Directions for Testing the Simple Sorts

First understand the simple sorts; bubble sort, insertion sort, and selection sort. Make sure that you know the algorithms for these sorts. Look at the applets for these sorts and write out the steps for the algorithm on a separate document, or into the spreadsheet used to test the simple sorts. **You will turn in these algorithms for the simple sorts.**

We will write some of the Insertion Sort, Bubble Sort, and Selection Sort algorithms together in class. Make sure that the algorithms you wrote are correct.

We will then look at the code for these 3 simple sorts; “BubbleSortApp.java”, “InsertSortApp.java”, and “SelectSortApp.java”. We will compare it to the algorithms we have written and understand the code as a representation of the algorithms.

Then we will do some timing tests on these sorts. Run the sorts one at a time using “SimpleSortTest.java”. For each sort run it with 100 items, 10,000 items, and 100,000 items. Record the time that it takes to complete the code for each test into the spreadsheet “ComparingSimpleSorts.xlsx”.

Make a line graph that displays the results of your tests.

Now change the for loop in your “SimpleSortTest.java” file so that you are inserting numbers in ascending order. So they are already sorted, instead of being random.

For each sort run it again with 100 items, 10,000 items, and 100,000 items. Record the time that it takes to complete the code for each test into the spreadsheet “ComparingSimpleSorts.xlsx”.

Make a line graph that displays the results of your tests.

Now change the for loop in your “SimpleSortTest.java” file so that you are inserting numbers in descending order. So they are already sorted, but backwards, instead of being random or in order.

For each sort run it again with 100 items, 10,000 items, and 100,000 items. Record the time that it takes to complete the code for each test into the spreadsheet “ComparingSimpleSorts.xlsx”.

Make a line graph that displays the results of your tests.

Now comes the fun part. Think of some intelligent conclusions that you can make about this experiment. Write your conclusions into the spreadsheet “ComparingSimpleSorts.xlsx” on the last page which is named Conclusion. Think about when each sort works well, and when they do not work well. Which sort performed the best overall?