Binary Search Tree(**BST**)
Visualization

User manual

Date:16-05-2023
North Western University
Khulna ,Bangladesh

Table Of Contents

Introduction :

A binary search tree (BST) is a data structure that organizes elements in a hierarchical manner for efficient searching, insertion, and deletion operations. It is a type of binary tree in which each node has at most two child nodes, referred to as the left child and the right child.

The binary search tree follows a specific ordering property: for any node, all elements in its left subtree are smaller than the node, and all elements in its right subtree are greater than the node. This property allows for efficient searching and sorting of elements.

To visualize a binary search tree, we typically represent it graphically. Each node in the tree is represented by a circle or a box, and the edges connecting the nodes represent the parent-child relationship. The topmost node in the tree is called the root.

Here's an example of a binary search tree visualization:

markdown

```
        8
       / \
      3   10
     / \    \
    1   6    14
       / \   /
      4   7 13
```

In this visualization, the root node is 8. The left child of the root is 3, and its right child is 10. The left child of 3 is 1, and its right child is 6. The right child of 10 is 14, and so on.

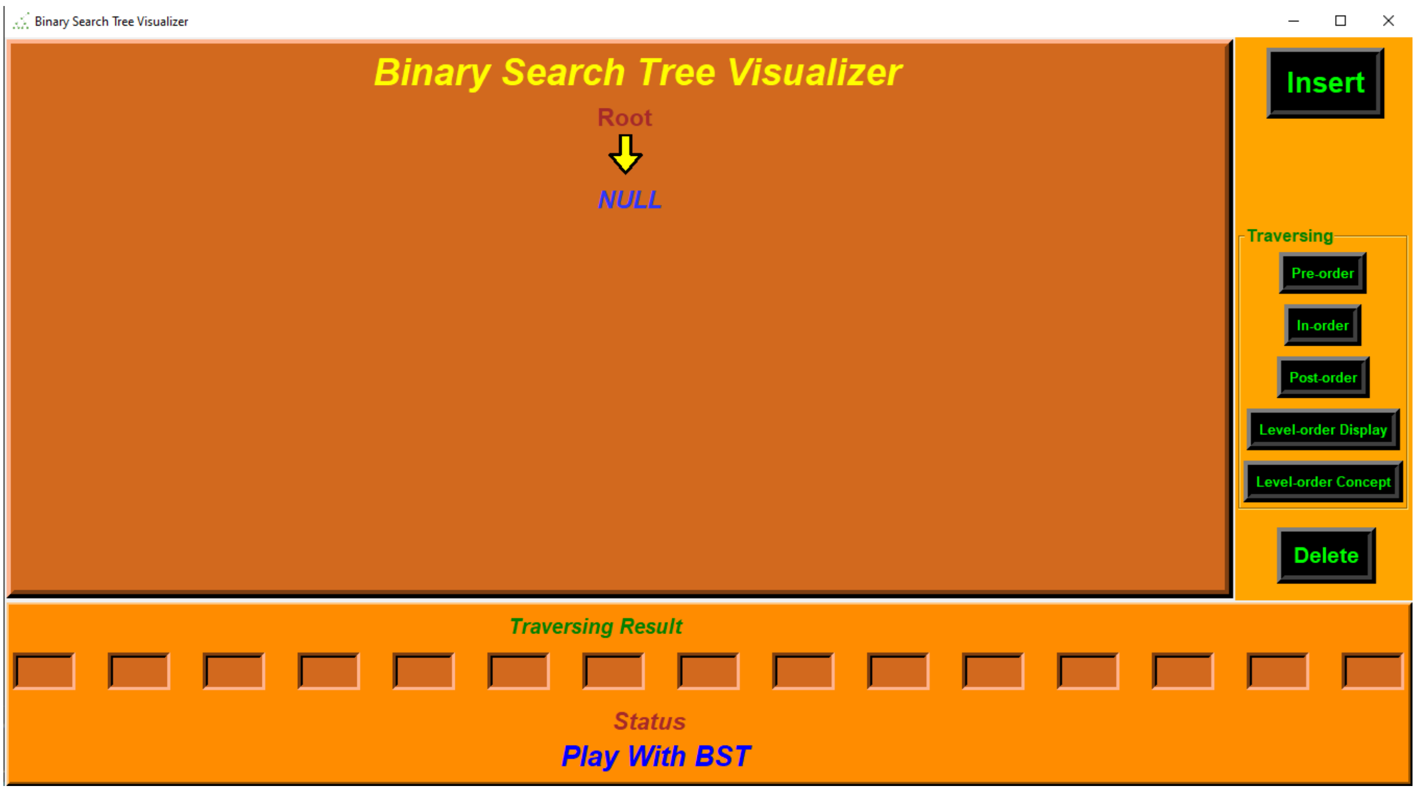
The visualization of a binary search tree helps us understand its structure and the relationship between the nodes. It enables us to visually analyze and reason about the tree's properties and perform operations such as searching for a specific element, inserting a new element, or deleting an existing element.

