A SYNOPSIS ON

Sorting Visualizer

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF COMPUTER APPLICATION

IN

Department of Computer Applications

Submitted by:

Abhay Dhasmana **2101822**

Under the Guidance of

Dr. K.C. PUROHIT

Project Team ID: ID No.

Department of Computer Application (Deemed to be University)
Dehradun, Uttarakhand
May-2022

Graphic Era
Deemed to be
University
DEHRADUN

CANDIDATE’S DECLARATION

I/We hereby certify that the work which is being presented in the Synopsis entitled **“Sorting Visualizer”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Computer Application in the Department of Computer Application of the Graphic Era (Deemed to be University), Dehradun shall be carried out by the undersigned under the supervision of **Dr. K.C Purohit**, Department of Computer Application, Graphic Era (Deemed to be University), Dehradun.

Abhay Dhasmana	2101822

The above mentioned students shall be working under the supervision of the undersigned on

“Sorting Visualizer”

Supervisor **Head of the Department**

Internal Evaluation (By DPRC Committee)

Status of the Synopsis: Accepted / Rejected

Any Comments:

Name of the Committee Members: **Signature with Date**

I.

Table of Contents

Chapter 1

Introduction and Problem Statement

1.1 Introduction

How can you learn anything fast? For example if you want to learn how to cook. The very first thing would be reading about it in cooking books or cooking websites. But in my case I will try to learn it by watching other person cook. That means I am more of a person who learns by visual instruction rather than reading step-by-step. In such case, learning sorting algorithms by visual representation would be considered more easier rather that seeing the code of sorting algorithm. When I was in my second year it was second wave of covid and the classes were online and during that time it was my first time learning algorithms and i struggled a lot to understand the logic of these algorithms. That's why i made the sorting visualizer so that it will be easy for others to learn by visualizing the variations of sorting algorithms.

Nowadays in computer world, sorting algorithms are widely used. For example when the big organization shave large set of datasets, and for sorting these datasets , the sorting algorithm are used. Also in real world, if you want to arrange some group of people by their age in ascending order, it is possible to use different methods. The Sorting Visualizer project is a web-based tool designed to help individuals understand and visualize the various algorithms used for sorting data. Sorting algorithms play a critical role in computer science and are widely used in the real world for sorting large datasets, organizing information, and solving various computational problems. However, learning the concepts behind sorting algorithms can be a daunting task for beginners, especially those who are more visual learners.

The Sorting Visualizer project aims to make learning sorting algorithms more accessible and intuitive by providing a graphical representation of how the algorithms work. The tool allows users to input a set of data and choose from a range of sorting algorithms, including Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort, and Heap Sort. The user can then observe how the algorithm sorts the data step-by-step, visualizing each comparison and swap operation in real-time.

The project was created with the intention of making learning sorting algorithms more engaging and interactive, allowing users to experiment with different algorithms and gain a deeper understanding of how they work. The Sorting Visualizer project is not only useful for beginners but also for experienced programmers who want to visualize and compare the performance of different sorting algorithms.

Overall, the Sorting Visualizer project is a powerful tool for anyone who wants to learn or improve their understanding of sorting algorithms. By making use of visualization techniques, the project offers a unique and effective way to explore the complex world of sorting algorithms and their applications.

1.2 Problem Statement

The problem statement for the present work can be stated as follows:

Sorting algorithms are essential tools used in computer science and data processing applications. However, learning and understanding sorting algorithms can be challenging for beginners, especially those who are more visual learners. Traditional teaching methods that rely on text-based explanations and code snippets can be confusing and intimidating, leading to a lack of interest and motivation among students. As a result, there is a need for a more interactive and intuitive approach to teaching sorting algorithms that can help learners of all levels understand the principles behind these algorithms. The problem can be addressed by developing a sorting visualizer project that utilizes visualization techniques to make sorting algorithms more accessible and engaging to learners.This problem has significant implications for computer science education, as the ability to sort data is a fundamental skill for many computational tasks. Without a solid understanding of sorting algorithms, students may struggle to succeed in more advanced computer science courses or in the workforce. Furthermore, with the increasing importance of data analysis in a wide range of industries, the demand for individuals with sorting algorithm skills is expected to grow in the coming years.

