Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavre

A Project Proposal

on

"Algolizer"

[Code No: COMP 206]

(For partial fulfillment of II/I Year/Semester in Computer Science/Engineering)

Submitted by:

Suraj Bhattarai (6)
Ashitom Budhathoki (9)
Anmol Dahal (11)
Abhinav Lamsal (27)

Submitted to:

Mr. Nabin Ghimire
Department of Computer Science and Engineering

Submission Date: 18th March 2024

Abstract

Algolizer is a GUI based program that aims to develop a visualization tool for various algorithms and data structures. This project aims to create a fun and interactive tool for learning about algorithms and data structures to address the increasing demand for educational tools in the field of computer science. The goal is to develop a platform that enables students and professionals to understand complex concepts through graphical representation of algorithms and data structures. Using C++ and Raylib, the project will implement graphical representation of various algorithms and data structures. The project's objective is to enhance the learning experience by offering a visually immersive way to understand algorithms and data structures. Algolizer will serve as an educational tool that bridges the gap between theoretical knowledge and practical understanding. Integrating this tool into curricula or self-learning resources will enhance the learning experience and encourage better understanding of algorithms and data structures.

Keywords*:* *Algorithm, Data Structure, Visualization tool, C++, Raylib*.

Table Of Contents

List of Figures

Acronyms/Abbreviations

GUI: Graphical User Interface

OpenGL: Open Graphics Library

OS: Operating System

FAQ: Frequently Asked Questions

LTS: Long Term Support

RAM: Random Access Memory

GB: Gigabytes

MB: Megabytes

Chapter 1: Introduction

Algolizer, derived from "algorithms" and "visualization," showcases the most popular algorithms and data structures through interactive visual approach. It's like a visual guide that helps you grasp these concepts easily, making learning and exploring computer science a fun and accessible experience.

1.1 Background

In the field of computer science, efficiency has always been a driving force behind innovations. As computer scientists continually move forward to optimize resource utilization, algorithms have emerged as fundamental tools in achieving this goal. The evolution of algorithms has been proof of creativity and ingenuity of great minds, leading to the development of numerous efficient solutions across various domains.

Recent trends in computer science highlight the growing need for visualizations of algorithms and data structures for better understanding. With the increasing complexity of problems, there has been a growing demand for better ways to understand and analyze algorithms and data structures. This increased demand has led to significant advancements in visualization techniques, making complex concepts easier to understand for more people.

Despite various resources being available related to algorithms and data structures, the visualization of these computational processes has always been a daunting task. The complexity and intricacy of algorithms often make it difficult for learners and even professionals to grasp their inner workings. This limitation highlights the significance of projects like Algolizer, which aims to bridge the gap between complexity and understanding by providing clear visual representations of the most popular and frequently used algorithm and data structures. Such initiatives contribute towards the advancements of algorithmic understanding and application in real-world scenarios.

1.2 Objectives

The primary objective of Algolizer is to provide a firsthand understanding of algorithms and data structures. Our specific objective include:

- Develop visualizations to enhance understanding of popular algorithms and data structures.
- Integrate performance analysis tools to help understand the time and space complexity of algorithms.
- Offer a wide range of algorithms and data structures to cover key concepts in computer science.

1.3 Motivation and Significance

Our project, Algolizer, is born out of a strong passion to simplify complex concepts for students venturing into the field of computer science. Having personally encountered the initial challenges of understanding algorithms and data structures, especially for those new to the discipline, we recognized the pressing need for a user-friendly and interactive learning tool. While existing software solutions do offer algorithm visualization, we observed certain drawbacks and limitations that we believe can be improved upon. This served as the driving force behind the creation of Algolizer, a platform that not only allows users to visualize algorithms in action but also empowers them to customize visualizations and delve into in-depth algorithmic analysis, thereby fostering a clear and straightforward understanding of these fundamental principles.

The significance of our project lies in its ability to address the shortcomings of current systems, offering a seamless and enriching learning experience. Algolizer sets itself apart with its intuitive interface, extensive library of algorithms, and robust performance analysis tools – all integrated into a single platform. This comprehensive approach not only simplifies complex concepts but also enables users to enhance their problem-solving skills and cultivate a deeper understanding of algorithmic thinking. By bridging the gap between theoretical knowledge and practical application, Algolizer

aims to enhance the learning experience surrounding algorithms and data structures, making it more engaging and accessible for learners and professionals alike.

