Max Heap Tree Visualization

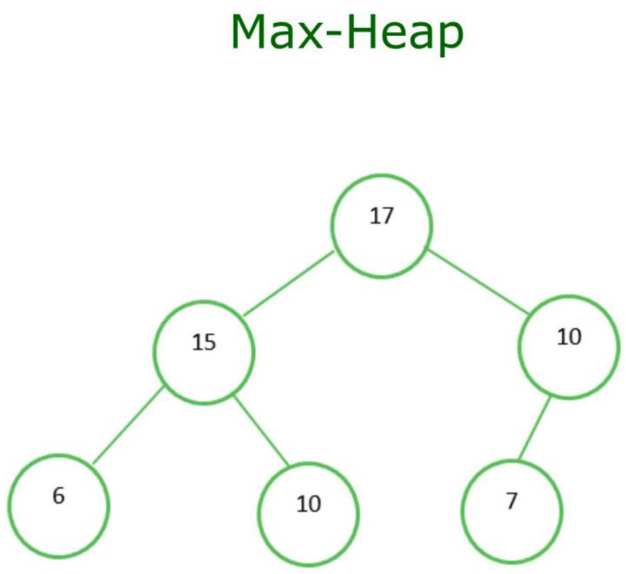
(User Manual)

Table of Contents

Introduction

Binary Tree and Heap: Binary trees and heaps are important data structures in computer science that are used to efficiently organize and manipulate data. They provide efficient storage and retrieval mechanisms for a wide range of applications, including search operations, sorting, and priority queue management.
A binary tree is a hierarchical data structure in which each node has at most two children, referred to as the left child and the right child. The structure of a binary tree allows for efficient searching, insertion, and deletion operations. Binary trees can be classified as binary search trees (BSTs) when they follow a specific ordering property, where the left child of a node is smaller than the node itself, and the right child is larger. This property enables efficient searching and sorting operations, making binary search trees a popular choice for storing and organizing data.
Heaps, on the other hand, are specialized binary trees that satisfy the heap property. A heap is a complete binary tree in which every parent node has a value that is either greater than or equal to (in a max heap) or less than or equal to (in a min-heap) the values of its children. This property allows heaps to be used efficiently for tasks such as priority queue management and heap-based sorting algorithms like heapsort. Heaps provide constant-time access to the maximum (or minimum) element, making them particularly useful in scenarios where quick access to the highest (or lowest) priority element is required.
Both binary trees and heaps have their unique characteristics and applications. Binary trees are versatile and can be used in a variety of situations where efficient searching, sorting, and tree-based operations are needed. On the other hand, heaps are specifically designed to optimize priority-based operations and are commonly used in tasks that involve managing elements with varying priorities. Understanding the properties and operations of binary trees and heaps is crucial for designing efficient algorithms and data structures. They form the foundation for more complex data structures, such as self-balancing binary search trees and priority queues. Mastery of binary trees and heaps allows programmers to effectively organize and manipulate data, leading to improved performance and scalability in a wide range of computational tasks.

Max Heap: A max heap is a complete binary tree in which the value of a node is greater than or equal to the values of its children. Max Heap data structure is useful for sorting data using heap sort



In a max heap, the left and right children of a node must have lower value compared to the parent. This property holds for every level of the heap, from the root node down to the leaves

Project Description

The purpose of the Max Heap Visualization Project is to develop an interactive and visual learning tool that enhances understanding and comprehension of max heaps, a fundamental data structure in computer science. The project aims to provide an intuitive platform for students, educators, and practitioners to explore the inner workings of min heaps, visualize their construction and operations, and grasp the concepts and algorithms associated with them.

The visualization project seeks to address the challenges often faced when learning abstract data structures, such as max heaps, by offering a visual representation that simplifies complex concepts. By providing dynamic and interactive visualizations, the project aims to bridge the gap between theoretical knowledge and practical implementation of max heaps, fostering a deeper understanding of their functionality, applications, and algorithms.