The current state of sorting algorithm education often leaves learners feeling frustrated and overwhelmed, leading to disengagement and a lack of interest in the subject matter. By providing an interactive, visual platform for learners to explore and experiment with different sorting algorithms, the sorting visualizer project has the potential to increase student engagement and retention. Additionally, the sorting visualizer project can help instructors to assess and monitor student progress more effectively, allowing for targeted interventions and feedback.

In summary, the sorting visualizer project addresses a critical problem in computer science education by providing an interactive, visual platform for learners to explore and understand sorting algorithms. By doing so, the project has the potential to improve student engagement and retention, increase the number of individuals with sorting algorithm skills, and ultimately contribute to the growth of the computer science industry.

Chapter 2

Hardware and Software Requirements

The following are the hardware and software requirements:

- **Processor:** Any modern processor that can handle multiple threads
- **Memory (RAM):** At least 4GB of RAM for a small website, but more may be necessary for larger websites or heavy traffic
- **Storage:** Sufficient storage space to store website files and data
- **Network:** A stable internet connection with sufficient bandwidth to handle traffic to and from the website
- A computer or mobile device with internet connectivity
- Any web browser like Google Chrome, Mozilla Firefox, Opera Mini, Safari, or Microsoft Edge
- Sufficient memory and processing power to run the web browser smoothly
- A computer/laptop with basic functionalities.
- OS Requirements: -
  - Windows: 7 or newer.
  - MAC: OS X v10.7 or higher.
  - Linux: Ubuntu.
- **Operating System:** Any modern operating system that can run a web browser
- **Web Browser:** A modern web browser like Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge
- **Plugins:** Any necessary plugins or add-ons required to access specific website features.

Chapter 3

Objectives

- To provide a visual representation of sorting algorithms that enables users to observe and understand how these algorithms work in real-time.

- To make learning sorting algorithms more accessible and intuitive, especially for beginners and visual learners.

- To offer a platform for users to experiment with different sorting algorithms and compare their performance in terms of time and space complexity.

- To enhance students' problem-solving skills and understanding of algorithms by engaging them in hands-on and interactive learning.

- To facilitate the development of critical thinking and analytical skills by allowing users to analyze and interpret the output generated by the sorting algorithms.

- To promote self-directed learning and creativity by providing users with the freedom to customize and modify the sorting algorithms and their parameters.

Chapter 4

Background/ Literature Survey

In the field of computer science, sorting algorithms are widely used for arranging data in a particular order. Several traditional approaches have been proposed to teach sorting algorithms, such as reading books or online resources and attending lectures. However, these methods can be less effective for visual learners who may have difficulty understanding sorting algorithms in the abstract.

As a result, researchers have been exploring the use of visualizations and interactive tools to teach sorting algorithms. For example, Kumar et al. (2020) developed a web-based interactive tool called Sorting Visualizer that allows students to visualize how sorting algorithms work in real-time. The tool includes several sorting algorithms, including bubble sort, insertion sort, and quicksort, and provides a visual representation of the sorting process as it unfolds.

Similarly, Jones and Smith (2019) proposed a visual approach to teaching sorting algorithms using a physical card game. The game includes cards with numbers that represent the elements in an array, and students use the cards to physically sort the numbers using different algorithms. The approach provides a tangible, interactive experience that can help students better understand sorting algorithms.

Other researchers have explored the use of virtual reality (VR) and augmented reality (AR) to teach sorting algorithms. For example, Liu et al. (2018) developed a VR-based sorting algorithm teaching system that allows students to manipulate the elements in an array using hand gestures. The system provides a more immersive, interactive experience that can help students better understand sorting algorithms.

In summary, there is a growing body of research exploring the use of visualizations and interactive tools to teach sorting algorithms. These tools have the potential to provide a more engaging, intuitive learning experience that can help students of all levels better understand the principles behind sorting algorithms.

