Sorting Algorithm Comparison

This semester we will be learning a total of NINE sorting algorithms. You should already know four (heap sort, tree sort, bubble sort, selection sort). This exercise will encourage you to implement as many sorting algorithms as you can, and then to race them against each other to see how they perform. **This exercise is not being submitted or graded.**

1. Checkout the project 132Fall2016Lab13
2. Take a look at the interface called SortingAlgorithm. You will implement several concrete implementations of this interface, each of which represents a different sorting algorithm. (In particular Bubble Sort, and Selection Sort… hopefully more!) Note that we are NOT using generics (T extends Comparable<T>) because later we will want to expand this exercise to include sorting algorithms that are not comparison-based, and which work well with integers.
3. Implement the BubbleSort and SelectionSort algorithms (see the classes that are partially provided.)
4. Test your work by running the JUnit tests (SortTests). (The tests won't actually verify that your algorithms are the correct algorithms, they just check that they do successful sorts.)
5. Test the speed of your algorithms by running the SortRace program!

It would be very valuable to continue working on this exercise later. You could copy your Binary Search Tree project in to this project, and then use it to add a TreeSort class. If you completed the Heap exercise from a previous lab, you could use that to add a HeapSort for this project. As we learn other sorting algorithms this week, try to add those as well.