Wilson Quilli

Professor Elangovan

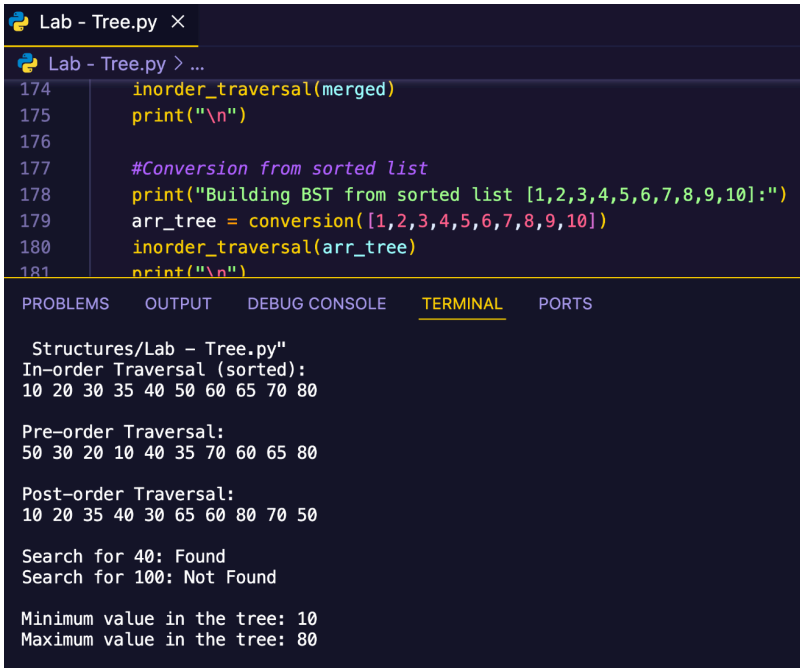
CMPSC 412: Data Structures Lab

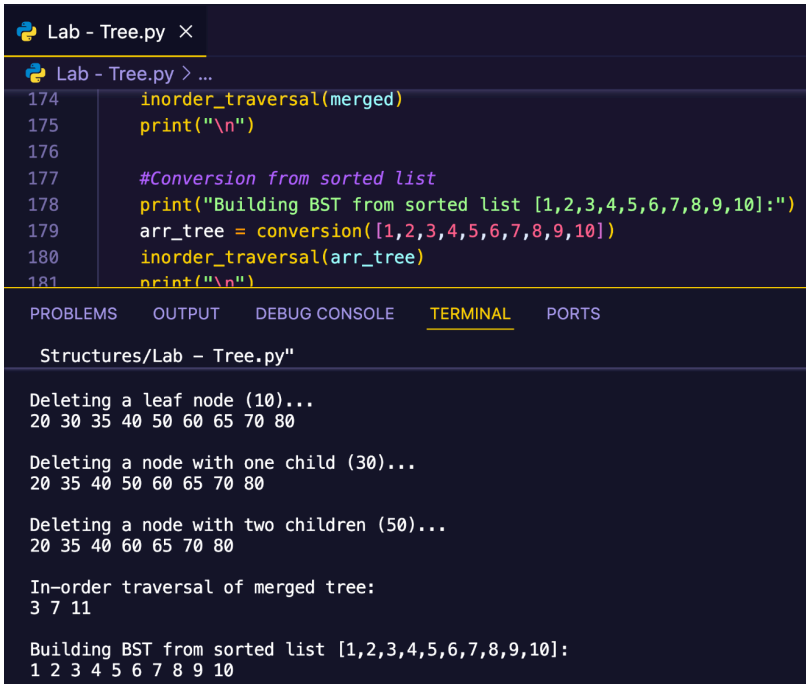
October 7th, 2025

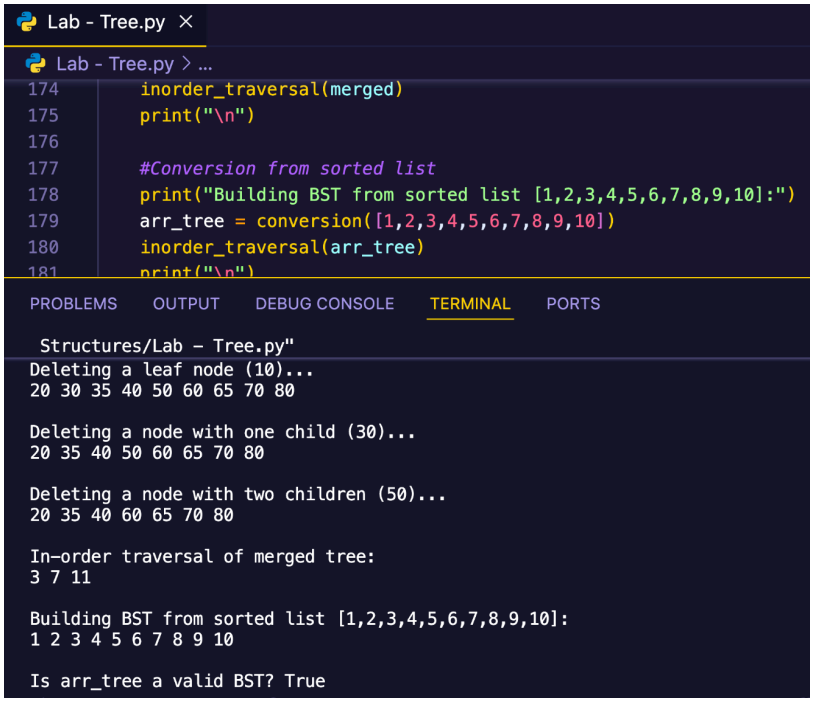
Lab: Trees

In this lab, I completed four exercises, using trees, more specifically, the Binary Search Tree. The first exercise I implemented the basic Tree operations, inserting a node, performing in-order traversal, performing pre-order traversal, performing, post-order traversal, finding a node, and finding the maximum and minimum value in the whole tree. The second exercise I implemented a function to remove a node from the tree, while keeping in mind the 3 cases of events that can occur when removing a node, removing a leaf node, removing a node with a child, and removing a node with two children. The time complexity of this operation is O(h), where h is the height of the tree. I think it's O(h), because the tree's height matters when searching for the node to delete and considering the three different cases. The third exercise I merged two trees together. The time complexity of this operation is O(n), where n is the size or amount of the nodes in the smaller tree, when performing the merge. The final exercise, I converted a list of elements into a Binary Search Tree. This also validates and checks if the tree is a Binary Search Tree by following the BST property