Minor Project I

Conceptualization of linear data structures and sorting algorithms through visualization

R2142220263 - Kovid Khanna
R2142220238 - Aryan Chaudhary
R2142220086 - Ishaan Narayan
R2142220263 - Devansh Goyal

Mentored by:
Dr. Virender Kadyan

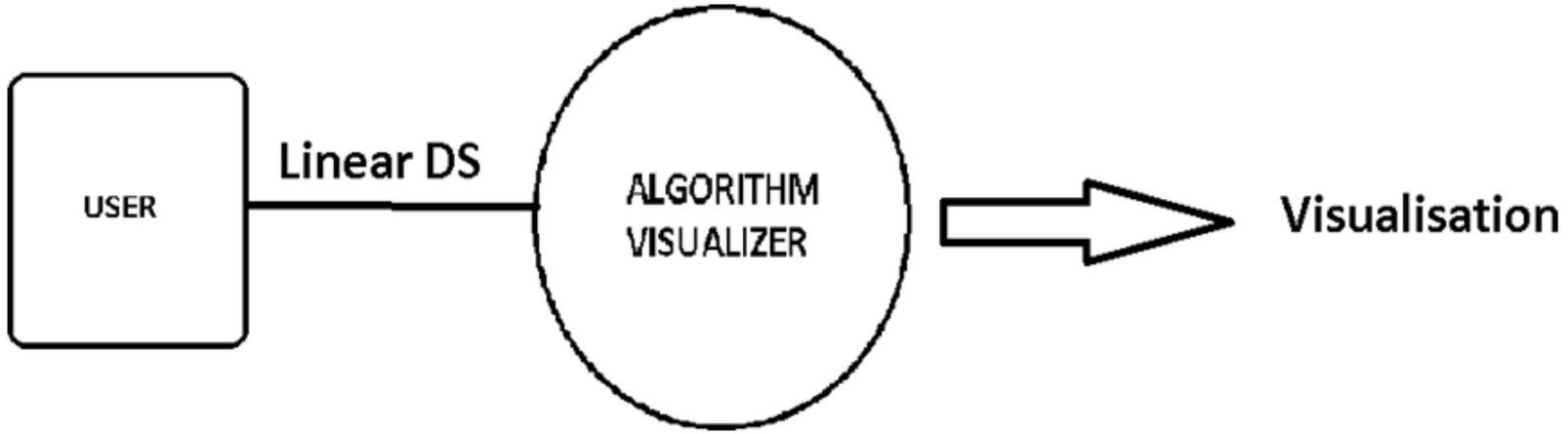
Content

- Introduction
- Literature review
- Objectives
- Methodology
- Features
- Implementation details
- Live demonstration
- Results
- Future scope
- Conclusion

Introduction

Understanding linear data structures is fundamental to mastering computer science concepts, particularly in the field of algorithms and data management. However, these concepts can be challenging for beginners to grasp when presented in traditional, static formats. To address this challenge, we will be developing an interactive framework that visually represents linear data structure algorithms.

Our framework is designed to break down complex operations into step-by-step visualizations (fig 1.), making it easier for users to follow and understand the underlying processes. By integrating key principles from the Design and Analysis of Algorithms, our tool not only simplifies learning but also provides deep insights into algorithmic efficiency. This educational tool is particularly useful for students, educators, and anyone looking to strengthen their understanding of data structures through a hands-on, visual approach.



