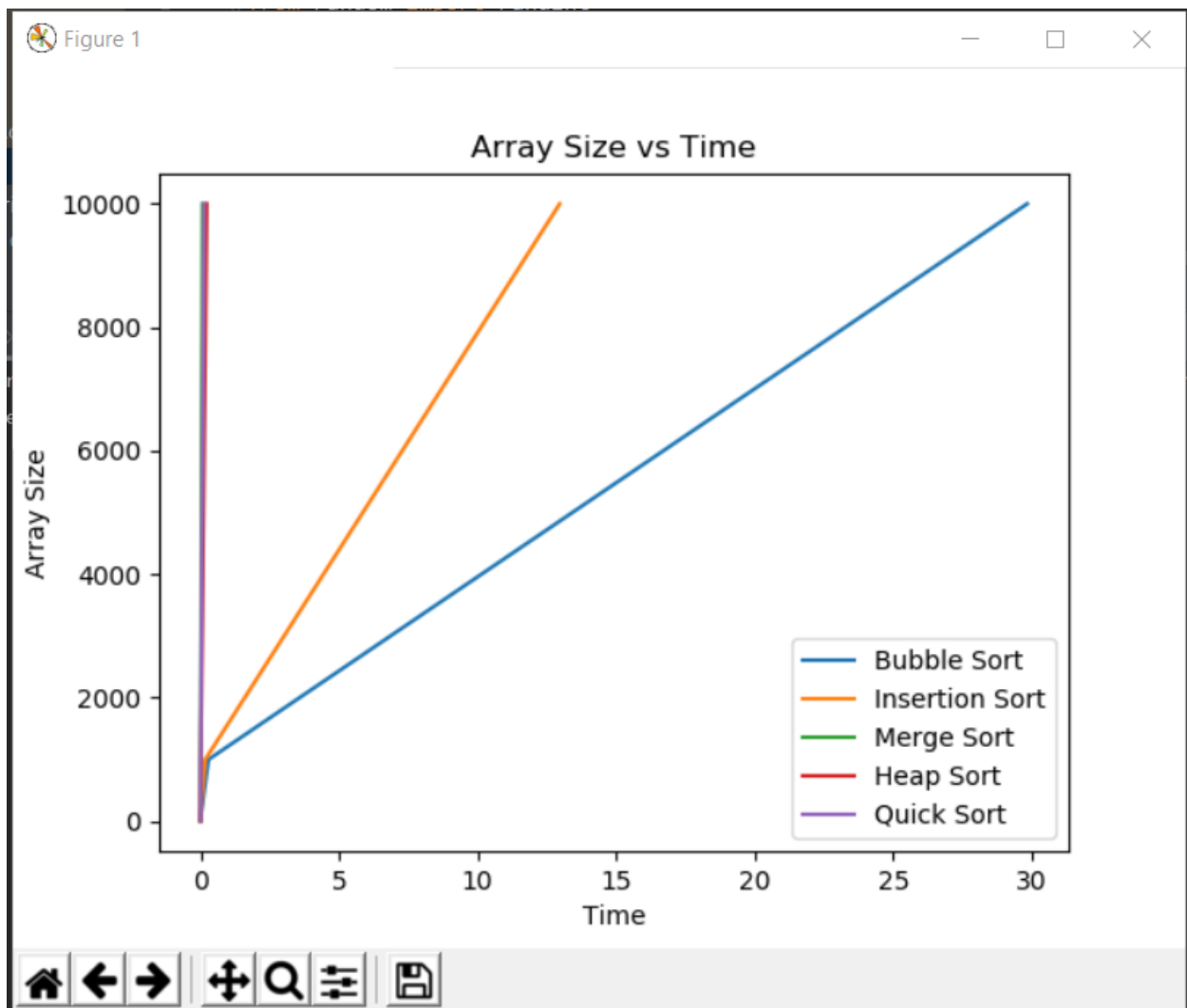
Time complexity of Heap Sort in Best, Average and Worst case is $O(n \log n)$

Quick Sort :