of each node having two children, with every node in the left being less than its parent and every node in the right being more than its parent. Ultimately, from this exercise, I learned more about the operations of trees and in terms of time complexity, the size and amount of elements matter.

Screenshots:






Lab - Tree.py
Lab - Tree.py > ...
174 inorder_traversal(merged)
175 print("\n")
176
177 #Conversion from sorted list
178 print("Building BST from sorted list [1,2,3,4,5,6,7,8,9,10]:")
179 arr_tree = conversion([1,2,3,4,5,6,7,8,9,10])
180 inorder_traversal(arr_tree)
181 print("\n")
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Structures/Lab - Tree.py"
Deleting a leaf node (10)...
20 30 35 40 50 60 65 70 80
Deleting a node with one child (30)...
20 35 40 50 60 65 70 80
Deleting a node with two children (50)...
20 35 40 60 65 70 80
In-order traversal of merged tree:
3 7 11
Building BST from sorted list [1,2,3,4,5,6,7,8,9,10]:
1 2 3 4 5 6 7 8 9 10
Is arr_tree a valid BST? True

Works Cited

GeeksforGeeks. (2025, September 24). *Binary Search Tree*. GeeksforGeeks. https://www.geeksforgeeks.org/dsa/binary-search-tree-data-structure/

GeeksforGeeks. (n.d.). Complexity of different operations in binary tree, binary search tree, AVL tree. *GeeksforGeeks*. https://www.geeksforgeeks.org/dsa/complexity-different-operations-binary-tree-binary-search-tree-avl-tree/