Literature review

Inference from Literature

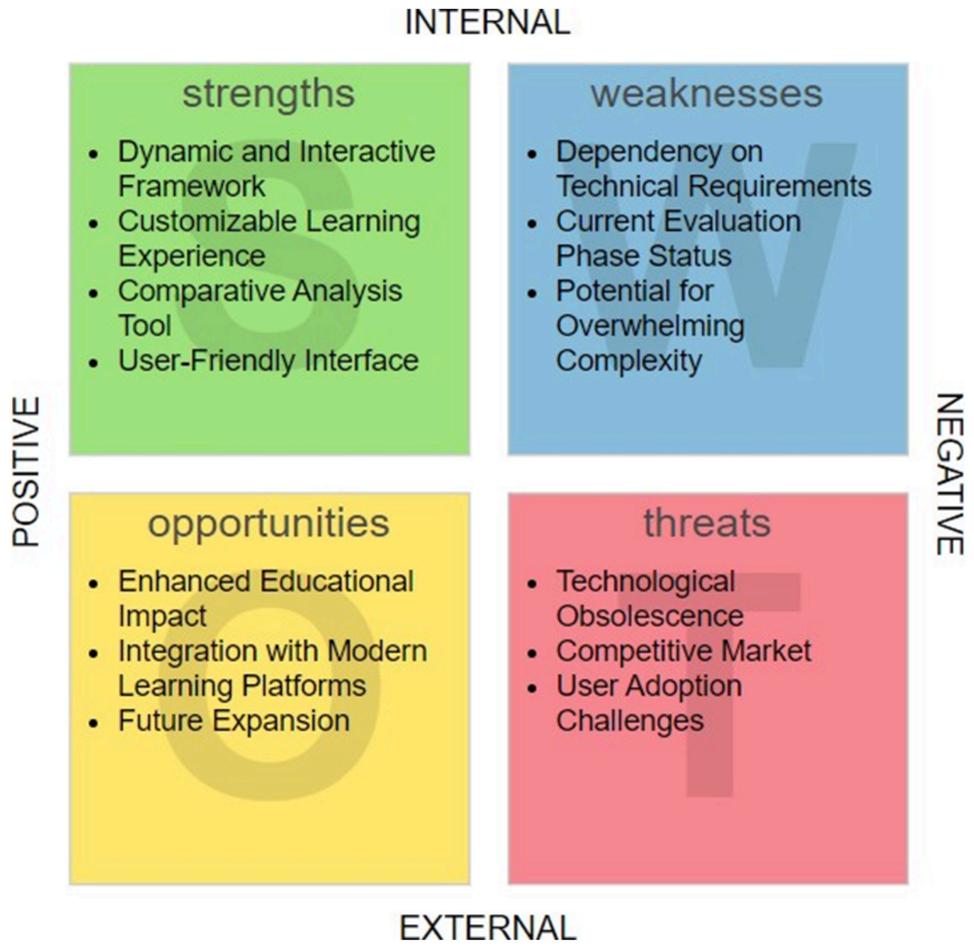
• Modern tools should offer interactive and dynamic visualizations to improve engagement and comprehension, addressing limitations of passive learning methods.

•Tools that encourage user interaction and active participation are more effective in teaching complex concepts than static visualizations.

•Providing options for users to customize their learning paths and select different algorithms supports varied learning preferences and needs.

•Effective visualization tools should include navigable animations, hypertext descriptions, and feedback mechanisms to enhance educational outcomes.

•Gathering user feedback and iterating on the design is critical for refining the tool and ensuring it remains effective and relevant.



Objectives

The Linear Data Structures Visualization project aims to create an interactive, educational platform that simplifies the understanding of linear data structures through dynamic, step-by-step visualizations.

- The First objective is to allow users to provide the necessary data for various linear data structures. The goal is to ensure that the user can easily input and interact with the data, which will be used for further visualization and learning purposes.

- The second objective is to process the data collected from the user by first validating the input, ensuring it meets the required constraints for each data structure. Once validated, the data is organized into the appropriate structure and prepared for further operations like searching or sorting.

- The third objective is to analyze the time and space complexity of the algorithms used in your project. This involves evaluating how the execution time and memory usage of each algorithm scale with the size of the input, and documenting these findings for performance insights.

- The last objective is to use OpenGL, ImGui, GLFW & GLAD to represent data structures visually. This involves initializing, designing graphical representations, implementing dynamic animations for operations like sorting and searching and providing interactive features to enhance the learning experience.

Methodology

- **Designing the input interface**

This framework will use a simple text based interface for taking user inputs; it will collect all the necessary parameters for linear data structures and algorithms.

- **Input validation**

The inputs will be checked for valid size. It will be ensured that the element type matches the requirement of the data structure. Validation of operations will also take place to avoid errors like underflow or overflow. Real time feedback will be provided if inputs violate constraints.

- **Input processing**

Once the data is validated, it will be stores in appropriate linear data structures. The requested operations will be performed while updating the state of the structure.

- **Operations**

Keeping track of the current state of the data structures and showing the changes after each operation is implemented.

- **Initialization of OpenGL, GLAD, GLFW and Dear ImGui**

Setting up the environment includes window management, defining the viewport, background color, and graphical settings using OpenGL and GLUT, with Dear ImGui providing an interactive GUI for user controls.

- **Designing graphical representations**

2D graphics will be used to represent the initialized linear data structure. Arrays will be represented as a set of boxes with index numbers. Stack elements will be represented as blocks stacked vertically.