1.4 Expected Outcomes

The expected outcomes of Algolizer are set to bring about a host of positive impacts, especially for users who are new to the field of computer science. Through the interactive visualizations and educational content provided by Algolizer, we anticipate a significant improvement in users' understanding of algorithms and data structures. This enhanced understanding will not only help them grasp the concepts more effectively but also enable them to retain the knowledge for an extended period.

Furthermore, Algolizer's user-friendly approach is designed to enhance engagement and make learning algorithms more accessible. By simplifying complex concepts and presenting them in a clear and interactive manner, Algolizer aims to foster a positive learning experience for users. This is expected to result in improved problem-solving skills and the development of algorithmic thinking abilities among users, empowering them to approach computational challenges with confidence and proficiency.

Additionally, Algolizer's potential impact extends beyond individual users to the broader educational and professional communities. Positive feedback and widespread adoption of Algolizer within these communities would signify its effectiveness and value in facilitating algorithm understanding and application. This recognition would further solidify Algolizer's position as a valuable tool for enhancing learning outcomes and supporting the growth of algorithmic knowledge and skills among learners and professionals alike.

Chapter 2: Related Works

There are already similar projects tackling the objectives of Algolizer, focusing on visualizing algorithms and data structures for educational purposes. These existing projects serves as valuable reference for Algolizer's development. These projects include:

2.1 VisualAlgo

VisualAlgo is an innovative platform designed to revolutionize the way algorithms and data structures are learned and understood. By combining interactive visualizations with comprehensive explanations, VisualAlgo provides users with a unique and engaging learning experience. Whether you're a student delving into the basics or a seasoned professional exploring advanced concepts, VisualAlgo offers a visually immersive journey that simplifies complex ideas and fosters a deeper understanding of algorithms and data structures.

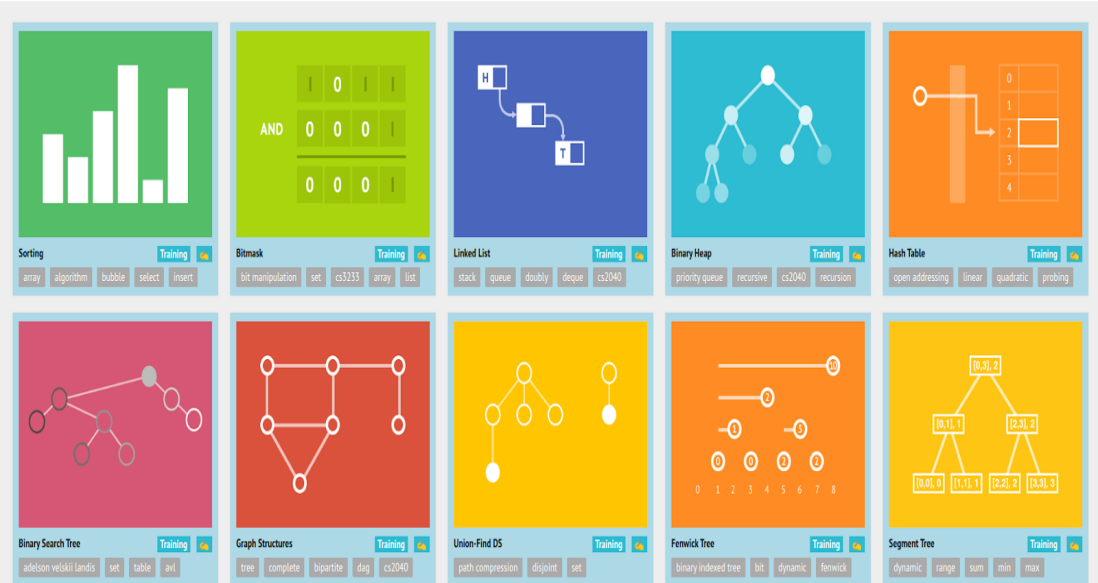

Figure 2. 1 VisualAlgo

2.2 Algorithm-Visualizer.org

Algorithm-Visualizer.org is a cutting-edge platform dedicated to enhancing algorithmic learning through interactive visualizations. It offers a wide range of algorithms and data structures, each accompanied by detailed explanations and step-by-step visualizations. Users can explore various sorting, searching, and graph algorithms, gaining valuable insights into their operations and complexities. Algorithm-Visualizer.org's intuitive interface and real-time animations make learning algorithms not only educational but also entertaining. Whether you're a student, educator, or professional, Algorithm-Visualizer.org provides a powerful tool to deepen your understanding of algorithms and sharpen your problem-solving skills.

