University of Texas Arlington

Department of Computer Science Engineering

CSE 5311 Design & Analysis of Algorithms

Fall 2021

Project-1

Sorting Algorithms

Instructor: Dr.Negin Fraidouni

Done by:

Vijay Ganesh Panchapakesan (Section 002)

(1001861777)

Abstract:

A Sorting Algorithm is used to rearrange a given array or list elements according to a comparison operator on the elements. The comparison operator is used to decide the new order of element in the respective data structure. In Practice, Sorting plays a very important role as many applications uses the sorted data. This project is about different sorting algorithms and to show how these algorithms performs if the size of the inputs vary i.e I have tested all the algorithms with data of different sizes. I have also compared their performance and tabulated them. The user first enters an unsorted data and on a button click, the data would be sorted based on the algorithm that the user selects and the execution time of the data using the selected algorithm is displayed.

For this, I used HTML, CSS and JavaScript to fulfill the requirement.

Project 1:

This project contains 7 sorting algorithms. The sorting algorithms used in this project are:

Sorting Algorithms used:

1. Insertion Sort
2. Selection Sort
3. Bubble Sort
4. Merge Sort
5. Heap Sort
6. Quick Sort
7. 3 Median Quick Sort

Implementation:

Initially I thought of programming in Python, as there were many graphs involved. Then I thought of giving the project in the web so for that I needed to know flask but I am unaware of Flask. So I decided to code in HTML, CSS and JavaScript as my programming language. For this project, I was deeply into JavaScript and I took a template from the web, for which I have given credits at the footer of my web page, and I edited it according to my requirement.

Simple Readme:

I have coded the project in HTML, CSS, and JavaScript. The user must follow the steps to get the output.

1. Go to the following website https://daaproject1001861777.netlify.app/
2. Then if the user wants to sort the data, then the user needs to click the Sort button and it would navigate the user to the Soring section
3. The user must enter the unsorted array in the textarea given and the each number should be separated by a space. This can be done by two methods:
    a. User can copy the data from the test_data.txt and check the performance
    b. The user can enter their own space separated data and check the results
4. Then the user can choose the sorting algorithm that they want to be performed on the unsorted data from the dropdown list provided
5. Once the sorting algorithm is selected, now the user can sort the unsorted data by clicking the “Sort” button and the sorted data would be displayed below in the textarea.
6. Along with the sorted data, the time taken for the execution in seconds is also displayed in the textbox given.

Programming Language:

1. For UI : HTML, CSS, Bootstrap, CanvasJS
2. For the Scripting: JavaScript

Data Structures mostly used were JSON and Array 95% of my code has array has data structure but only for CanvasJS I used JSON as my data structure to store the values

Sample Inputs:

Small Input Size:n=15

12 15 10 4 7 8 0 3 5 14 1 2 6 11 13

Average Input Size:n=40

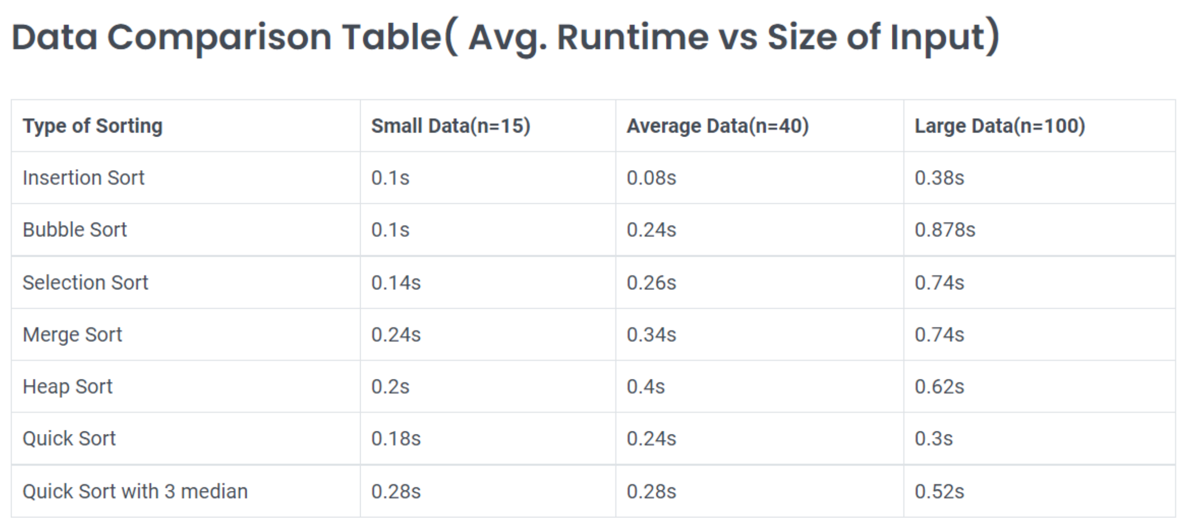
39 22 6 31 32 7 3 19 1 35 40 8 30 4 15 18 21 11 5 34 25 28 29 12 16 36 2 38 13 33 27 26 24 37 20 10 9 14 23 17

Large Input Size n=100

33 28 75 36 16 69 80 23 56 25 99 60 57 8 68 48 71 70 9 17 78 54 93 97 20 91 29 76 5 59 42 26 64 89 22 81 44 85 98 2 95 37 35 19 62 39 58 72 27 82 88 38 31 90 83 32 40 13 1 79 46 55 87 73 49 63 77 3 92 6 21 7 12 43 84 45 18 47 66 96 14 4 65 86 51 61 24 15 30 94 10 41 67 100 53 34 11 50 52 74

These are the inputs that were used to test and give the performance of each sorting Algorithm and tabulate the same.