By maintaining the ordering property of the binary search tree, we can efficiently perform various operations and achieve logarithmic time complexity, making it a valuable data structure in many applications.

Description

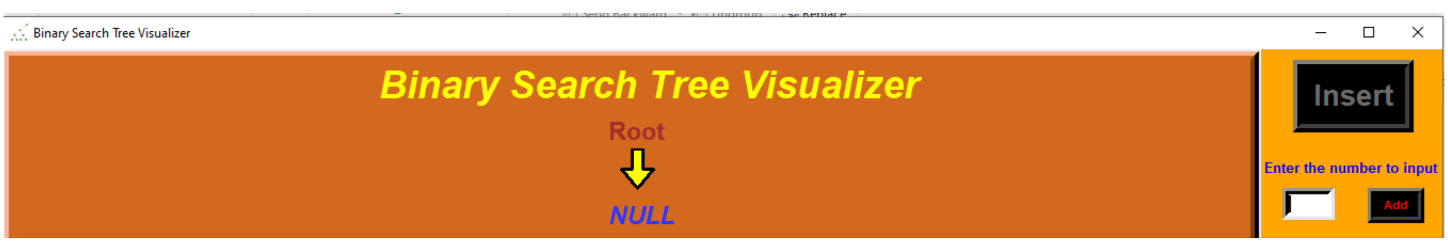
Landing Page:

This is the landing page of our program when we first run the code.



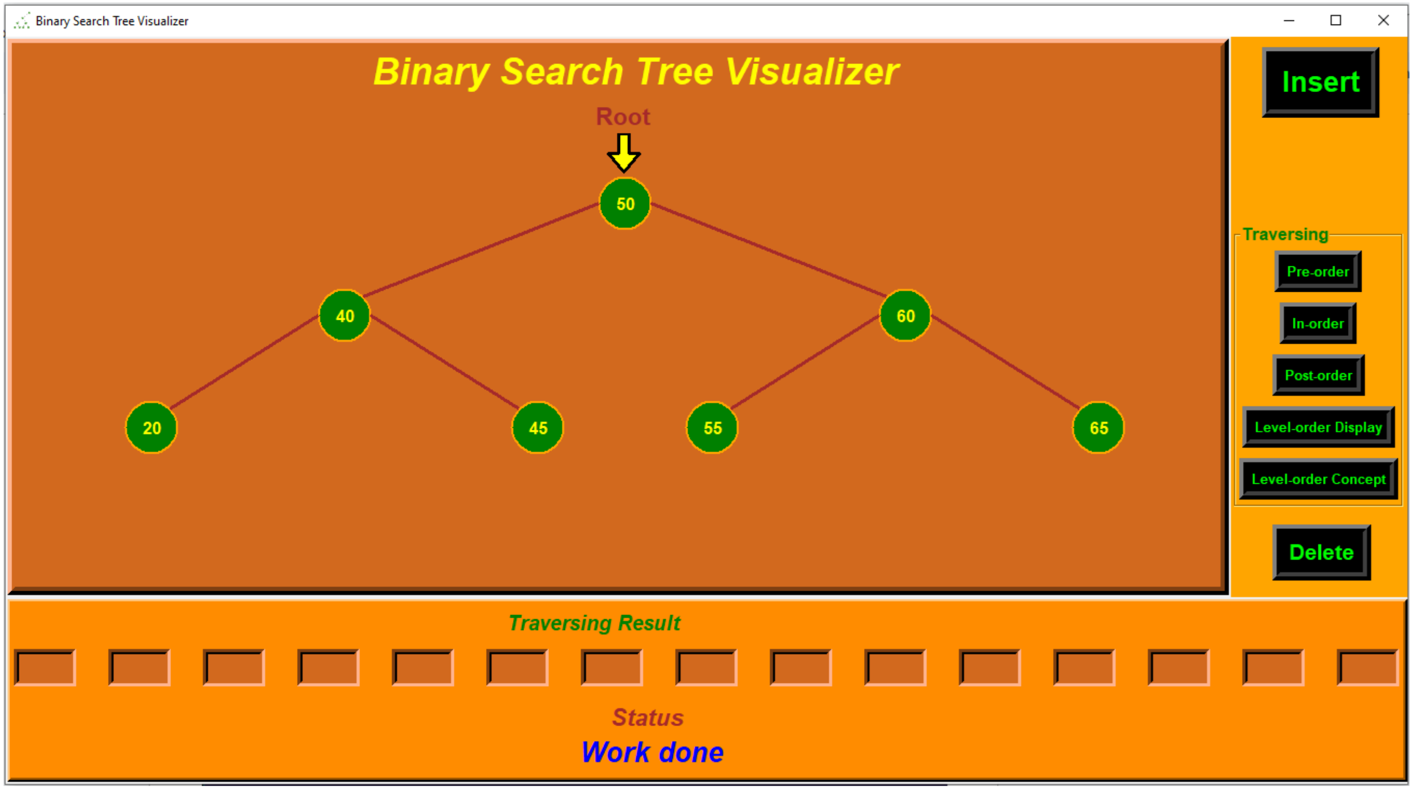
Input Section:

In the input field we can take user input with a space separating them.