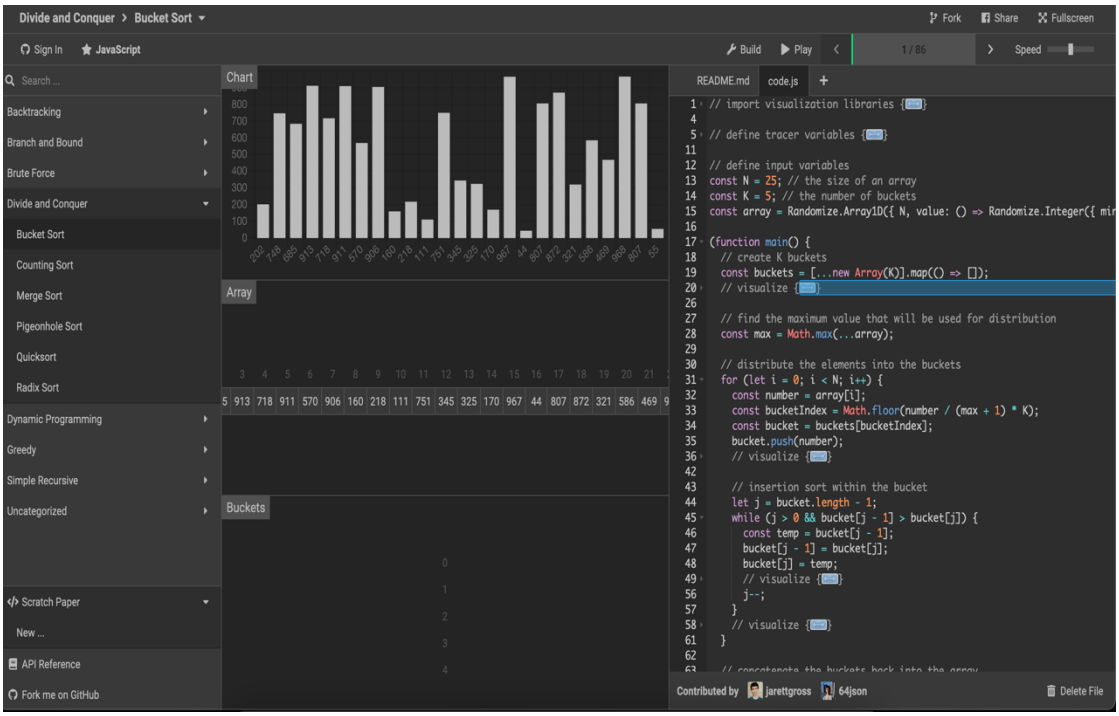

Figure 2. 2 Algorithm-Visualizer.org

Chapter 3: Procedure and Methods

In this chapter, we outline the sequential process guiding the development of Algolizer. We'll discuss how we approach different aspects of the project, ensuring a systematic and effective application.

3.1 Planning and Research

In this phase, our primary objective is to meticulously define the goals, scope, and timeline of Algolizer. We will conduct thorough research to identify the most critical features and functionalities that will meet the needs of our target audience. By understanding the preferences and challenges of our users, we will tailor Algolizer to provide a comprehensive learning experience. Additionally, we will establish a detailed project plan with clear milestones and deadlines, ensuring a structured and efficient development process. Our ultimate goal is to deliver a cutting-edge educational platform that empowers users to master algorithms and data structures effectively and efficiently.

Figure 3. 1 Workflow

3.2 Designing

During the designing phase of Algolizer, we will delve into developing the code layout and UI with a thorough understanding derived from our research efforts. Our approach involves breaking down the codebase into multiple modules, such as algorithms, visualization, UI components through use of various programming paradigms. In this way we ensure a cohesive and organized development process that facilitates efficient feature implementation and seamless integration of functionalities. This approach not only enhances the effectiveness of Algolizer but also aligns it with the envisioned functionality, catering to the needs and preferences of our users. Additionally, we will leverage our research insights to design an intuitive UI that promotes user engagement and simplifies the learning process for users within Algolizer.