Data Comparison Table(Runtime vs Size of Input):

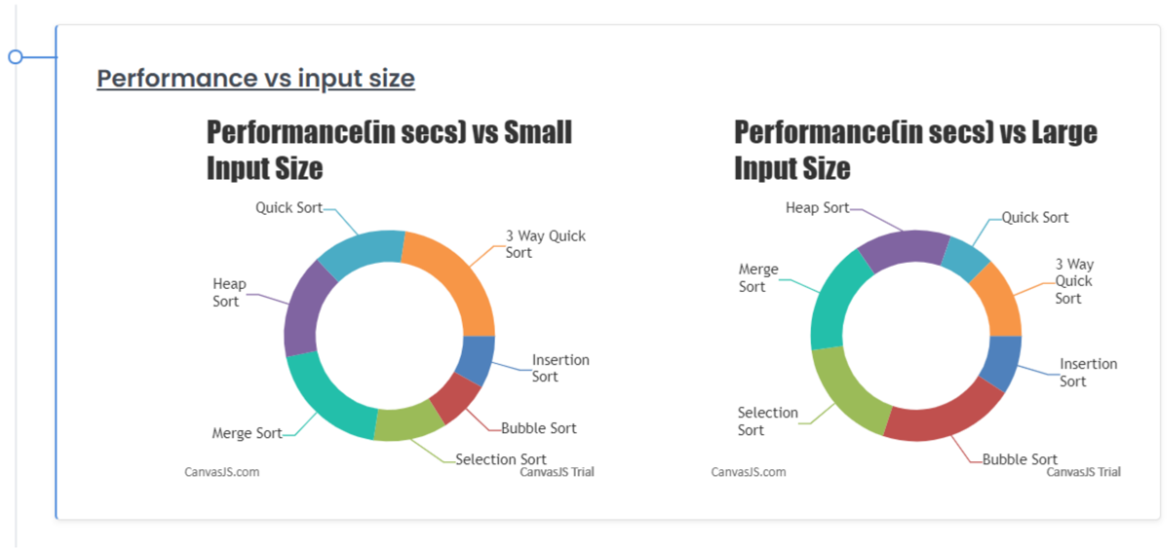


Data Comparison Table(Avg. Runtime vs Size of Input)

Type of Sorting	Small Data(n=15)	Average Data(n=40)	Large Data(n=100)
Insertion Sort	0.1s	0.08s	0.38s
Bubble Sort	0.1s	0.24s	0.878s
Selection Sort	0.14s	0.26s	0.74s
Merge Sort	0.24s	0.34s	0.74s
Heap Sort	0.2s	0.4s	0.62s
Quick Sort	0.18s	0.24s	0.3s
Quick Sort with 3 median	0.28s	0.28s	0.52s

The above table describes the performance of each sorting algorithm for different sizes of inputs.

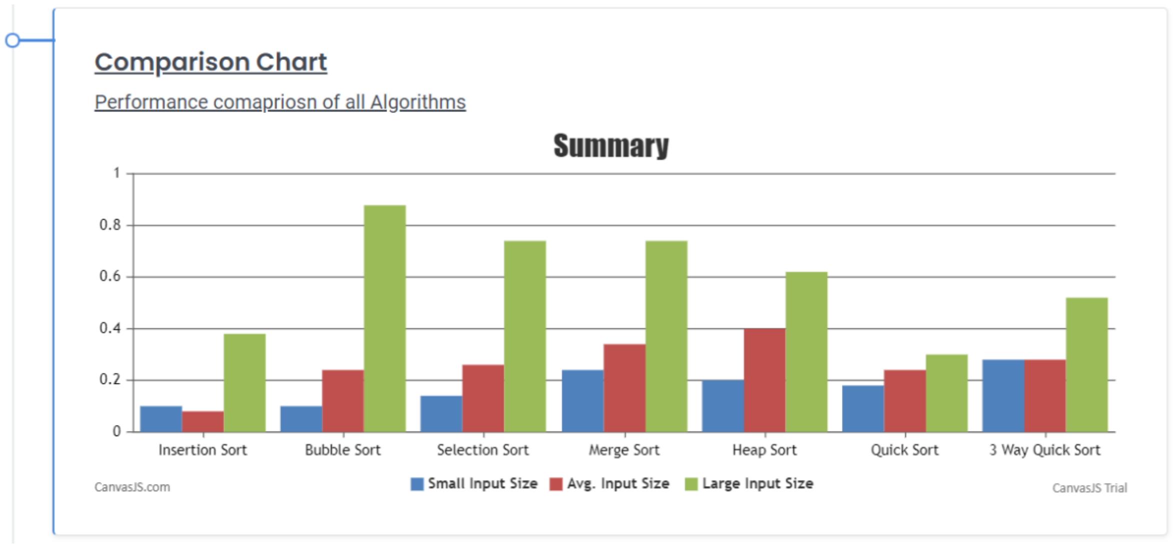
Data Visualization:



The Chart on the left describes how the performance for each sorting algorithm for small input size(n=15)

The Chart on the right describes how the performance for each sorting algorithm for small input size(n=100)

Data Comparison Chart:



The chart above describes how all algorithm performs on small input size, average input size and large input size

From the comparison graph what could be concluded is that Bubble sort performs the worst for large Input size and the Quick sort performs the best for large input size.

For the small Input size, Insertion sort performs the best and Merge Sort and 3 Median Quick Sort performs the worst.

For Average input size, Insertion sort performs better compared to other algorithms.

Acknowledgement:

I, Vijay Ganesh Panchapakesan(1001861777) from Section 002 have completely implemented the sorting algorithms and prepared this report.

I hereby thank Dr.Negin Fraidouni for the support and guidance for developing this application.