

CSCI B505 – Fall 2018

Programming Assignment 1: Due online August 31 , 2018, 11:59pm EST.

This is a toy assignment for you to get used to our computing platform. It should be very short, but you'll practice file I/O and plotting.

What to turn in: There will be **2 files** that you will be turning in via canvas. The first one is your source code (please stick to healthy coding practices: indent, put in comments, etc); the second one should contain (a) plot and (b) the discussion, as described below. Please try to submit a PDF for the second file. However, other files such as .docx will also be accepted.

What to do:

- **Implement** Bubble Sort (CLRS, p. 40) by writing YOUR OWN code (in C, C++, Java, or Python).
- We provide you with an input file that has 10000 random numbers, one number per line. Using this input file, run your Bubble Sort on the first 500, 1000, 1500, ... up to 10000 numbers and **measure its running time**.
- When you're measuring the running time, insert commands into your code that will compute the time elapsed for the sorting only. Disregard time spent on reading the input.
- Run each measurement 3 times to get the average running time. This means that you will be making $20 \times 3 = 60$ runs, but you should have 20 observations.
- **Visually plot** your measurements on a graph, where the X-axis is the number of inputs and the Y-axis is the time.
- **Discuss** what you see (is this linear, logarithmic, quadratic, exponential, etc?). Why do you think that this is happening? Do you think that the running times could change depending on whether the input is already sorted or not, and why? Also, if you ran into any special difficulties or made any interesting observations, feel free to mention them here.

Misc.:

- Be sure to check that your algorithm is sorting the numbers correctly.
- Python users: if your running times are getting out of hand, feel free to reduce the input size by some constant. For example, your inputs could be 50, 100, 150, ... , 1000 (you are still making 20 observations). Please do not use pypy.
- Users of other languages: if your running times are too fast that the plot doesn't make sense, try to increase the input size as much as you want.