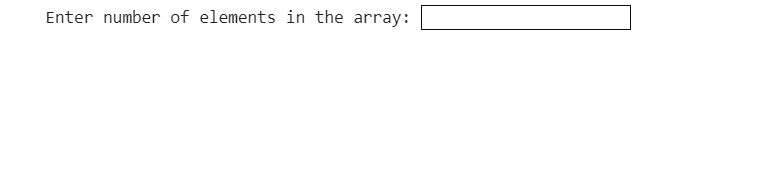
DAA Programming Project

Sorting Algorithms

**Name: Shreyas Jagadeep Shete UTA-ID: 1001888859**

# Sort Algorithm Test.

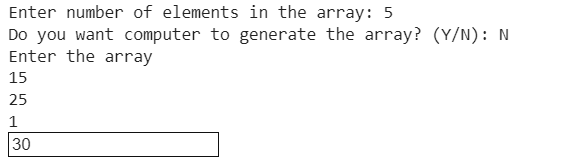
1. On running Sort Algorithm, we need to enter the number of elements (int value) in the array we want the program to run on,



1. Choose if the computer should generate the array randomly (enter “Y”) or if user wants to enter it (enter “N”),



1. If you want user to enter, following screen appears wherein, you enter a value and hit “enter” key to enter next value and so-on,



1. Irrespective of the array generated by computer or entered by user, sorted algorithm is displayed and the taken for each run is displayed as shown below,







