SORTING VISUALIZER

A Minor Project Report
in partial fulfillment of the degree

Bachelor of Technology
in
Computer Science & Artificial Intelligence

By
2003A51015 A.Sathwika
2003A51004 D.Poojitha
2003A51054 J.Deekshitha

Under the Guidance of
Mr.Jagadish sir Asst. Prof, CS&AI Department

SCHOOL OF COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE
SR UNIVERSITY, ANANTHASAGAR, WARANGAL
April, 2023.

ABSTRACT

- Sorting Visualizer will be displaying the working mechanism of various sorting algorithms like, Bubble sort, Selection Sort, Insertion Sort, Quick Sort, Heap Sort and Merge Sort.
- The main objective of developing this Visualizer is to make a learner comfortable in learning these techniques quickly and easily.
- We know the sorting algorithms are the most widely used algorithms in many aplications including Discrete event simulation, Operating Systems, real time systems and many other as well.

PROBLEM STATEMENT

- Since I faced the problems of sorting during the course of algorithm design in the first year of my studies, there is an understanding that the visual representation is a vital part of the studying process.
- The main goal of the thesis was to create a program which would serve as a tool for understanding how most known sorting algorithms work. There was an attempt to make the best possible user experience.

OBJECTIVES

- The main Objective of Sorting Visualizer project is to learn and visualize the sorting algorithms how these algorithms works We can see the steps of swapping or any divide and conquer part of these bars. Here 5 different types if sorting algorithms is used to visualize.
- A visualization of data is implemented as a bar graph, after which a data sorting and algorithm may be applied.The resulting animation is then performed either automatically or by the user.

Literature Review

- Sorting visualizers are typically implemented as web-based or desktop applications. They usually generate a randomized array and then watch as the sorting algorithm sorts the data in real-time.
- This helps to gain a better understanding of the sorting process and how different sorting algorithms work
- Other studies have focused on making the visualizations more accurate and reliable, such as by incorporating error-correction techniques.

Literature Review

- Overall, sorting visualizers are a useful tool for computer scientists and data scientists
- They can help to better understand how different sorting algorithms work and can also be used to compare different sorting algorithms and their performance.
- Research into sorting visualizers is ongoing, with the focus on making them more user-friendly and efficient.

Proposed Methodology

- This will involve identifying the sorting algorithms to be implemented, the visual representation of the sorting process, and the user interface design.
- This web-based applications include JavaScript, HTML, CSS, and React.
- This will involve writing the code to perform each of the sorting algorithms. Some common sorting algorithms include bubble sort, insertion sort, selection sort, quicksort, mergesort, and heapsort.
- The final step is to involve the running of the application and ensuring that it performs as expected.

PROPOSED ALGORITHM

Bubble Sort

Bubble sort is a simple sorting comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order.

Insertion Sort

Insertion sort is a comparison-based algorithm in which each element is compared with the elements before it and inserted at the correct place.

Selection Sort

It is a comparison-based algorithm in which each element is compared with all the other elements and the smallest element is selected and swapped with the first element.

PROPOSED ALGORITHM

Merge Sort

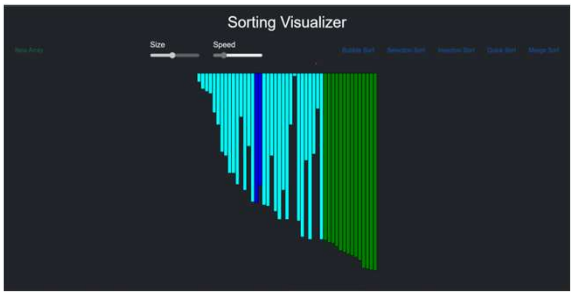
Merge sort is a sorting algorithm that uses a divide and conquer approach. It divides the array into two halves, sorts them recursively, and then merges the sorted halves.

Quick Sort

Quick sort is a sorting algorithm that uses a divide and conquer approach. It selects a pivot element from the array and partitions the array into two halves based on the pivot element. The two halves are then sorted recursively.