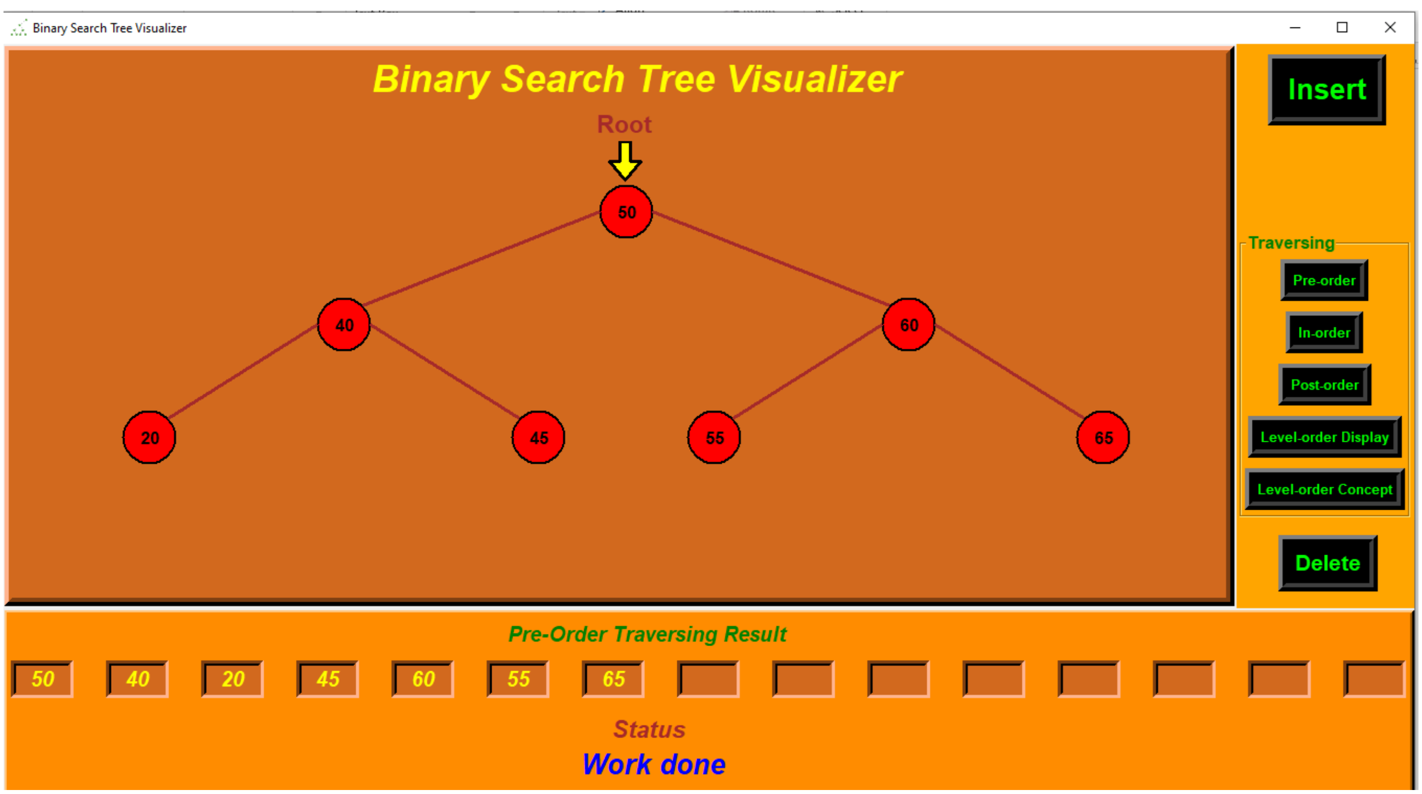


Generate Data :