- **Dynamic animations**

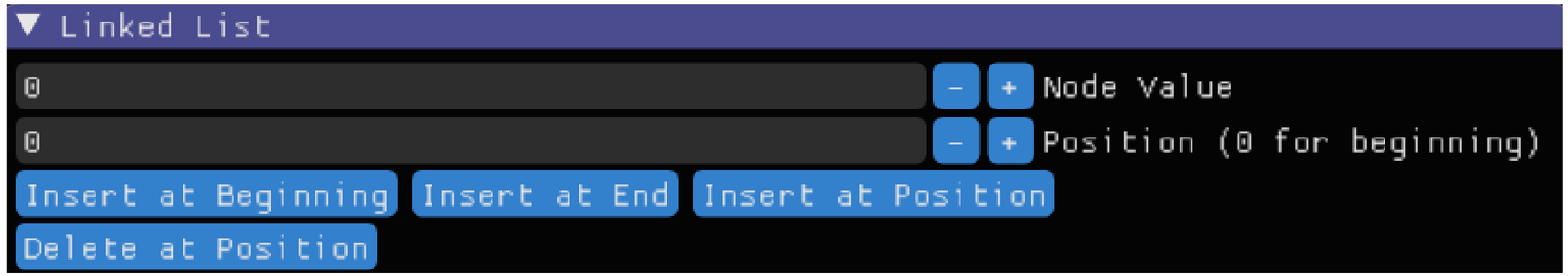
Implementation of animations for different operations and algorithms will take place accordingly. For searching and sorting highlighting of comparisons and swaps will be done.

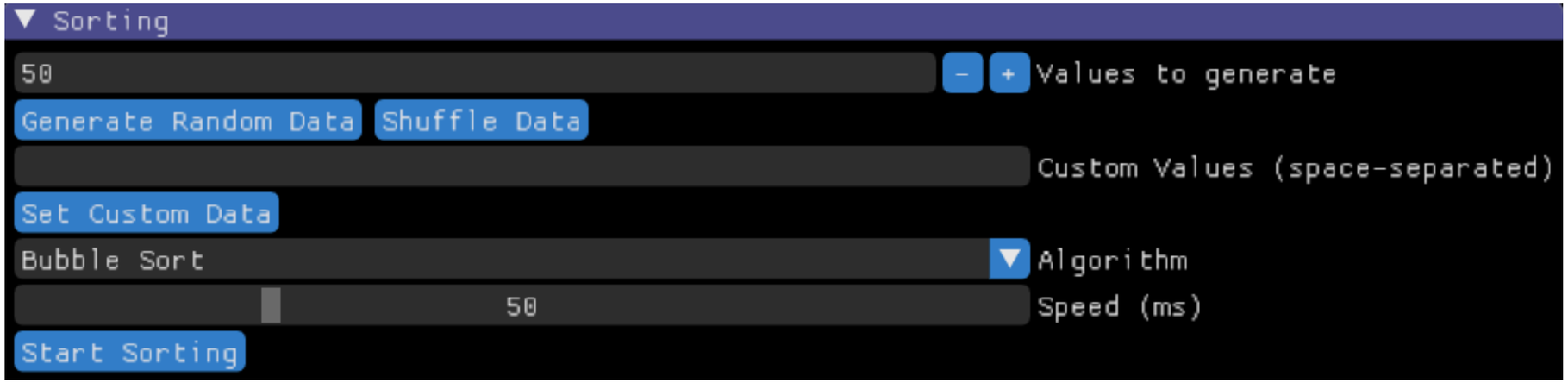
- **Testing**

The project will be tested for a variety of inputs to ensure the system handles edge cases and is user friendly. The framework will also be tested on number of users and feedback will be taken and necessary changes according to the feedback will be done.

Features

- Interactive GUI for operations like enqueue, dequeue, push, pop, insertion, deletion & sorting etc.
- Real-time feedback on linear data structures such as queue, stack and linked lists as well as sorting algorithms.
- Lightweight and fast.
- Robust and reliable making it suitable for educational purposes.





