|  |
| --- |
|  |
|  |
| |  |  |  | | --- | --- | --- | | Anuvind M P | [Date] | [Course title] | |

Sort-search algorithm visualizer

**Guide : Dr. Divya Udayan**

**Group Members :**

**Anuvind M P (AM.EN.U4AIE22010)**

**Harishankar Binu Nair (AM.EN.U4AIE22023)**

**R S Harish Kumar (AM.EN.U4AIE22042)**

Amrita school of Computing, Amritapuri

**Abstract**

In the realm of computer science education, algorithm visualization serves as a powerful tool for enhancing understanding and intuition about the fundamental processes that underpin computation. This project introduces a comprehensive algorithm visualization tool that not only demonstrates the inner workings of sorting and search algorithms but also allows users to actively engage with the algorithms through their own input data.

Implemented using Pygame, Tkinter, and MySQL, our tool offers a unique approach to algorithm visualization by enabling users to input their own arrays or graphs for sorting and searching. This interactive capability provides users with a hands-on experience, allowing them to witness firsthand how algorithms such as Bubble Sort, Insertion Sort, Selection Sort, Heap Sort, Quick Sort, Merge Sort, Radix Sort, Linear Search, Binary Search, Breadth First Search, and Depth First Search operate on their specific data sets.

The visualizations generated by our tool provide a real-time representation of algorithm execution, with user-input data dynamically incorporated into the algorithmic process. This dynamic interaction not only enhances understanding of algorithmic concepts but also fosters a deeper appreciation for the efficiency and effectiveness of various algorithms.

Users have the ability to interact with the visualizations, pausing and stepping through the algorithms to observe each step in detail. Additionally, the tool offers customization options, allowing users to adjust array sizes, animation speeds, and algorithm parameters to explore different scenarios and gain insights into algorithm behaviour under varying conditions.

By leveraging the Pygame library for sorting algorithms, the Tkinter library for search algorithms, and MySQL for data storage, our tool provides a comprehensive and immersive learning experience for users interested in algorithm visualization. Whether used for educational purposes or practical algorithm analysis, our tool aims to empower users to explore algorithm behavior and enhance their computational thinking skills.

**Introduction**

In the field of computer science, understanding algorithms is essential for developing efficient and effective software solutions. Algorithm visualization is a powerful tool that aids in this understanding by providing a visual representation of how algorithms work. This project introduces a comprehensive algorithm visualization tool that not only demonstrates the inner workings of sorting and search algorithms but also allows users to actively engage with the algorithms through their own input data.

The tool is designed to be interactive and educational, with a focus on user input. Leveraging the Pygame library for sorting algorithms, the Tkinter library for search algorithms, and MySQL for data storage, our tool provides a robust platform for exploring algorithm behaviour. Users can input their own arrays or graphs, enabling them to see how algorithms such as Bubble Sort, Insertion Sort, Selection Sort, Heap Sort, Quick Sort, Merge Sort, Radix Sort, Linear Search, Binary Search, Breadth First Search, and Depth First Search operate on their specific data sets.

The visualizations generated by our tool offer a real-time representation of algorithm execution, with user-input data dynamically incorporated into the algorithmic process. This dynamic interaction not only enhances understanding of algorithmic concepts but also fosters a deeper appreciation for the efficiency and effectiveness of various algorithms.

One of the key features of our tool is its interactivity. Users can interact with the visualizations, pausing and stepping through the algorithms to observe each step in detail. Additionally, the tool offers customization options, allowing users to adjust array sizes, animation speeds, and algorithm parameters to explore different scenarios and gain insights into algorithm behavior under varying conditions.

Whether used for educational purposes or practical algorithm analysis, our tool aims to empower users to explore algorithm behavior and enhance their computational thinking skills. By providing a hands-on experience with algorithm visualization, we hope to inspire curiosity and deepen understanding of the algorithms that power modern computing systems.

**Requirements & Specifications**

1 page

**Project plan**

**Design strategy**

Both total 2 or 3 page

**Code snippets**

3 or 4 pages

**Summary**

1 page