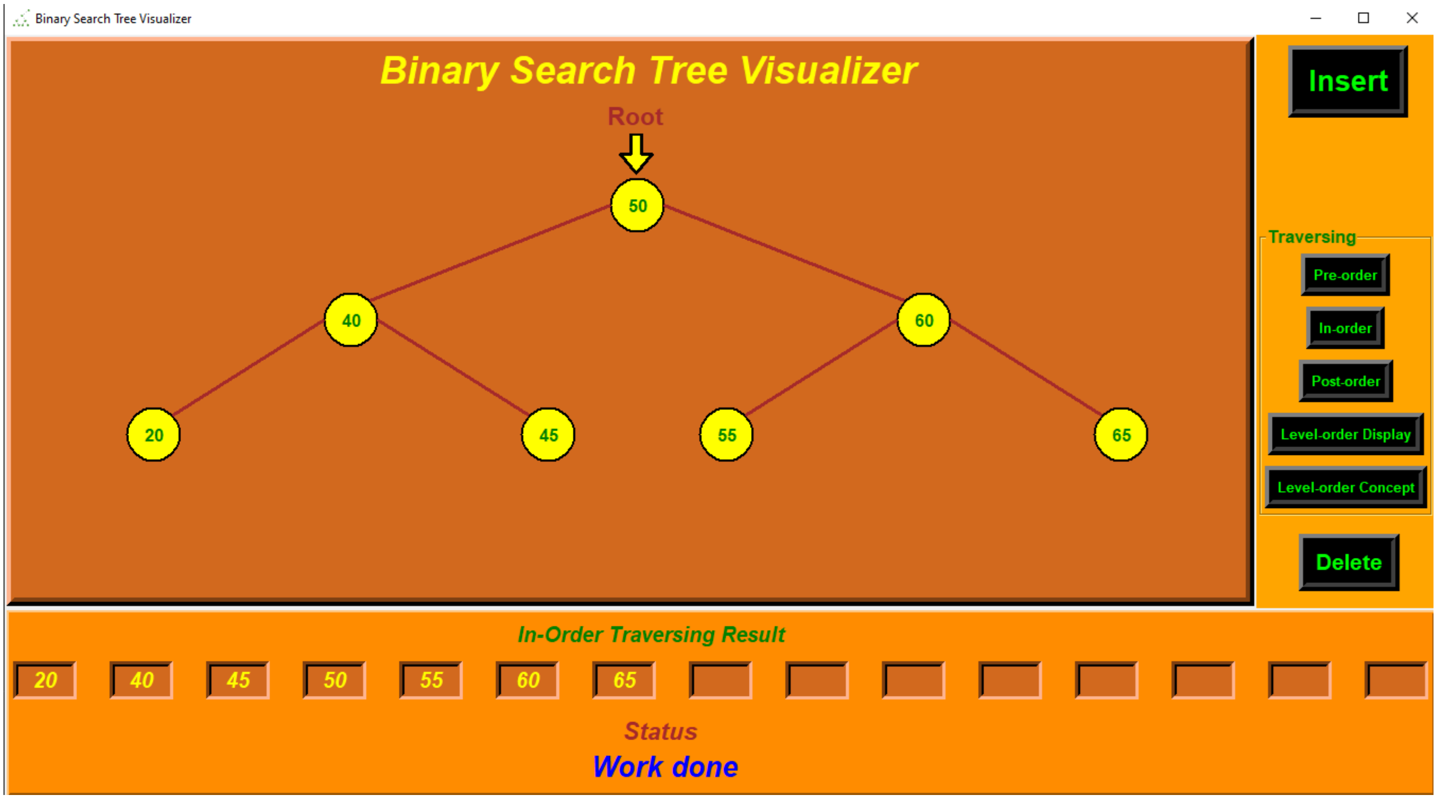
After giving the user input we press generate button to generate the data into canvas.