Implementation details

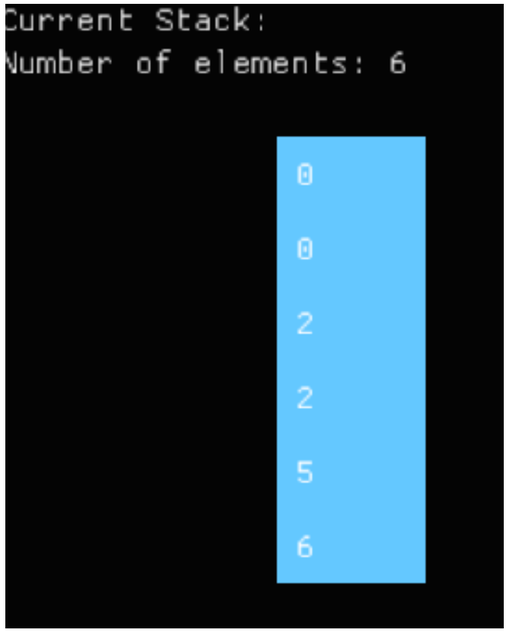
- Core logic - includes all the data structure classes and their functions along with all the functions for the different sorting algorithms used are being implemented in C++.
- GUI is being handled by ImGui allowing users to interact more graphically with the data structures.
- GLAD & GLFW is being used for window and graphical elements management.
- OpenGL is doing all the rendering of the data structures and sorting algorithms.
- Different other standalone C++ libraries are being used miscellaneous tasks.

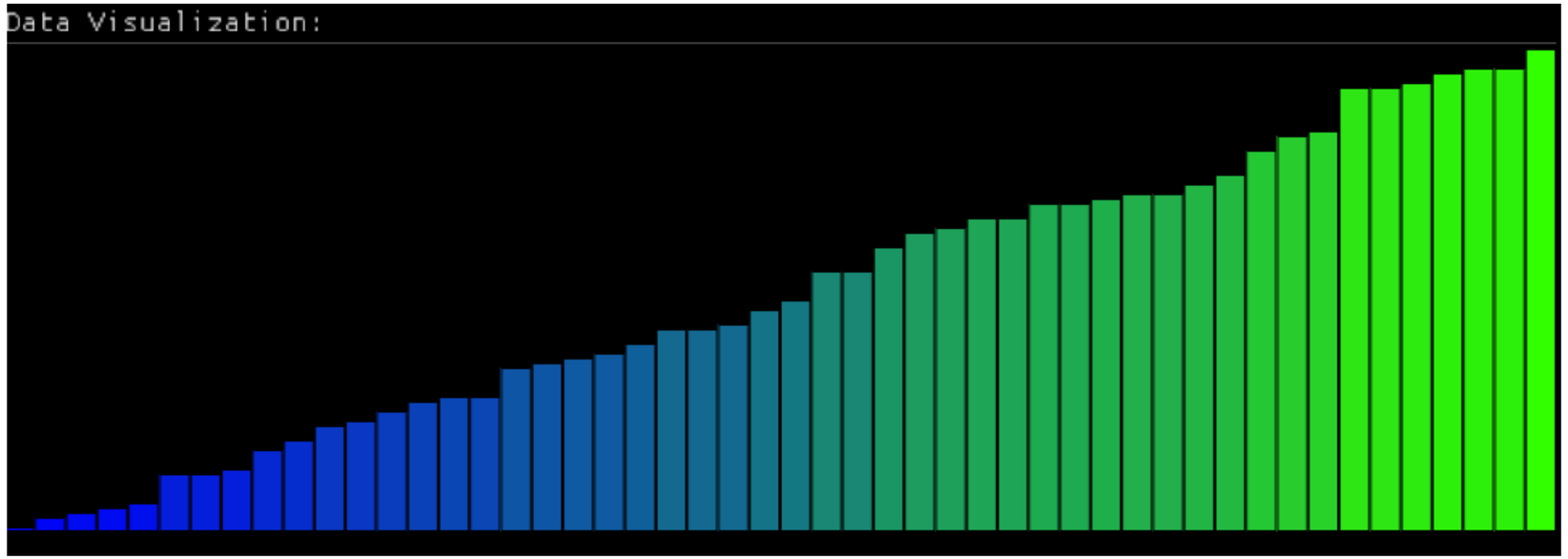
```
#include <imgui/imgui.h>
#include <imgui/imgui_impl_glfw.h>
#include <imgui/imgui_impl_opengl3.h>
#include <GLFW/glfw3.h>
#include <vector>
#include <string>
#include <thread>
#include <chrono>
#include <cstdlib>
#include <ctime>
#include <cmath>
#include <queue>
#include <algorithm>
```

