Name: David Southwell

Lab Number: Binary Tree MiniLab

Lab Title: Binary Trees

Date: 21 May 2017

Before the Lab:

What I expect to learn: How to traverse binary trees and implement algorithms to do so

Concepts I think the Lab explores: Binary trees

Java classes I expect to modify: BinaryTreeLab, BinaryTree, LLQueue

Where I think there may be problems: utilizing the LLQueue class in this minilab.

After the Lab:

What I actually learned: How to traverse binary trees in different ways and how the Visitor class works in conjunction with BinaryTree to print the values of the tree.

How well the Lab allowed me to explore the target concepts: Very well, but I had trouble understanding recursion in terms of the DFS method for awhile.

What problems I encountered (or none) and how I handled them: I had trouble implementing recursion in the DFS method, but I eventually simplified my code enough to understand it.

Things in the write up that confused me or were especially helpful: The hints about using recursion were helpful, but it still took me awhile to completely understand what the pseudocode meant.

Questions:

1. An empty binary tree’s attributes include a placeholder for the data (root) and its left and right children.
2. The constructor does match my expectations.
3. Breadth-first order means adding elements in a top to bottom, and left to right order.
4. Yes the results are different from the BFS, and it was the preOrder DFS like I had expected