Lab Report 06

*<<Damian Sclafani>>*

# Problem

State the given problem clearly in one’s own words. Copy and pasting from the assignment will result in an overall 0 for the entire lab report.

We have to make a program that will read a file of fruits (each piece of data has a name and weight), put the data into a binary tree, and be able to print the binary tree in preorder, inorder, and postorder. Also, it needs to be able to remove objects off of the tree.

# Solution

Explain how the program solves the problem. This description must be detailed and high-level without using the direct implementation – do not just copy and paste the programming solution’s code. You may think of this as explaining how the software works to another non-computer entity, like a human. It should cover the finer points of the lab while justifying implementation decisions. While pseudocode may be used it must be accompanied by said clear and understandable description. Flow Charts and graphics are strongly encouraged, and in some cases required.

1. The user enters the file they want scanned in
2. The program reads the file and sets each piece of data as a Fruit inside of an array
3. The program puts all data in the array into the binary tree
4. The program prints out the binary tree preorder
5. The program prints out the binary tree inorder
6. The program prints out the binary tree postorder
7. The user enters the value they want removed
8. The program searches for the value and removes it, swapping another object in the binary tree in its place
9. The program prints out the resulting binary tree inorder

# Implementation Problems Encountered

Enumerate the issues that arose from creating this solution. Include major syntax, run-time, and logical errors with their respective solutions. If you did not have any problems then you may put, “No problems encountered”, but if the solution is not correct then this section will receive no points.

The only problem I ran into was figuring out how to write the postorder method.

# Lab Report Questions

1. Describe the difference between a self-balancing tree and a non-self-balancing tree. A self-balancing tree will make sure that the binary tree is always balanced according to the numbers given. The non-self-balancing tree does not do this.
2. What is the Big O complexity for searching in a balanced binary search tree? O(lg(n))
3. What is the Big O complexity for searching in a non-balanced binary search tree? O(n)