Purpose: The purpose of the Max Heap Visualization Project is to develop an interactive and visual learning tool that enhances understanding and comprehension of max heaps, a fundamental data structure in computer science. The project aims to provide an intuitive platform for students, educators, and practitioners to explore the inner workings of max heaps, visualize their construction and operations, and grasp the concepts and algorithms associated with them

Project Scope: The scope of the Max Heap Visualization Project encompasses the development of a comprehensive software tool that visualizes the construction, manipulation, and operations of max heaps. The project aims to provide an interactive and intuitive platform for users to explore and understand the concepts and algorithms associated with max heaps. The following aspects define the scope of the project

Features:

- Visual representation of max heap structure, including nodes, parent-child relationships, and heap property.

- Interactive animations to demonstrate heap construction, insertion, deletion, and heapification operations.

- Intuitive and user-friendly interface for easy navigation and interaction with the visualization.

○ Options to customize input elements, such as values or keys, and observe the resulting heap modifications.

○ Visualization of key operations, including insertion of elements, deletion of the minimum element, and heapification.

○ Support for varying heap sizes, accommodating large to small datasets for visualization.

○ Thorough testing of the visualization tool to ensure the correctness, accuracy, and reliability of the displayed heap operations.

Tech Stack: The choice of the tech stack for the Max Heap Visualization Project depends on various factors, including the development platform, required functionalities, scalability, and the development team's expertise. Here are some commonly used technologies and frameworks for building a max heap visualization project:

Frontend:

1. HTML5: The standard markup language for creating the structure and content of the visualization application.

2. CSS3: Styling the user interface and applying visual effects to enhance the visualization.

3. JavaScript: The primary programming language for implementing interactivity, animations, and handling user input.

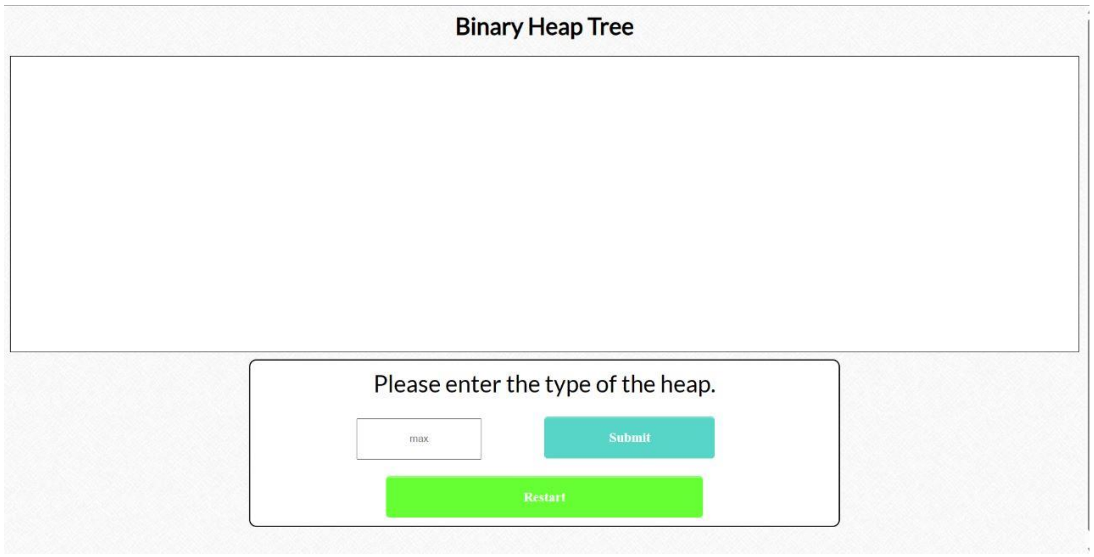
.

Why we use Java script: JavaScript is a popular choice for implementing the Max Heap Visualization Project due to its versatility, widespread adoption, and extensive support for web-based applications. JavaScript is a client-side scripting language that runs directly in web browsers, making it an ideal choice for creating interactive and dynamic visualizations. Its wide adoption ensures compatibility across different devices and browsers, enabling users to access the min heap visualization project without any additional installations.

