SPRING 2017: **COT 5407** INTRO. TO ALGORITHMS

[HOMEWORK 1; DUE JAN 19 AT START OF CLASS]

General submission guidelines and policies: ADD THE FOLLOWING SIGNED STATEMENT. Without this statement, your homework will not be graded.

> I HAVE ADHERED TO THE COLLABORATION POLICY FOR THIS CLASS. IN OTHER WORDS, EVERYTHING WRITTEN DOWN IN THIS SUBMISSION IS MY OWN WORK. FOR PROBLEMS WHERE I RECEIVED ANY HELP, I HAVE CITED THE SOURCE, AND/OR NAMED THE COLLABORATOR.

Read the handout on **Homework guidelines and collaboration policy** from your course website before you start on this homework. This is very important.

Problems

0. (**Regular**) Did you follow the instructions above?

1. (**Regular**) We discussed the **invariant** for SELECTIONSORT in class. You can find the pseudocode and invariants for INSERTIONSORT and MERGESORT in Chapter 2 of your text (Cormen, et al.). Write down precise **invariants** for BUBBLESORT (see p40 of text) and MERGE (not MERGESORT) (see p31 of text).

3. (**Regular**) Most sorting algorithms work correctly even if all items are not unique, i.e., there are repeats. A sorting algorithm is called **stable** if numbers with the same value appear in the output array in the same order as they do in the input array. Which of the algorithms SELECTIONSORT, INSERTIONSORT and BUBBLESORT are stable? If they are not stable give a small example with at most 4 items to prove your answer.

4. (**Extra Credit**) In our first class (Sep 2), we discussed and analyzed a simple algorithm for the SEARCH problem. We discussed two variants – one where X, the number to be searched, was bounded on both sides, and another where x was bounded below, but unbounded above. Binary search was the best strategy for the first version. The best strategy for the second version involved doing a doubling search followed by a binary search. This could be thought of as doing a LINEARSEARCH for m, the smallest exponent of 2 greater than x. What if we consider doing doubling search for m? Can we push this even further? Analyze the best algorithm for this problem.