

ALGORITHMS PROJECT REPORT: -

ABSTRACT: -

An efficient and fastest sorting algorithm is necessary because in the field of computer science, they are a method of organizing data in certain order and can be used to organize messy data to be easier to use. Therefore, developing a strong understanding of sorting algorithms and how they work is a crucial fundamental of computer science.

INTRODUCTION: -

Keeping in mind the view and importance of algorithms we have developed a project consisting of sorting algorithms visualizations in a user-friendly interface. Below is the list of algorithms implemented:

- Insertion sort
- Bubble sort
- Merge sort
- Heap sort
- Quick sort
- Quick sort optimized
- Radix sort
- Bucket sort
- Counting sort

PROGRAMMING DESIGN: -

Our project implementation is based on React-JS /NodeJS programming language consisting of node modules. It provides a wide range of libraries to make and design beautiful websites and apps with real user interface with various features like play, pause speed control (0.25x to 4x) and complexities. It shows various colors with its color key for

better visualization so that it becomes very helpful to understand algorithms easily.

EXPERIMENTAL SETUP: -

We have developed a website for our project. The sorting process will be shown clearly and shown in the form of Bars/graphs, from lower to higher graph indicates numbers arranged in ascending order. There is a dropdown for selecting a particular sorting algorithm to be run on the given dataset ranging from 10 elements to 1 thousand and up to 1 million files. For setup, we need to download Nodejs, Reactjs additional libraries, and IDE.

RESULTS AND DISCUSSION: -





■ Sorted ■ Comparing ■ Swapping

Bubble Sort

Bubble Sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements and swaps them if they are in the wrong order. The pass through the list is repeated until the list is sorted. The algorithm, which is a comparison sort, is named for the way smaller or larger elements "bubble" to the top of the list. Although the algorithm is simple, it is too slow and impractical for most problems.