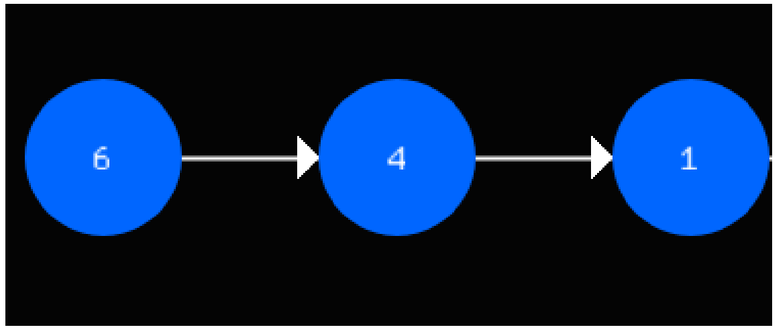
Live demonstration

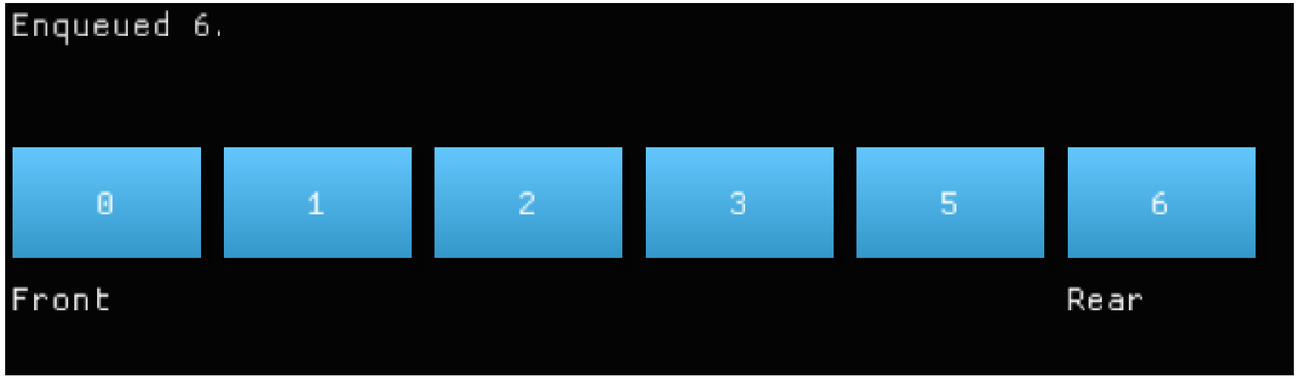
Now we will be demonstrating our project.

Results


Current Stack:
Number of elements: 6
0
0
2
2
5
6


Data Visualization:


6
4
1


Enqueued 6.
0
1
2
3
5
6
Front
Rear

Future scope

- **Expansion to non-linear data structures -** The framework can be extended to include non linear data structures like trees (binary trees, AVL trees) and graphs. This will broaden the range of algorithms and data structures that are being visualized such as tree traversal or graph search, making the tool more comprehensive for learners.

- **Integration of advanced algorithms -** Complex algorithms such as dynamic programming, backtracking and graph algorithms (Dijkstra's, A*) would provide deeper insights into how data structures are used in more advanced computational problems.

- **Collaborative and AI - assisted learning -** Real time collaborative features for group learning and AI-driven educational assistance, like providing personalized feedback, will enhance user interaction and learning efficiency, making the tool suitable for both classrooms and self-paced education.

- **Customizable simulations -** Future versions could allow users to customize their own simulations by adjusting parameters such as operation frequency and memory constraints.This would make the tool more flexible for advanced users and researchers.

Conclusion

The development of this dynamic and interactive framework represents a significant leap forward in the field of computer science education, particularly in the understanding of linear data structures. By offering real-time visual feedback and intuitive interfaces, this tool provides a more engaging and accessible approach to learning complex algorithms. It simplifies traditionally difficult concepts through step-by-step visualization, making it easier for students and educators to explore, experiment, and comprehend the mechanics of linear data structures and their operations.

Moreover, the framework's emphasis on interactivity fosters a deeper learning experience, allowing users to see the immediate results of their actions and analyze the behavior of data structures and algorithms in real-time. This hands-on approach bridges the gap between theoretical knowledge and practical application, thus enhancing retention and understanding of key concepts.

As the tool evolves, there is immense potential to broaden its capabilities and incorporate more advanced algorithms, collaborative features, and support for non-linear data structures, making it a comprehensive educational resource. Ultimately, this project lays the foundation for a robust learning platform that can grow and adapt to the needs of students, educators, and professionals in the field of computer science.

THANK YOU