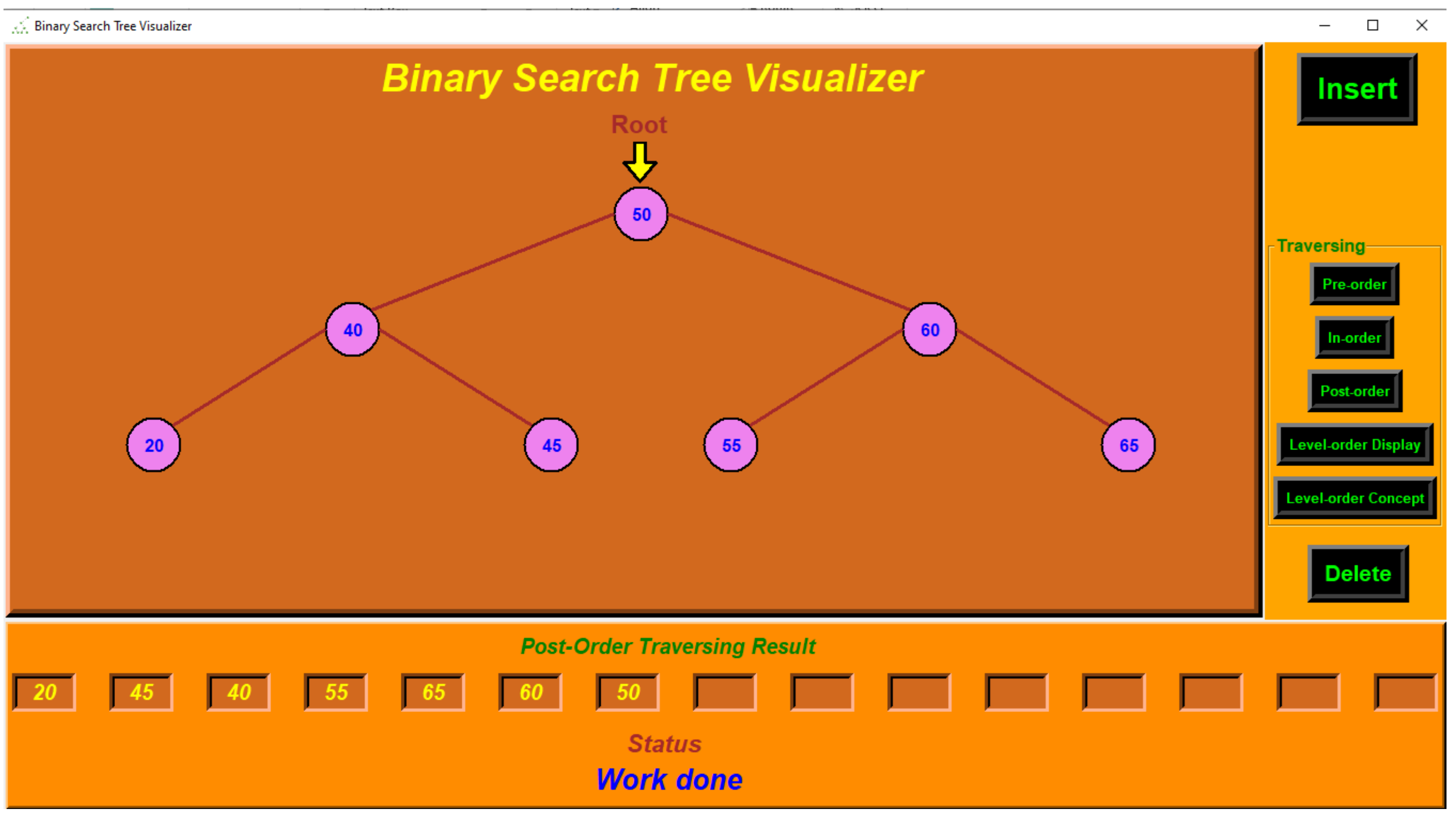
Pre-order: Left Root Right.



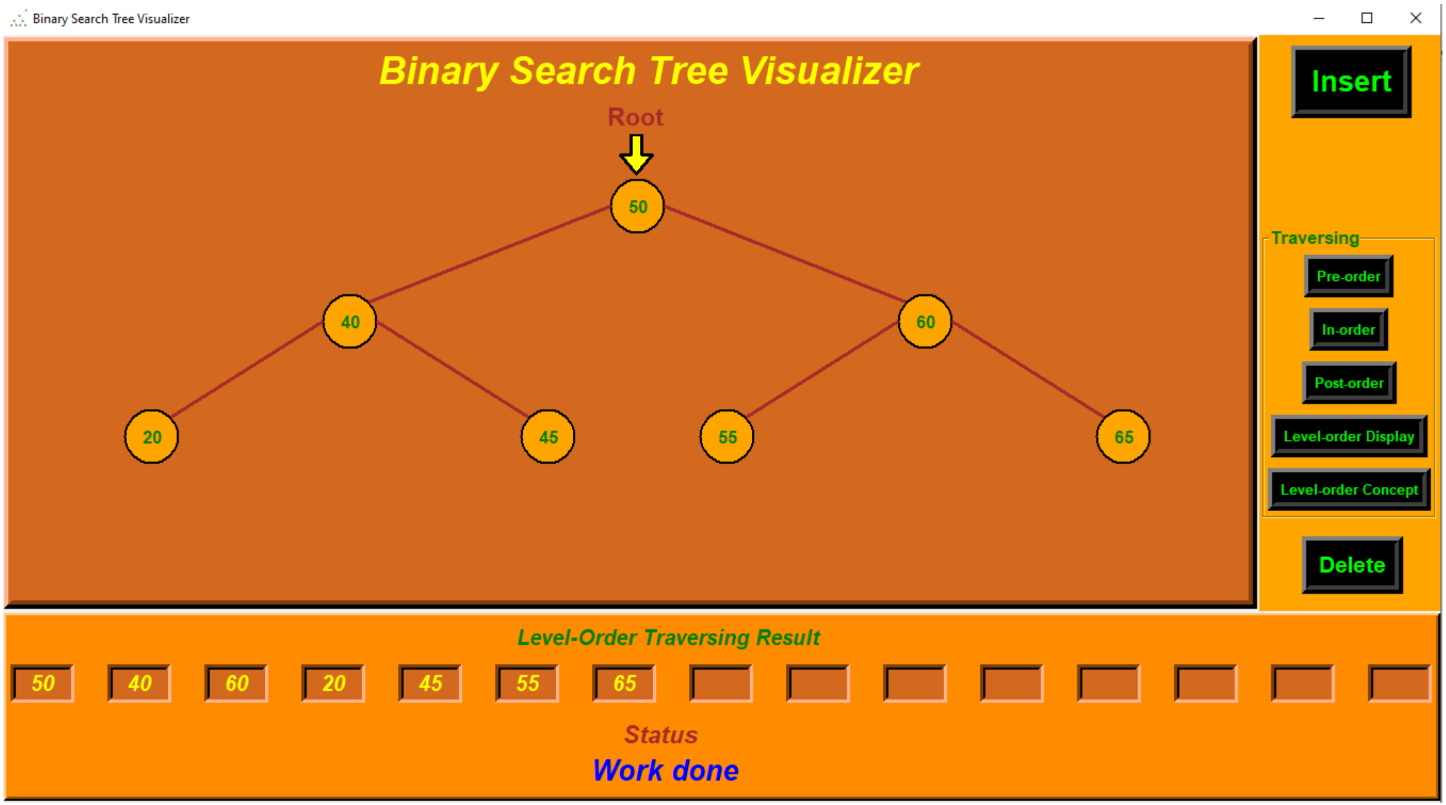
In-order: Root Left Right.



