Approach Document Double Threaded Binary Tree

# Assignment Objective

Implement the double-threaded binary tree algorithm correctly and efficiently

* Apply common software data structures to applications.
* Apply a biblical worldview to data structures and algorithms.
* Understand the concept of Binary Trees
* Understanding the concept of double-threaded binary trees
* Understanding what the use case of bit fields are
* Understanding how recursion is used in binary trees

# Assignment Requirements

The functions I need to create:

Within the BSTNode header file:

* Must add bit fields
* Add setters and getters for those created bit fields

Within the BST header file:

* Rewrite insertHelp() 🡪 Helps insert values into the BST
* Create findPrior() 🡪 Helper function
* Create findNext() 🡪 Helper function
* Modify printHelp() so it works with the threaded nodes
* Create printInorder() 🡪 to do an in-order printing of the tree
* Create printReverse() 🡪 Prints the reverse order of the tree

Within main:

* Insert the values given within the BST
* Print out the size of the BST
* Print out the tree structure
* Print out the inOrder of the tree
* Print out the reverseOrder of the tree

# Approach

* Review the lab requirements
* Search for websites to help
* Watch YouTube videos to understand what is happening
* Set up the approach document
* Review lecture slides and read the book
* Tackle each function one at a time with the main influence on the inserthelp() function
* Go to tutoring if I get stuck

# Build Log

09/19/23: Today, I will look through the files I am given to try to understand what is going on within each file. I will watch YouTube videos and scroll through various websites to try and understand the concept of Double Threaded Binary trees. On top of that, I will be looking at the requirements within the programming assignment 2 Microsoft Word document and the PowerPoint to help me begin thinking about how I want to best approach this lab. To start things off, I will simply add the bitfields to the BSTNode class. In addition to adding the bit fields themselves, I will also add getters and setters for the left bit and right bit respectively, and once I know I have those running I will be finished for the day.

09/21/23: Today (09/19/23), I went to tutoring, and after talking with my tutor a big help will be looking at the Java code given within the programming assignment 2. That Java code should be a great outline for at least three of the functions that I am required to do. On top of that, that website should be a great outline to look at for understanding double-threaded binary trees a little more in-depth, as I am currently struggling to understand them in application. The current functions that I will try and hopefully accomplish are insertHelp(), findPrior(), and findNext(), as this should be very similar to findPrior(), all of these will be implemented within the BST header file.

09/25/23: I will (hopefully) be finishing my assignment today, at least get the program running and iron out any bugs tomorrow. So, the last functions I will work on are printInorder(), printReverse(), and modifying printhelp(), all within the BST header file. Hopefully, if all goes according to plan, then I should be able to finish by today. Also, looking at the Java code as a template to follow should allow me to execute my plan effectively and efficiently. Afterwards, I will create the BST object in main to insert all the values given in the “Programming Assignment 2” document and then print out the values as the tree and both in-order and in-reverse-order.

09/26/23: I have completed the functions that I wanted to do yesterday; however, they are not printing exactly as intended, and I must do a little debugging to see what exactly is going wrong. However, I think I have an idea of what to change.

09/29/23: Final Update: The lab is complete and is fully functional to what was listed in the Programming Assignment 2 Microsoft Word Document.

Each function implementation location:

BST.h: findNext() (Line 288), findPrior() (Line 268), insertHelp() (Line 143), printHelp() (Line 250), printReverse() (Line 111), printInOrder() (Line 97),

BSTNode.h: Created bit fields (Line 22 & 23), setters (Line 40, 41) getters (Line 42, 43)