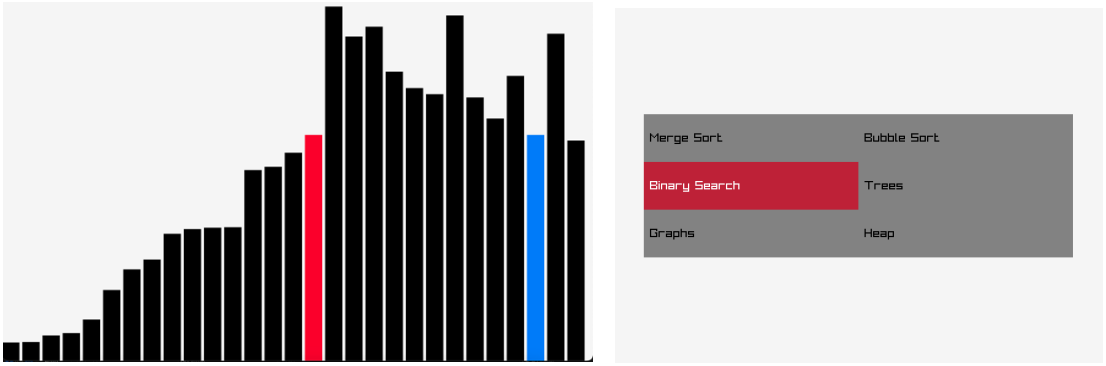

Figure 3. 2 UI Design

3.3 Development

During the development phase of Algolizer, our approach will be centered around leveraging the capabilities of C++ and Raylib for graphics rendering, with the goal of delivering a visually captivating and immersive experience for users engaging with algorithm visualizations. By the use of CMake, we aim to streamline the compilation process, optimize code efficiency, and maintain a structured and organized codebase,

ensuring smooth development iterations and faster feature implementation. Collaborative efforts within the team will be facilitated through GitHub, serving as a central platform for version control, code sharing, and real-time progress tracking. This collaborative environment will enable seamless communication, efficient code review, and effective bug resolution, leading to the creation of a robust and feature-rich software solution that meets the needs and expectations of our users.

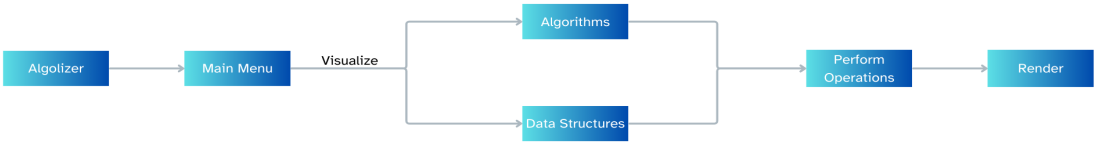

Figure 3. 3 Program Flow Diagram

3.4 Testing and Debugging

During the development phase of Algolizer, testing and debugging play a crucial role in ensuring the platform's functionality and reliability. We will conduct regular testing to identify and address any bugs or issues that may arise during the development process. This includes functional testing to verify that features work as intended, usability testing to assess the user experience and performance testing to optimize speed and efficiency. We will prioritize user feedback and conduct beta testing to gather insights from real users, allowing us to make necessary improvements and enhancements.

3.5 Documentation

In the final phase of the Algolizer project, the focus will be on creating comprehensive documentation to guide users and developers in utilizing the platform effectively. Documentation is essential for providing users and developers with comprehensive information about Algolizer, including its features, functionalities, and usage guidelines. The documentation will cover aspects such as installation instructions, system requirements, user interface navigation, and troubleshooting tips. Additionally, we will provide a user manual and FAQ section to assist users in utilizing Algolizer effectively.

Chapter 4: System Requirement Specifications

In this chapter, we delve into the System Requirement Specifications for Algolizer. The following sections provides an insight for the software and hardware requirements necessary for developing this interactive algorithm visualization platform.

