Lab 16 Guide

Topics of the Day:

- Search in Collection
- Binary Search
- Binary Search Trees

Exercises

BST Methods

Students must implement the BST class which extends BinaryTree. They must implement methods "contains" and "add". The methods can be written both iteratively and recursively. The recursive approach seems more intuitive unless the student is just struggling with recursion in general. If a student is trying the recursive approach, make sure they have a solid base case.
Common mistakes will be iterating until currNode.myNext == null when it should be currNode == null. Some students may also check compareTo values in an opposite manner or check for equality to -1 instead of < 0.

BST Iterator

Exercise: Create an Iterator (poorly written skeleton provided) that uses a nodeStack to iterate through items. next() is the hard part. Students should know that next() returns the top node in the stack, and they need to update the stack such that the next smallest value is on top. For the constructor of the iterator, you can walk them through pushing the "smallest" node into the stack. Try having them draw out a sample tree and decide on what order elements should be pushed in.

BST Constructor

Students must make a constructor which takes in 2 ArrayList inputs. The first input represents the items in preorder. The second input represents the items in inorder. Due to the given inorder, compareTo should not be used to determine which item is greater.
The biggest pitfall is getting started. Have students try small cases and see how the tree should be developing. This problem can be solved iteratively and recursively. Encourage students to try both approaches.