Project Functionalities and Feature

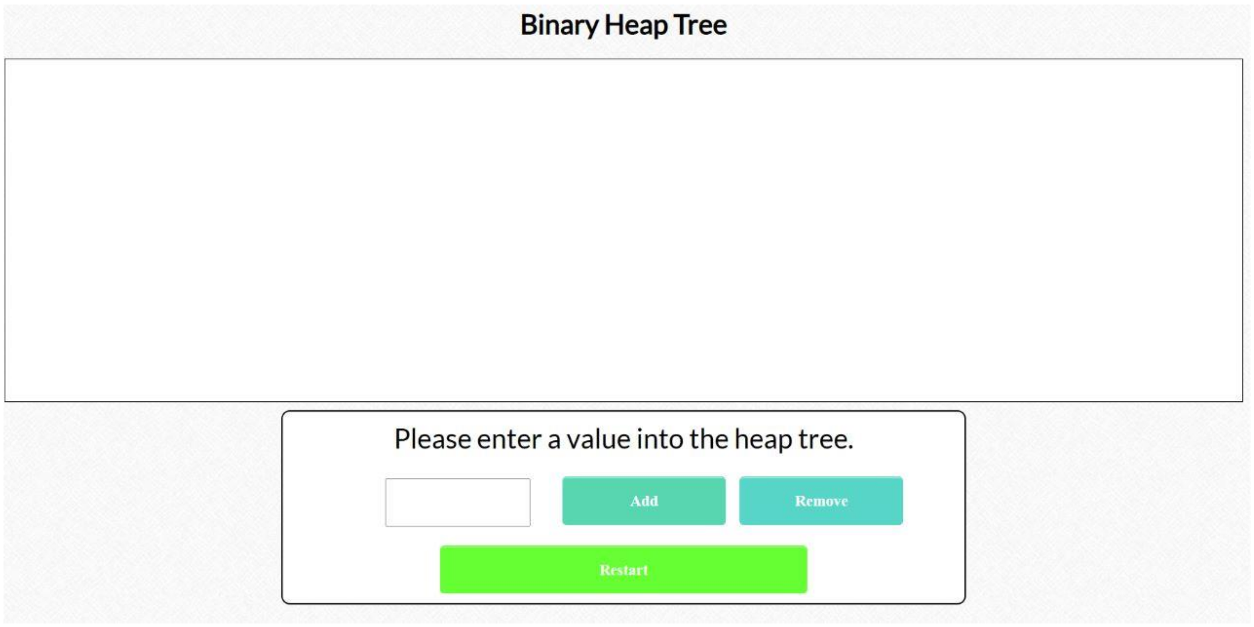
The project provides an interactive visualization of the max heap data structure, allowing users to explore and manipulate the heap in real-time and offers a step-by-step mode that guides users through the construction and modification of the max heap. These visual effects provide a dynamic and engaging experience, making it easier to comprehend the heap's operations and changes.

Homepage: When we first run our program. And show the homepage



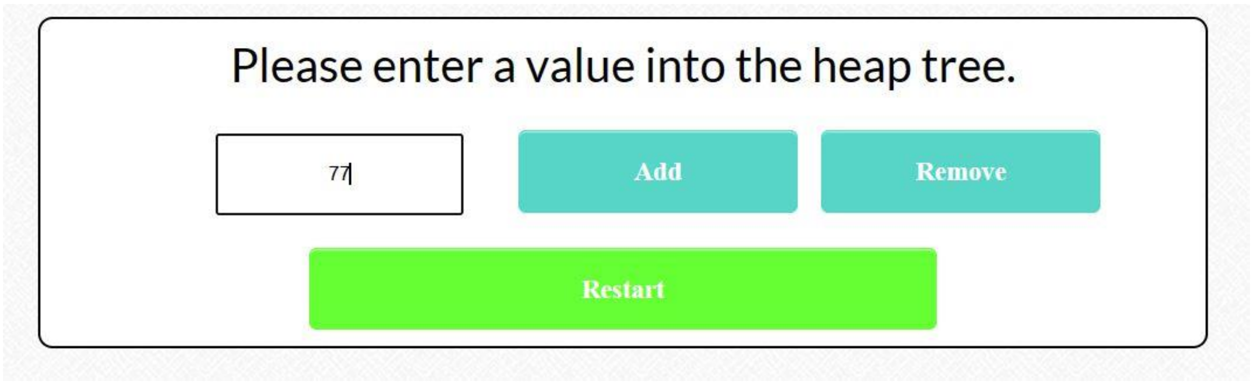
This Homepage simply contains a navigation bar with two buttons that are home and binary heap. To visualize the algorithm we need to go to the max option by clicking the submit button. (Clicking the restart button will start again)

Binary Heap Page: Binary heap page mainly insert value and show the tree.