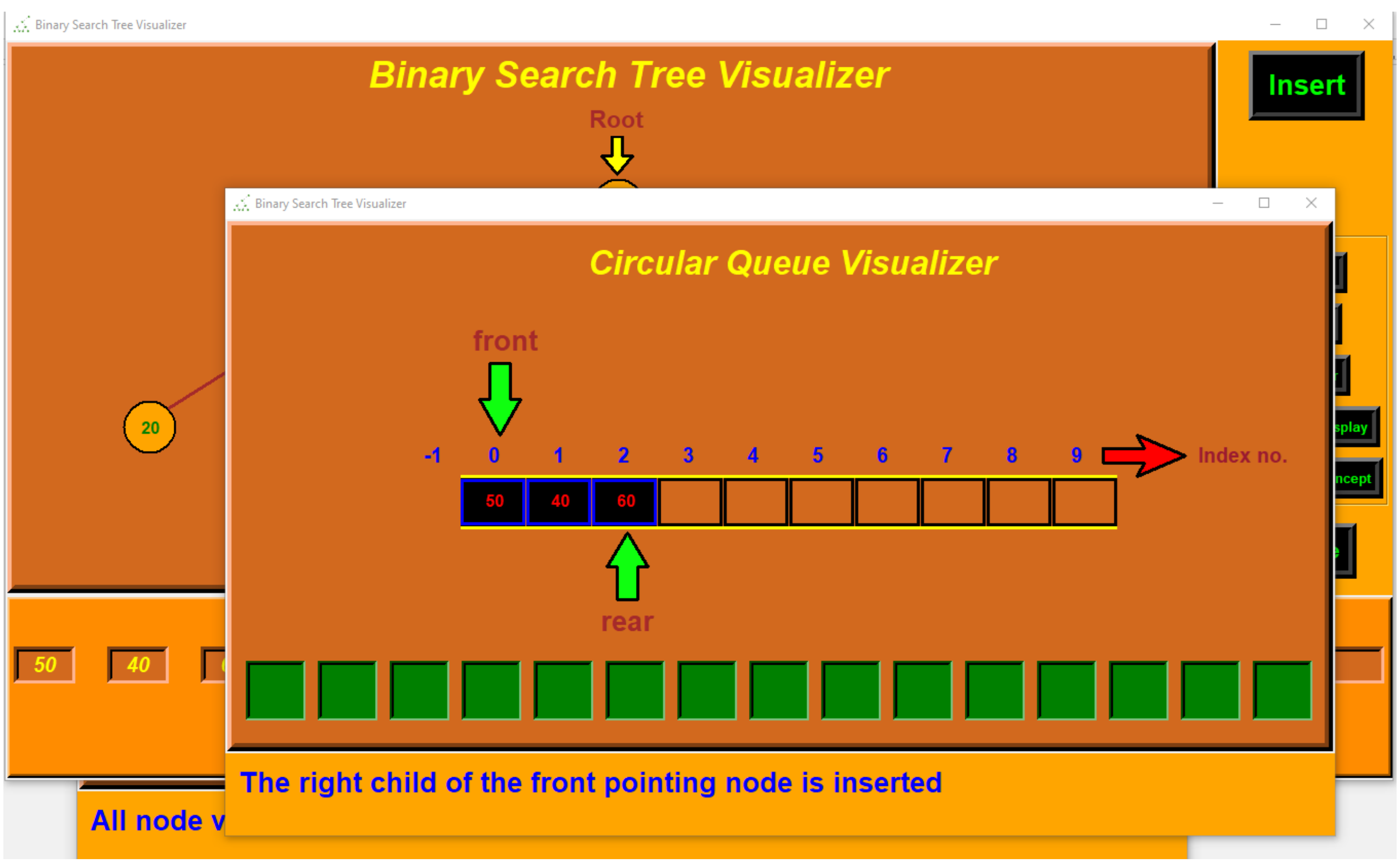
Post-order: Left Right Root.



