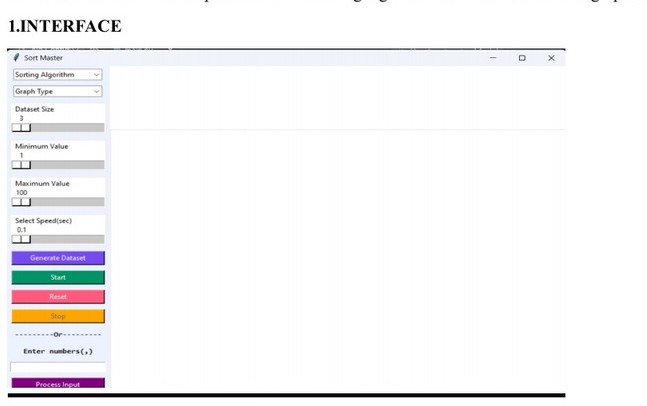
OBJECTIVE



ABSTRACT

Sorting algorithms are basic in computer science and play a

vital role in various applications.

The Sorting Visualizer Project is an interactive tool developed in Python using the Tkinter library, aimed at providing users with a visual understanding of various sorting algorithms.

The project offers a user-friendly graphical interface where users can select from a range of sorting

algorithms, input custom data sets, and control the sorting speed. The project implementation

includes several sorting algorithms such as Bubble Sort, Selection Sort, Merge Sort, and others.

Overall, the Sorting Visualizer Project serves as a valuable educational tool for individuals

seeking to deepen their understanding of sorting algorithms and their applications.

INTRODUCTION

Sort Master is designed to explore and facilitate easy understanding of all sorting algorithms.

The GUI is designed to be easy to use and navigate, with buttons and sliders that make it simple

for users to control the sorting process. For example, there might be a button that you can click to start the sorting process, and sliders that you can move to control the speed of the sorting animation. This project offers a variety of different algorithms that you can use to sort your data. Some of these algorithms include Bubble Sort, Selection Sort, and Merge Sort. Each

algorithm has its own unique way of sorting data, and you can choose the one that best fits your

needs. The ability to input custom data sets or generate random data enhances the tool's

versatility, enabling users to test algorithms with their own data.

You can also adjust the speed

to make the sorting process go faster or slower, depending on your preferences. Our GUI is

designed to offer users a diverse range of graph types for visualizing the sorting process. These

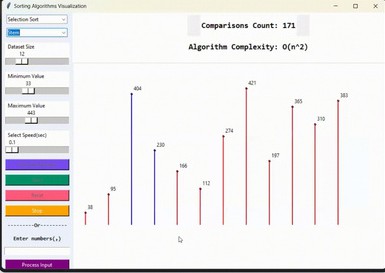
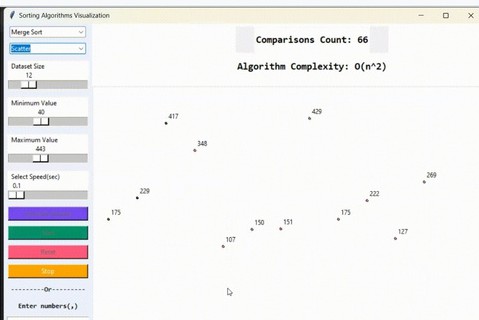
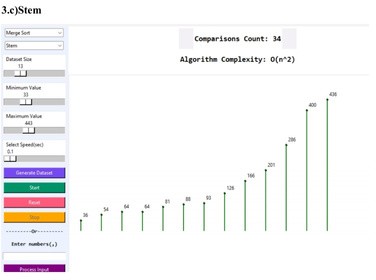
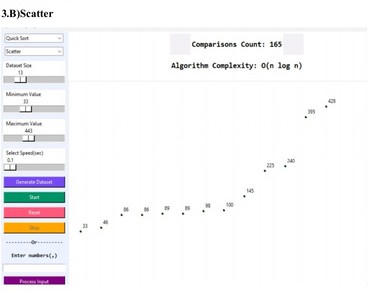
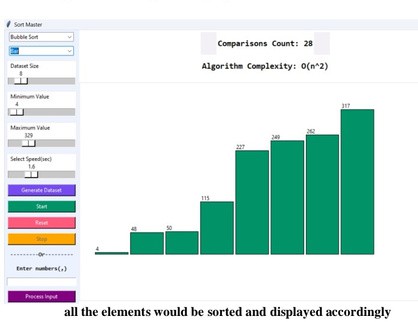
include bar graphs, stem plots, and scatter graphs, each providing a unique perspective on how

the sorting algorithms rearrange data elements. As the sorting algorithm runs, it provides a realtime visualization of the sorting process. This means that you can see the data being sorted as

it happens, with colours and movements that show how the algorithm is rearranging the data

elements. Additionally, Sort Master calculates and displays the time complexity and number of

comparisons made during the sorting process. This visual representation helps to illustrate



TECHNOLOGIES

PYTHON:

Python is a high-level programming language known for its simplicity and readability. It's widely used for various applications, including web development, data analysis, artificial intelligence, and more. In the context of a sorting visualizer, Python serves as the primary programming language for implementing the logic behind the application's functionality.