4.1 Software Specifications

1. Operating System: Algolizer will be developed and tested on Windows, macOS, and Linux to ensure compatibility and functionality across different platforms.
2. Programming Language: C++ will serve as the primary programming language for developing Algolizer's core functionalities and logic.
3. External Library: Raylib, an open-source C library for rendering graphics, will be used to create interactive algorithm visualizations and user interface elements.
4. Development Tools: CMake will be used for compiling and building Algolizer's codebase, while Git and GitHub will be employed for version control and collaboration among development team members.

4.2 Hardware Specifications

1. Processor: A modern multi-core processor for efficient compilation and rendering.
2. RAM: Minimum 2GB of RAM to ensure smooth performance during development and testing phase.
3. Storage: Algolizer requires at least 500MB of free disk space to store project files, libraries, and development tools without running out of storage capacity.
4. Display: A monitor with a resolution of 1280x800 pixels or higher for optimal experience.

Chapter 5: Project Planning and Scheduling

The project's timeline is depicted in the Gantt chart below, showcasing the planned start and end dates for each phase, major milestones, and the overall project duration.

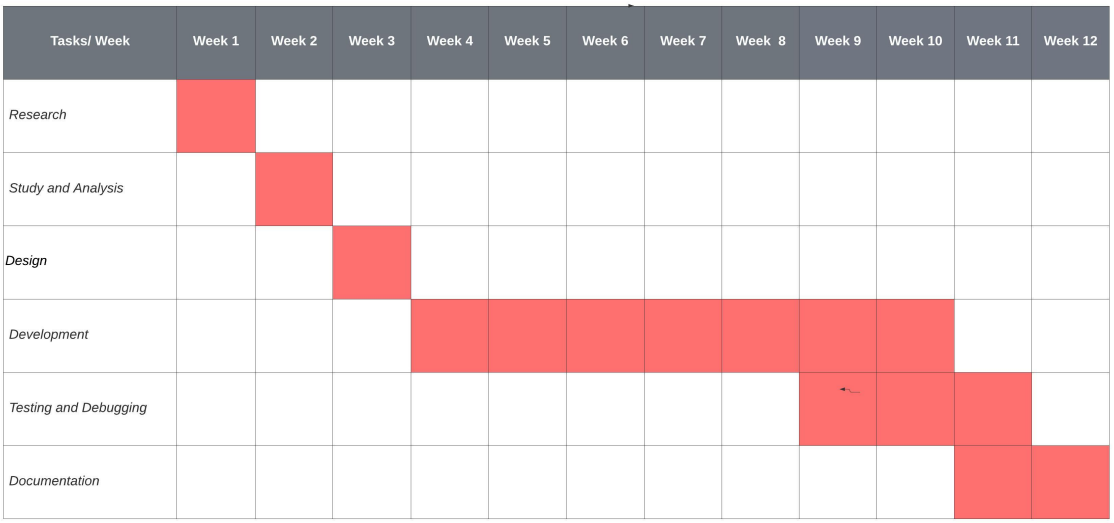

Figure 5. 1 Gantt Chart

5.1 Tasks

1. Research and Brainstorming (Week 1): Conduct research and brainstorming sessions to explore project ideas.
2. Study Analysis and Resource Collection (Week 2): Analyze research findings and gather resources for the selected project.
3. Design Program Workflow (Week 3): Develop a simple design for the program's user interface and functionality.
4. Development (Week 4-10): Initiate coding and implement program functionalities.
5. Testing and Debugging (Week 9-11): Test the program thoroughly and fix any bugs or issues.
6. Documentation (Week 11-12): Create documentation for the program.

References

[1] VisuAlgo: visualizing data structures and algorithms through animation. [Online]. Available: https://visualgo.net/en. Accessed on: Mar. 7, 2024.

[2] Algorithm Visualizer. [Online]. Available: https://algorithm-visualizer.org/. Accessed on: Mar. 8, 2024.

[3] "Learn OpenGL, extensive tutorial resource for learning Modern OpenGL." [Online]. Available: https://learnopengl.com/. Accessed on: Mar. 8, 2024.

[4] "OpenGL - The Industry Standard for High Performance Graphics." [Online]. Available: https://www.opengl.org/. Accessed on: Mar. 9, 2024.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms, Third Edition," MIT Press, 2009.

[6] Raylib Wiki. [Online]. Available: https://github.com/raysan5/raylib/wiki. Accessed on: Mar. 9, 2024.

[7] R. Sedgewick and K. Wayne, "Algorithms," Pearson Education, 2011.