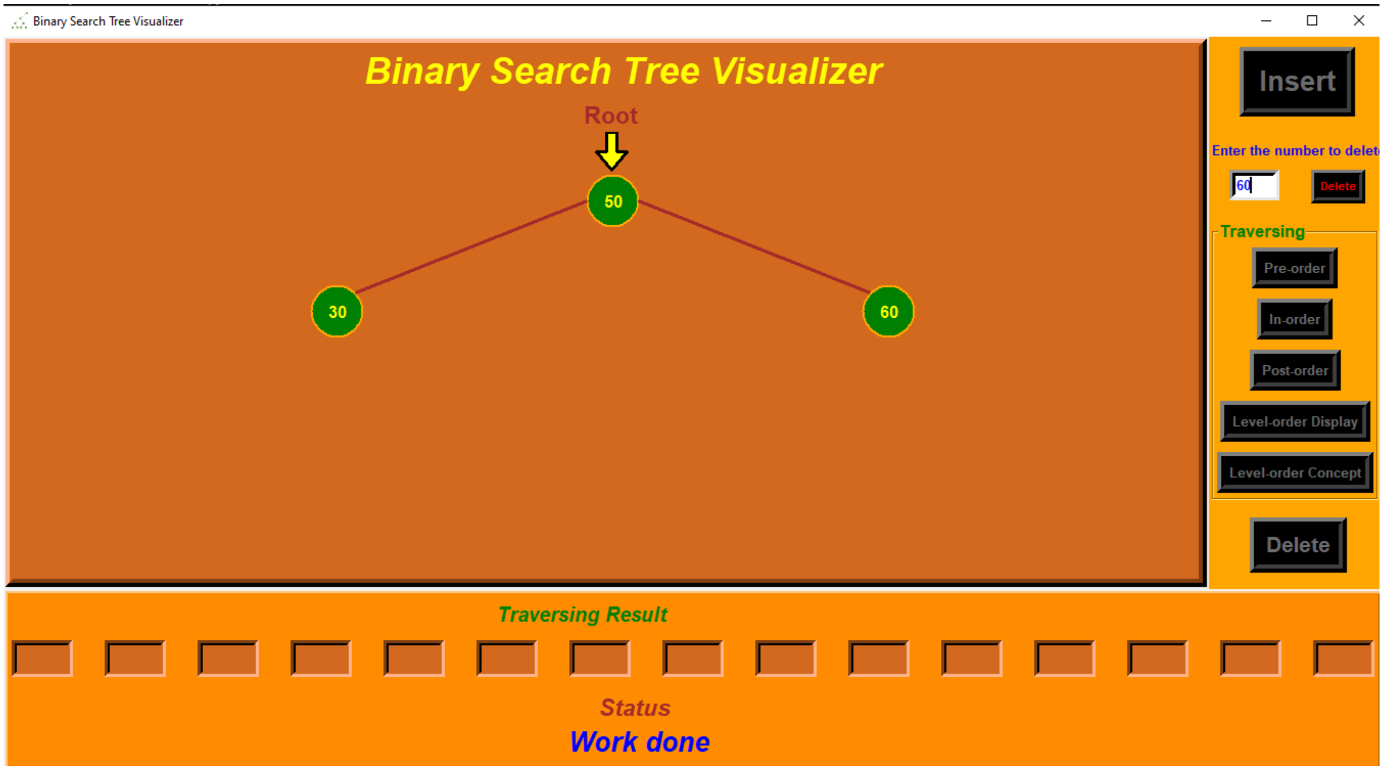
Level-order display:

