CSCI 204 Sorting Activity 2

Student name(s)_____solution_____

In this exercise, you are asked to compare the three sorting algorithms we've discussed so far, the bubble sort algorithm, the selection sort algorithm, and the insertion sort algorithm.

1. Come up with the Big-Oh notation for each of the three sorting algorithms and briefly state the reason(s).

   All three sorting algorithms, bubble sort, selection sort, and insertion sort have the complexity of O(n^2). The reason is that they all go through two levels of loop which on average would be length of n each.

2. Examine and execute the **testsorting.py** program given at the course website. Change the value of the variable **size** so that the sorting algorithm works with different sets of random numbers. Record the time needed for sorting 5,000, 8,000 and 10,000 items for the **test_large()** function. If you run on your own computer and it takes too long to run the 10,000 items case, you may adjust the sizes so that the longest run takes a few seconds (no more than 10 seconds) and the differences among the different sorting algorithms are noticeable.

3. Which of the sorting algorithms takes the longest time to complete the task? Can you see why? Discuss the issue briefly.

   On linuxremote, I tested the algorithms with large data sets (10,000 items). The time needed to complete the program is very consistent. The bubble sort algorithm takes the longest, the insertion sort is next, and the selection sort takes the least amount of time (though not by a large amount compared to insertion sort).

   This is mostly due to the fact that insertion sort does more swaps than selection sort, even though the amount of comparisons are comparable.

Lastly, not anything you need to exercise, but pay attention to the fact the Python is a very flexible language. In the **testsorting.py** program, you notice that the function names such as **bubbleSort**, **selectionSort**, and **insertionSort** can be used as variable names. These features make Python easier to use, compared to many other programming languages.