Performance Total Time Taken=28 sec

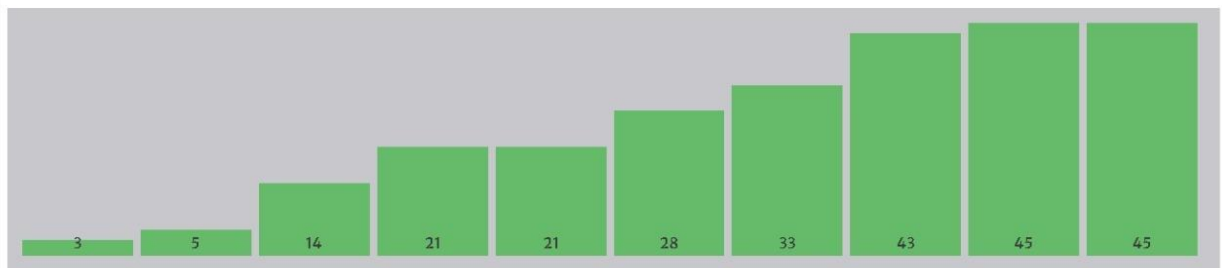
Worst-case time complexity: $O(n^2)$

Average time complexity: $O(n^2)$

Best-case time complexity: $O(n)$

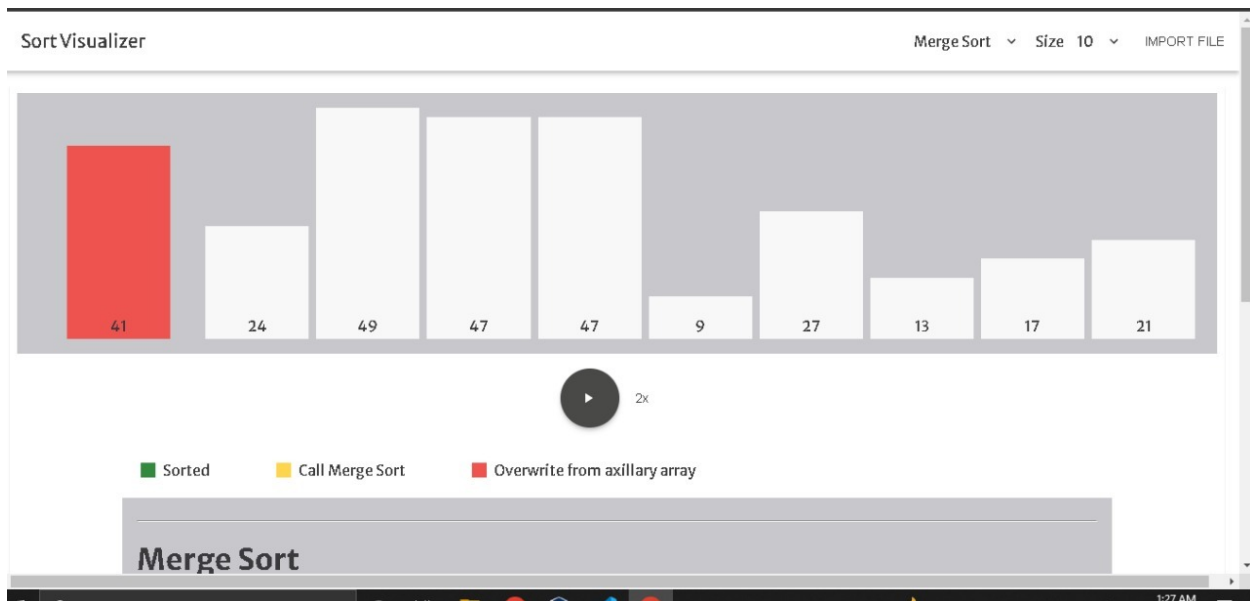
Space Complexity: $O(1)$

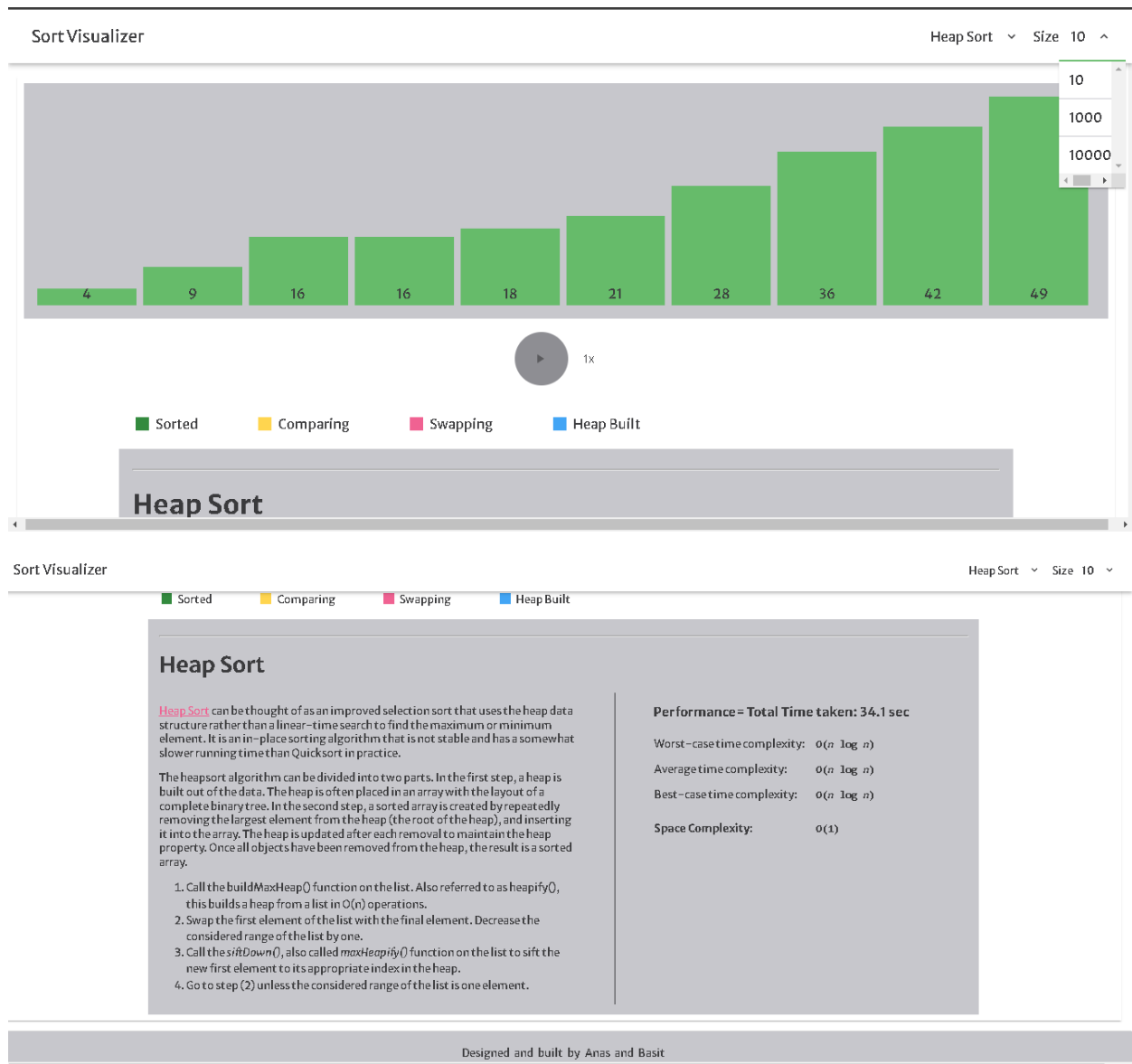
Designed and built by Anas and Basit



■ Sorted ■ Comparing ■ Swapping

Bubble Sort





These are some screenshots, with complexities of each algorithm. Some algorithms take time to sort, and some are fast. It depends on its complexity. Merge Sort ($n \log(n)$) are very much faster than Bubble sort(n^2) and so on. In above screenshot, heap sort on speed of 1x it takes 34.1 seconds to sort.

CONCLUSION: -

We have designed our best to implement the algorithms in a user-friendly interface so that everyone can understand algorithms easily.

REFERENCES: -

After a lot of research, we can design these algorithms. It took a lot of research in this project. Some references are

https://en.wikipedia.org/wiki/Insertion_sort

<https://www.geeksforgeeks.org/bucket-sort-2/>

<https://www.geeksforgeeks.org/radix-sort/>

<https://reactjs.org/docs/getting-started.html/>