[CS302-Data Structures] Homework 3: Basic Sorting

Instructor: Kostas Alexis
Teaching Assistants: Heyang Qin, Hemanta Sapkota, Huan Nguyen
Fall 2020 Semester

The goal of this programming assignment is twofold. In particular, it relates to you comparing the relative performance of different sorting algorithms on the same dataset and the same algorithm on different datasets.

First choose a sorting algorithm from the following list {selection sort, insertion sort, bubble sort} and another algorithm from the following list {merge sort, quick sort}.

Case 1: Randomly generate [1,000, 10,000, and 100,000] integer values (in the range of 0 to 10^6) to be inserted into the data structures. Note that the list (i.e., the input values) itself should not be sorted and all algorithms should use the same input file when the input size is the same.

Case 2: Also, test the speed of algorithms when the input array is already sorted (for the same input data).

The following output should be provided for an average of 10 sorts of each algorithm and input size:
- the CPU time (use the same machine)
- the number of comparisons
- the number of swaps

Note: You can only use (i.e., copy) code from the book and slides. You need to develop the rest of the code yourself.

You can run 2 instances of algorithms that take more than one hour.

Deliverables: (as source code or a pdf file)
- Source code for your classes (one for each sort algorithm for a total of 2)
- A main file that tests the classes with randomly generated data.
- Test output of your algorithms (with a tabular of each run and their averages)
- Refer to the theoretical complexity of the aforementioned sorting algorithms and relate the expected worst-case and average case with your results in terms of timing the CPU performance. Fit a function of CPU time and problem size and identify if this matches the theoretically expected result.
- Provide your input on the following: how is design of computationally efficient solutions important with respect to economic and social factors. How is efficient computing, and the design of fundamental algorithms for such, aligned with an

efficient economy and how can it support the goal for a resource-wise and society-wise sustainable future?

What to turn in: A softcopy of your sourcecode and reports should be uploaded to WebCampus. Combine the files in a single zip file.