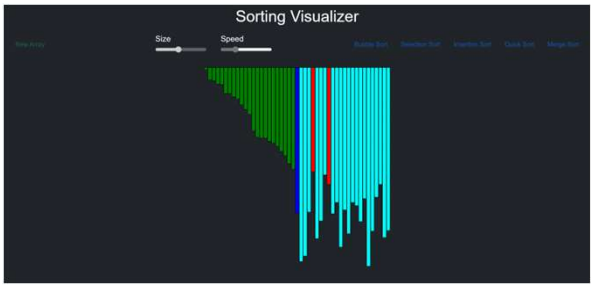
Results

Bubble Sort



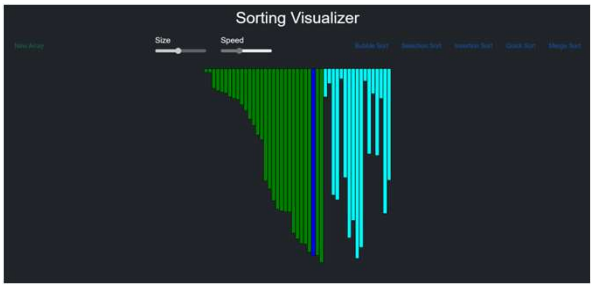
Results

Selection Sort



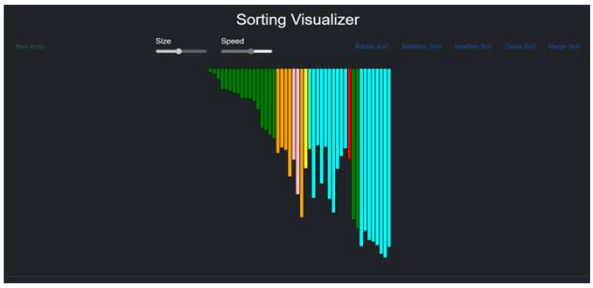
Results

Insertion Sort



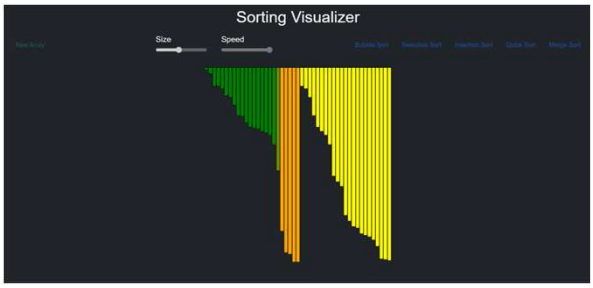
Results

Quick Sort

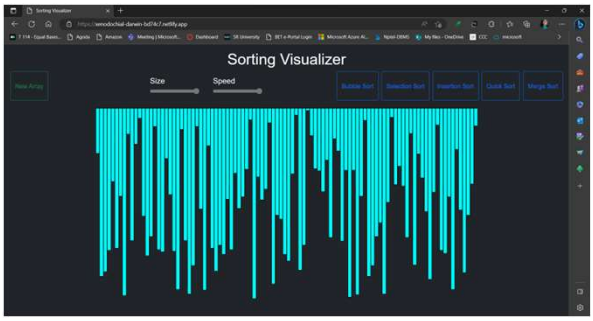


Results

Merge Sort



Web Page Format



Conclusion

- In conclusion, a sorting visualizer project can be a fun and educational way to learn about sorting algorithms and practice programming skills.
- By implementing different sorting algorithms and visualizing the sorting process, you can gain a deeper understanding of how they work and their relative efficiency.
- Overall, this project is a great way to combine programming and visualization skills to create a useful and interactive tool.

References

1. CORMEN, T. H.; LEISERSON, C. E.; RIVEST, D. L.; STEIN, C. Introduction to algorithms. Second Edition. 2001. ISBN 0-262-03293-7.
2. KNUTH, D. The Art of Computer Programming: Fundamental Algorithms. Third Edition. 2004. ISBN 0-201-89683-4.
3. SIPSER, M. Introduction to the Theory of Computation. Boston, MA: PWS Publishing Company, 1997. ISBN 0-534-94728-X
4. KNUTH, D. The Art of Computer Programming: Sorting and Searching. Second Edition. 2004. ISBN 0-201-89685-0.

THANK YOU