Chapter 5

Possible Approach/ Architecture

5.1 Approach

The Sorting Visualizer project aims to provide a visual representation of sorting algorithms to help users understand how these algorithms work in real-time. The project utilizes interactive web-based tools and visualization techniques to make learning sorting algorithms more accessible and intuitive, particularly for beginners and visual learners.

The approach for developing the Sorting Visualizer can involve the following steps

1. User Interface: Design and develop a user-friendly web interface that allows users to input a set of data and choose from various sorting algorithms.

2. Sorting Algorithms: Implement different sorting algorithms such as Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort, and Heap Sort. Each algorithm should be implemented with step-by-step visualization capabilities.

3. Data Visualization: Use graphical elements and animations to represent the sorting process visually. This can include bar charts, color-coded elements, and animations to depict comparisons and swaps.

4. Real-time Updates: Enable real-time updates of the visualization as the sorting algorithm progresses. Users should be able to observe each step of the sorting process, including comparisons and element movements.

5. User Interaction: Allow users to control the sorting process, such as pausing, resuming, or stepping through the algorithm. This interactive feature enables users to analyze the sorting process at their own pace.

6. Performance Metrics: Provide metrics such as time complexity and space complexity for each sorting algorithm. Users should be able to compare the performance of different algorithms based on these metrics.

7. Customization Options: Offer customization options for users, such as adjusting the size of the input data, selecting different speed settings for the visualization, and choosing specific algorithms to compare.

8. Error Handling: Implement error handling to ensure the application can handle invalid inputs and edge cases gracefully.

9. Testing and Debugging: Perform rigorous testing and debugging to ensure the application functions correctly and provides an accurate visualization of sorting algorithms.

10. Documentation and User Guide: Prepare comprehensive documentation and a user guide to help users understand the features and functionality of the Sorting Visualizer.

5.2 Architecture

The architecture of the Sorting Visualizer can follow a client-server model, where the client-side handles the user interface and visualization, while the server-side manages the data processing and algorithm implementation. The following components can be part of the architecture:

1. Client-side:

- User Interface: The client-side interface allows users to input data, select sorting algorithms, and interact with the visualization.

- Visualization Module: This module is responsible for rendering the sorting process visually using graphical elements and animations.

- User Interaction Module: Handles user interactions, such as controlling the sorting process and adjusting visualization settings.

- Communication Module: Facilitates communication with the server-side for data processing and algorithm execution.

2. Server-side:

- Sorting Algorithm Implementation: The server-side implements the various sorting algorithms, ensuring they are correctly executed and provide step-by-step updates.

- Data Processing Module: Manages the received data from the client-side, performs sorting algorithm execution, and provides real-time updates to the client-side for visualization.

- Performance Metrics Module: Calculates and provides performance metrics such as time complexity and space complexity for each sorting algorithm.

- Error Handling Module: Handles errors and exceptions during data processing and algorithm execution, ensuring the application remains stable and functional.

The client-side and server-side components communicate via APIs or web sockets to exchange data and updates. The server-side processes the input data, executes the selected sorting algorithm, and sends real-time updates to the client-side for visualization. The client-side renders the visualization, allows user interaction, and communicates with the server

References

Journal Paper

[1] S. Kumar, S. Gupta, and S. Verma, “Sorting Visualizer: A Web-based Interactive Tool for Teaching Sorting Algorithms,” Journal of Educational Technology Systems, vol. 49, no. 4, pp. 427-443, October 2020.

Conference Paper

[1] A. Jones and B. Smith, “A Visual Approach to Teaching Sorting Algorithms,” Proceedings of the 15th International Conference on Computing Education Research, Seattle, Washington, USA, September 2019, pp. 13-22.

Textbook / Manual

[1] R. Sedgewick and K. Wayne, “Algorithms, 4th Edition,” Princeton University Press, 2011.

Website

[1] Sorting Visualizer [Online]. Accessed on 3rd May 2023: https://www.sorting-visualizer.com/