CS 3410 - Homework 01

Due date: see course Schedule and Vista.

1. Evaluate the following algorithm via the approach outlined in Section 5.7 (p.188 DS). Your goal is to figure out the complexity of the algorithm empirically. You might start by trying to figure out the complexity on paper and try your result experimentally. Ultimately, you will probably need to try a number of different F(n)'s. In your write up, show me what you have tried and why it does or does not seem to work. You're going to have to play detective on this one. Start by designing an experiment. What range of values will you consider for the length of *list*? What will you store in *list*? How will you generate the values to store in *list*? *etc*. You have also been provided a program, *TestingShell.java* which has some code written for you (not the algorithm) to help with the experiment. There is a *displayRunTime* method that will put the timing results in a text file (append). You'll probably want to modify this. Finally, when you have some experimental results to analyze, just open the text file in Excel. In Excel, you will compute T(n)/F(n). These numbers can be very small. To aid in interpretation, it is perfectly fine to multiply all values by a constant, say a million (or more or less).

    ```
    for( int i=0; i < list.length-1; i++ )
    {
            if( list[i] > list[i+1] )
            {
                    swap( list[i], list[i+1] );
                    i = -1;
            }
    }
    ```

    You'll need to write a *swap* method and it will probably have a different signature than the one shown above.

    Present your results according to the following outline:

    > (Title Page)
    > CS 3410 – HW 01
    > Name
    > Date
    >
    > *Summary of your progress on assignment*
    >
    > *Describe in words how you conducted your experiment.*
    > *Present graphs of your results.*
    > *Summarize and draw conclusions from your results.*
    > *Repeat as necessary.*

    Submit to Blackboard:
    1. All classes used for experiment
    2. Report (*.doc* or *.docx*)