TKINTER:

Tkinter is a standard GUI (Graphical User Interface) library for Python. It provides a set of tools and widgets to create interactive graphical interfaces for applications. Tkinter allows developers to design windows, buttons, sliders, and other GUI components, making it ideal for creating the user interface of a sorting visualizer.

SCREENSHOTS

EDUCATIONAL VALUES

1. Visual Learning: Many students grasp concepts more easily through visual representation. Sorting visualizers demonstrate abstract algorithms in a tangible way, making it easier to understand how they work.
2. Immediate Feedback: They provide instant visual feedback on the process of sorting, helping students see the immediate impact of each step and the overall progression toward a sorted list.
3. Comparison of Algorithms: By displaying multiple sorting algorithms side-by-side, sorting visualizers allow students to compare their efficiency, steps, and performance, fostering a deeper understanding of algorithmic complexity.
4. Interactive Learning: Many visualizers are interactive, letting students manipulate data and see the effects in real-time. This interactivity can enhance engagement and retention.
5. Concrete Examples: Sorting visualizers offer concrete examples of algorithmic concepts such as time complexity, space complexity, and stability, making theoretical aspects more relatable.

Submitted By :

Renuka Deshpande, Sanjana Dorkhande,

Vaibhavi Niwant.

Guided By :

Prof. Avantika Wadaskar

In conclusion, "Sort Master" effectively enhances the learning experience by providing a clear and interactive visualization of sorting algorithms. By making abstract concepts more accessible and engaging, it helps students gain a deeper understanding of algorithmic principles and improves their ability to compare and implement different sorting techniques. The immediate visual feedback and interactive elements also aid in developing essential skills like debugging and problem-solving, making "Sort Master" a valuable educational resource in computer science education.

REFERENCES

1. Fundamentals of computer algorithms Ellis Horowitz Sartaj Sahni Solutions pg.no.121.127.

[Online].Available:https://kailash392.wordpress.com/wp- content/uploads/2019/02/fundamentalsof-computer-algorithms-by-ellis- horowitz.pdf

1. Introduction to Computing and Problem Solving Using Python |E balagurusamy Available from:https://fliphtml5.com/manzw/rsdl/basic.
2. Duta structures & Algorithm in Python By Michale T. Goodrich Roberto Tamassial Michale H.Goldwasser.

Availablefrom:https://nibmehub.com/opac- service/pdf/read/Data%20Structures%20and%20Algorithms%20in%20Python.pdf

1. GeeksforGeeks. Available from:

[https://www.geeksforgeeks.org/i.](http://www.geeksforgeeks.org/i)

1. Stackoverflow. Available from:

https://stackoverflow.com/i

CONCLUSION



Renuka Deshpande, Sanjana Dorkhande,Vaibhavi Niwant Prof. Avantika Wadaskar

Department Of Computer Engineering, MKSSS Cummins College Of

Engineering For Women Nagpur

SORT MASTER

|  |
| --- |
| The project is a sorting visualizer application that demonstrates the functionality of diverse sorting algorithms. It offers users an interactive platform to visualize the sorting process in real-time and understand the performance of algorithms like Bubble Sort, Quick Sort, Selection Sort, Merge Sort, and Insertion Sort.  With an intuitive GUI interface,users can interact with the algorithms, customize animation speed, and observe sorting operations dynamically.  The project aims to provide a comprehensive and educational experience for users interested in exploring |
| PROPOSED SYSTEM |
| KEY FEATURES  Graphical Interface: Easy-to-use with buttons and sliders to control the sorting process.  Algorithm Selection: Includes multiple sorting algorithms such as Bubble Sort, Selection Sort, Merge Sort, and more.  Custom Data Sets: Allows users to input custom data or generate random data.  \*Speed Control: Adjust sorting speed with sliders for a faster or slower animation.  Diverse Graph Types: Visualize sorting with bar graphs, stem plots, and scatter graphs.  Real-time Visualization: Watch the sorting process with dynamic, colorful animations.  Performance Metrics: Displays time complexity and the number of comparisons made during sorting |
|  |