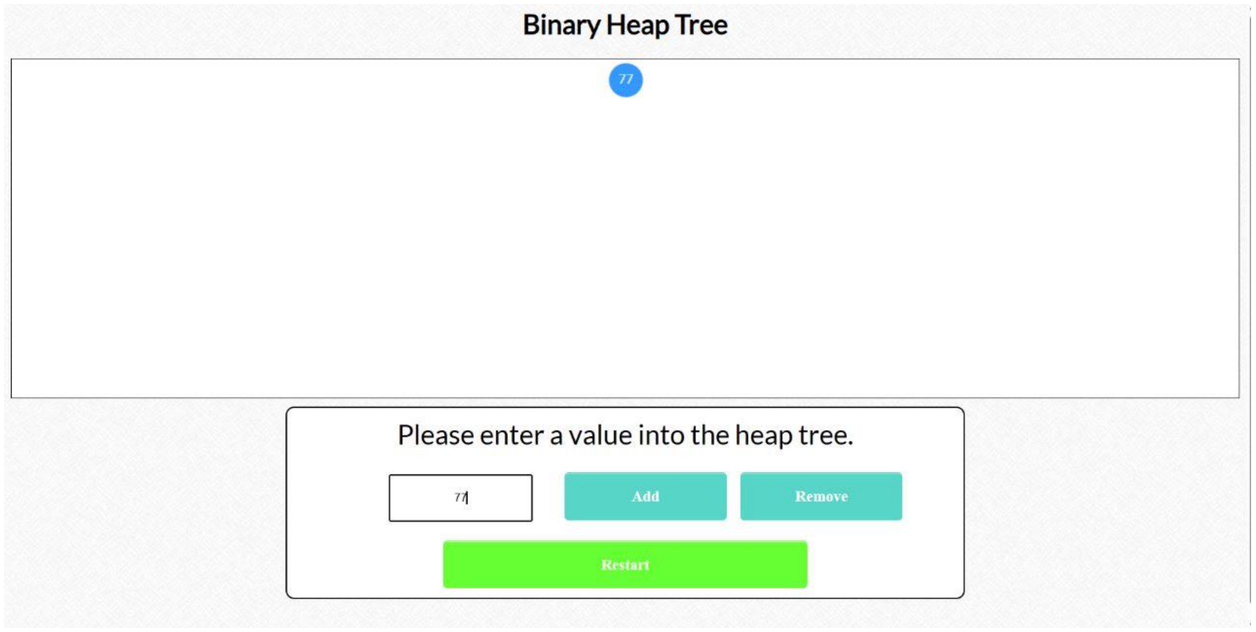
Blank bar to insert any random data

Example:

