**Approach**

* Gather a better understanding on what a Double-Threaded Binary Tree is.
* Look at current BST files and get working, understand what is happening with the currently implemented BST from author.

We discovered that we were given a normal binary search tree. The given BST already has the existing functions we needed. We just needed to modify them.

* Determine what objects must be modified to create a threaded-BST.
* Work on BSTNode.h first

Add Boolean fields to indicate whether a node pointer is a thread or a regular pointer. DO NOT ADD ADDITIONAL POINTERS TO BSTnode.

We ended up using bitfields instead of Boolean fields discovering they work almost the same way. It was an easy change.

Add or modify setter/getter methods to access Boolean instance variables.

We created set and get methods based off of our bitfields, lpType and rpType.

* BST.h

Rewrite inserthelp() to take advantage of modified BSTNode where a pointer can be either regular pointer or a thread.

We utilized our bitfields lpType and rpType to help with this, when we used setLeft we had to utilize also root->left() in our BSTnode constructor. The right side was similar.

Modify printhelp() to work with threaded nodes.

Printhelp() was difficult to grasp what the Threads were doing. However, using our bitfields, it made this a breeze and we were able to keep most of the functionality of the already given printhelp() function.

Add printInorder() to do in order printing of tree WITHOUT recursion.

printInorder() wasn’t too bad, we had a good understanding of how the inorder traversal works, just had to modify it to work with threads using bitfields.

Add printReverse() to do reverse order printing of tree WITHOUT recursion. printReverse() was nearly identical to printInorder(), just changing lefts to rights and vice versa.

ADDITIONAL NOTES:

Could utilize helper functions to create methods above.

* Use following <int, string> pairs to build tree.
  + 77, “seventy-seven”
  + 70, "seventy"
  + 75, "seventy-five"
  + 66, "sixty-six"
  + 79, "seventy-nine"
  + 68, "sixty-eight"
  + 67, "sixty-seven"
  + 69, "sixty-nine"
  + 90, "ninety"
  + 85, "eighty-five"
  + 83, "eighty-three"
  + 87, "eighty-seven"
  + 65, “sixty-five”
* No smart pointers.