CSCI B505 – Fall 2018

Programming Assignment 3: Due online September 30 , 2018, 11:59pm EST.

What to turn in: There will be **2 files** that you will be turning in via Canvas. The first one is your source code containing your two algorithms (please stick to healthy coding practices: indent, put in comments, etc); the second one should contain a plot 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** Merge Sort (CLRS, p. 34) by writing YOUR OWN code (in C, C++, Java, or Python).
- **Implement** a linear time algorithm for the following problem:
  Given a list A of n distinct numbers and an integer i, with $1 \leq i \leq n$; find the element $x \in A$ that is larger than exactly $i-1$ other elements of A.
  Hint: Note that we can solve this problem in $\mathcal{O}(n \log n)$ time, since we can sort the numbers using merge sort and then simply index the ith element in the output array; but we want to do it faster. Implement the algorithm described in (CLRS, p. 220).
- We provide you with an input file that has 100000 distinct random numbers, one number per line. Using this input file, run your Merge Sort on the first 5000, 10000, 15000, ... up to 100000 numbers and **measure its running time**.
- Run each measurement 3 times to get the average running time. This means that you will be making 2x3 = 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.

Misc.:

- 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 500, 1000, 1500, ... , 10000 (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 need.