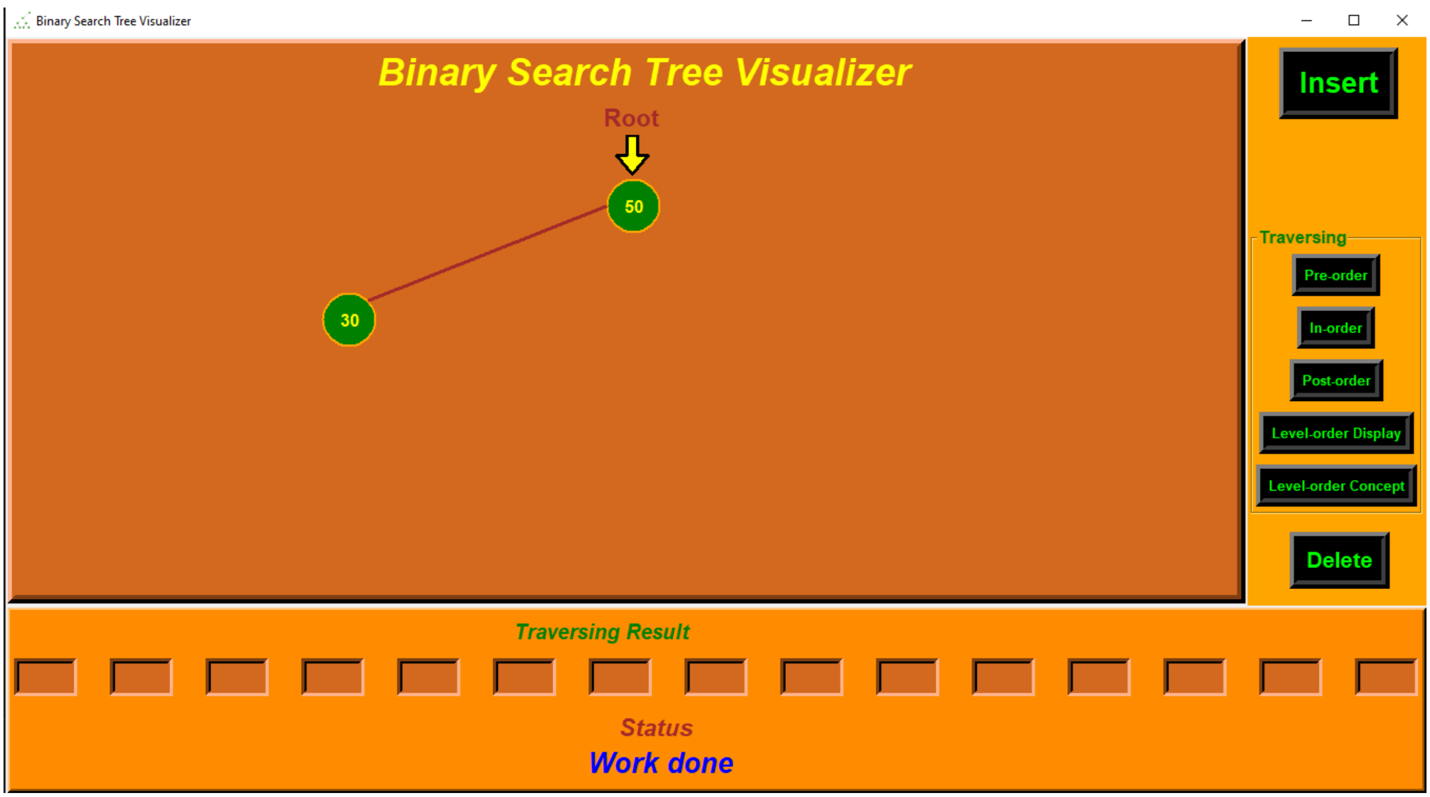


Level-order Concept:



Delete Root: If we want to delete a root, we can delete it by inserting the delete root.





Conclusion:

Binary search tree visualization is a powerful tool that helps understand the structure and behavior of binary search trees. By representing nodes and their relationships graphically, it provides a clear and intuitive way to analyze and manipulate the tree.

Through visualization, we can observe how nodes are organized in a binary search tree, with each node having a key and two child nodes (left and right). The tree's properties, such as the left subtree containing only smaller keys and the right subtree containing only larger keys, become apparent in the visual representation.

Visualizing a binary search tree also allows us to observe the tree's balance. A well-balanced tree ensures efficient operations, such as searching for a specific key, inserting a new node, or deleting a node. Imbalanced trees may lead to performance issues, as the operations may take longer to execute.

Moreover, visualizing a binary search tree can aid in detecting and resolving potential issues, such as duplicate keys or incorrect ordering of nodes. By identifying such problems visually, we can make necessary adjustments and maintain the integrity and correctness of the tree.

In conclusion, binary search tree visualization is a valuable tool for understanding and analyzing the structure and behavior of binary search trees. It provides a visual representation that helps comprehend the relationships between nodes, the tree's balance, and potential issues. By utilizing visualization techniques, we can improve our understanding and make informed decisions when working with binary search trees.

Special Thanks to

Md. Shymon Islam
Lecturer Department Of CSE
North Western University Khulna, Bangladesh

Developed by

. Sourav Biswas
ID:20212025010
.Md Toufik Ahmmed
ID :20212022010
.Effar Ara Eva
ID:20212010010