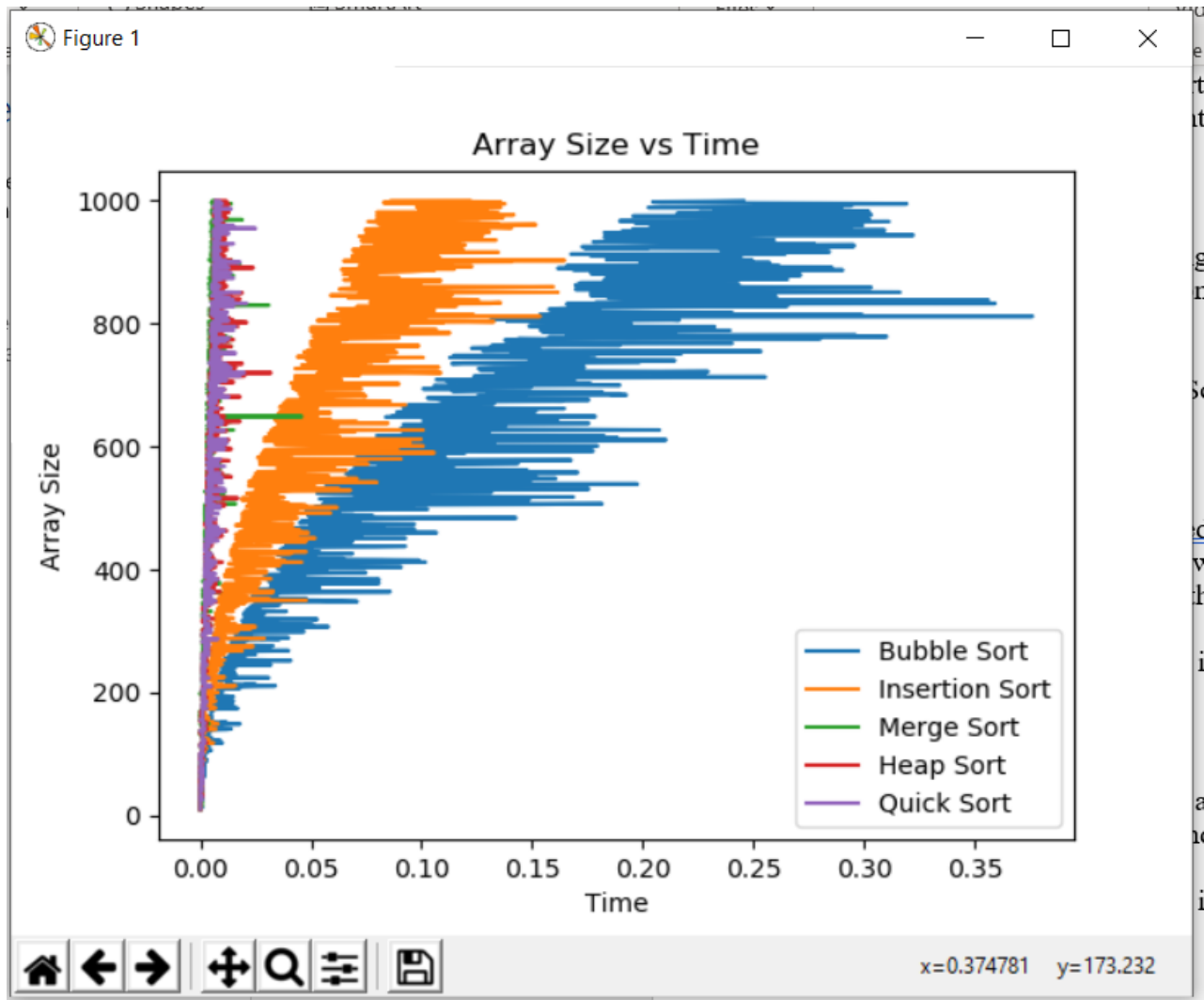
Like Merge Sort, QuickSort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot.

Time complexity of Heap Sort in Best and Average case is $O(n \log n)$ and Worst case is $O(n^2)$

The graph showing the run time comparison between Bubble sort, Quick Sort, Insertion Sort, Merge Sort and Heap Sort



The graph showing the run time comparison between Bubble sort, Quick Sort, Insertion Sort, Merge Sort and Heap Sort for larger input



Data Structures used in the Project Implementation :

The data structures used for implementation of the algorithms in this project is Array. The Python standard library provides a module called random that offers a suite of functions for generating random numbers. A randomized array has been used in order to generate random numbers for sorting purpose.