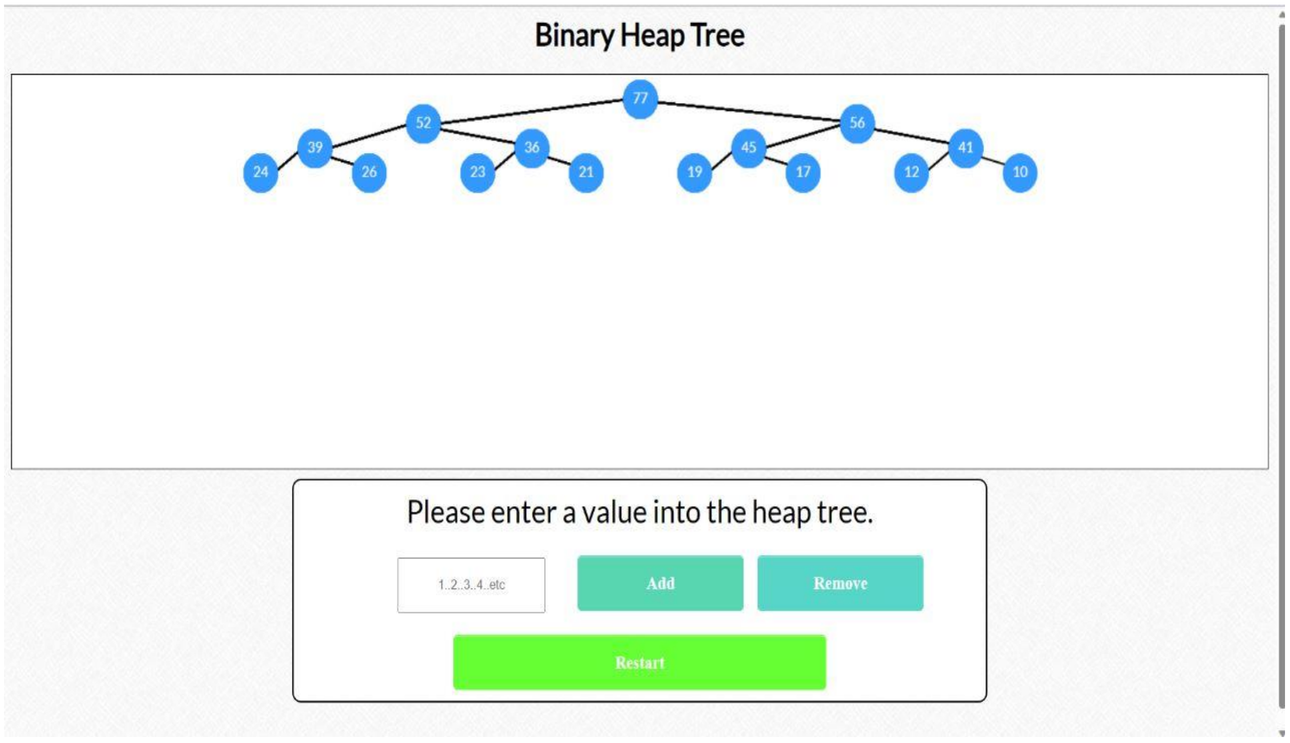


Value insert the click add button. Then the value will be added to the tree

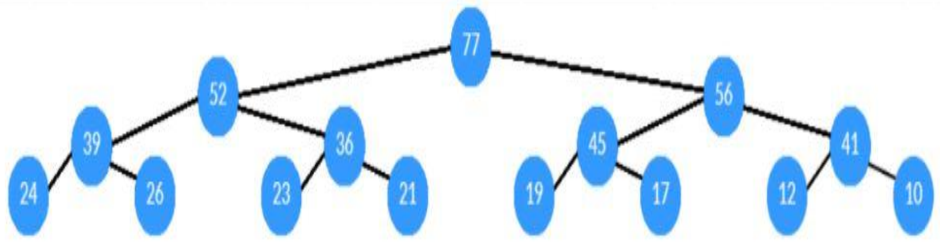
Example:



Similarly a tree can be formed by value insert and add. Then final result show the tree.



Max Heap Visualization



Conclusion

In conclusion, the Max Heap Visualization Project is a valuable tool for understanding and exploring the intricacies of min heaps. By providing an interactive and visually engaging platform, this project facilitates a comprehensive learning experience for users. The visualization enables users to observe the construction, manipulation, and operations of min heaps in real time, enhancing their understanding of the underlying algorithms and data structures.

Through the project's features such as interactive visualization, step-by-step demonstration, and animation, users can grasp the fundamental concepts of min heaps, including heap construction, element insertion, minimum element retrieval, element deletion, and heapify operations. Visualization brings these abstract concepts to life, making them more tangible and easier to comprehend.

Special Thanks to:
Md. Shymon Islam
Lecturer
Department Of CSE
North Western University, Khulna

Developed by:

Shahariar Alam Alif

ID:20212030010

Ritom Sascha Nag

ID:20212038010

Animesh Roy

ID:20212052010