Programming Project - Search Trees

Part 1: Complete the implementation of the LinkedBinarySearchTree class from the jsjf package presented in this chapter. Specifically, complete the implementation of the removeMax, findMin, findMax, find, getLeft, & getRight methods.

Part 2: Create a new sorting method called bstSort which sorts an array of elements **using a binary search tree** to order the elements. Add your method do the Sorting class from earlier this semester.

Part 3: Create a class called LinkedBSTOrderedSet that implements the methods in the OrderedSet interface. Note that a LinkedBSTOrderedSet is nearly identical to a regular LinkedBinarySearchTree - the only difference is that duplicate elements are not added to a list.

Part 4: Create a driver called **Project8.java** to test your implementation. Sort 2 different arrays of at least 20 **randomly generated** numbers using bstSort(). Display the original & sorted array and the minimum & maximum elements. Then, add 15 random numbers between 1 and 20 to a LinkedBSTOrderedSet. Print a list of the unique numbers generated and the number of unique numbers.

Submission information:

1. Include your name as a comment at the top of each source code file
2. Create a document which *briefly* describes your implementation. Include a copy of your bstSort, find, findMin, and removeMax methods from LinkedBinarySearchTree and the add() method from LinkedBSTOrderedSet. Also include sample output. This document will make up a substantial portion of your grade.
3. Zip your source code (.java files) or your entire project. Ensure that you include all source code require to run your program.
4. Include your first and last name in the .zip filename
5. Upload your zip file & document **separately** to Canvas.

Make sure you follow each of the submission instructions. A minimum of 10 points will be deducted for any missing/incorrect submission information.

Be prepared to discuss and demo your project in class following the due date.