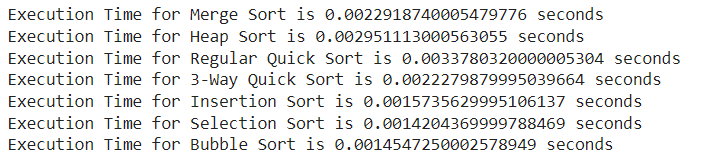




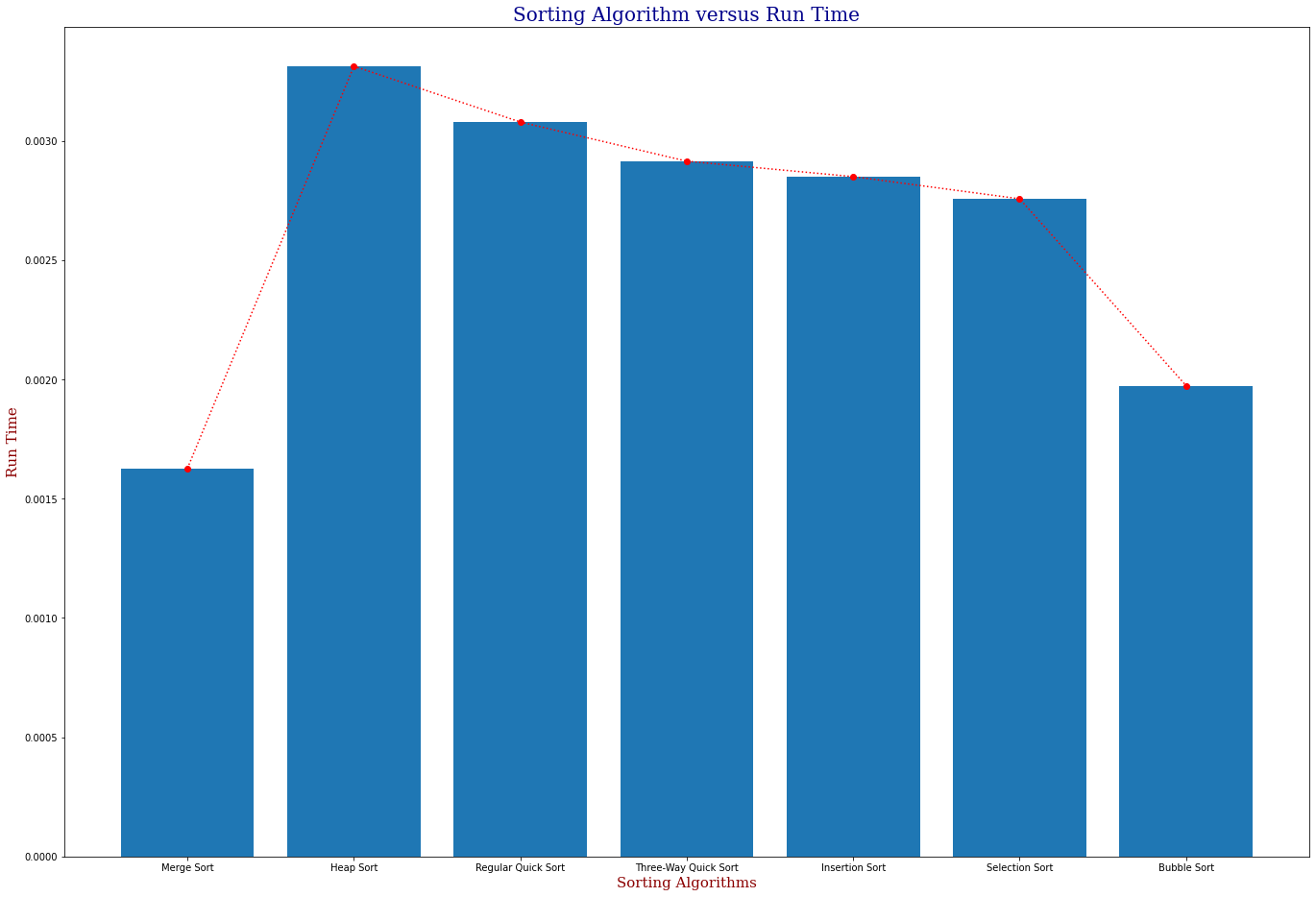








1. Graph is generated for the run where-in X-axis represents different sorting algorithm and Y-axis represents the time



**Project and Algorithm Descriptions.**

Data Structures used in project.

1. Array
2. Merge Sort Algorithm.

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

Best Case Time Complexity: O(n\*log n)

Worst Case Time Complexity: O(n\*log n)

Data Structures:

* 1. Array

1. Heap Sort Algorithm

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 minimum element and place the minimum element at the beginning. We repeat the same process for the remaining elements.

Best Case Time Complexity: O(n\*log n)

Worst Case Time Complexity: O(n\*log n)

Data Structures:

* 1. Array

1. Quick Sort Algorithm

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..

Best Case Time Complexity: O(n\*log n)

Worst Case Time Complexity: O(n^2)

Data Structures:

* 1. Array

1. Insertion Sort Algorithm

Insertion sort is a simple sorting algorithm that works similar to the way you sort playing cards in your hands. The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct position in the sorted part.

Best Case Time Complexity: O(n)

Worst Case Time Complexity: O(n^2)

Data Structures:

* 1. Array

1. Selection Sort Algorithm

The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. The algorithm maintains two subarrays in a given array.

1) The subarray which is already sorted.

2) Remaining subarray which is unsorted.

In every iteration of selection sort, the minimum element (considering ascending order) from the unsorted subarray is picked and moved to the sorted subarray.

Best Case Time Complexity: Ω(N2)

Worst Case Time Complexity: O(1)

Data Structures:

* 1. Array

1. Bubble Sort Algorithm

Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in the wrong order.

Best Case Time Complexity: O(n^2)

Worst Case Time Complexity: O(n^2)

Data Structures:

* 1. Array

Conclusion:

1. No algorithm is better than other algorithms. It can be only compared on input values provided.
2. The comparison of all sort algorithms mentioned above depend on two criteria,
   1. Size of data set.
   2. Number in data set to be sorted.

**References for this document:**

* https://www.geeksforgeeks.org/
* https://colab.research.google.com/drive/1AHfrZbqJNSWWihN\_Z60IRriljEQvWs3R#scrollTo=KJ8YjthzKm6z