CPSC 223: Project Part #3 *Due at Final*

Overview. This is the last and final part of the project. To finish up your project you must do the following.

1. **Finish the algorithm implementations.** Implement the last two of the three existing sorting algorithms.

2. **Develop and implement a new/novel algorithm.** As mentioned in part 1, you must come up with a new sorting algorithm. One approach is to develop a "hybrid" algorithm that combines (ideally the strengths) of the three algorithms you looked at and/or those we looked at in class. An example of this approach is used in Tim Sort. Alternatively, you can also develop an idea "from scratch," which could take inspiration from the approaches you've looked at this semester or elsewhere (e.g., you might look at ways people naturally perform sorting tasks). You will need to describe your general algorithm as well as implement it. Note that you will be evaluated on both your description of the algorithm and implementation (include tests showing the algorithm works correctly).

3. **Experimentally evaluate algorithm performance.** Run a series of performance tests to generate data that can be used to compare the runtime efficiency of each of the four algorithms you are considering. As mentioned in part 1 of the project, you must consider comparisons on "midsize" and "large" cases. You must use enough data and perform enough runs over the data for each algorithm to show the general runtime complexity of each algorithm. Track your performance data (e.g., using a spreadsheet). Note that you will be evaluated on your "benchmarks" (test data sets) and quality of evaluation.

4. **Performance result visualization.** Generate graphs to visualize the results of your performance tests. Your graphs should be labeled (including a title, a key, and appropriate x and y axis labels). Your graphs should also be formatted well (i.e., professional looking) and it must be possible to clearly see each algorithm's performance results. For the latter, you may need to split the graphs into separate graphs as needed. For questions on this part, please feel free to ask me.

5. **Prepare your in-class presentation.** You will have at most **6 minutes** for your in-class presentation. All team members in your group should speak for about the same amount of time during the presentation. Your presentation should include at a minimum:

(a). The three algorithms you chose and the "basic idea" behind each (including best and worst case using Big-O notation). [1 slide $\approx$ 1 min]

(b). The algorithm your team developed (i.e., the "basic idea" of the algorithm). [1 slide $\approx$ 1–2 min]

(c). A summary of the benchmark (data sets) you used for performance evaluation [1 slide $\approx$ 1 min]

(d). The performance graphs and brief analysis of the results (e.g., high level analysis of how the approaches performed and why). [$\approx$ 2–3 min]

Your presentation will be graded on both *content and clarity*. This means you must take the time to **practice** your presentation!

6. **Prepare and turn in your final report.** Turn in the hardcopy of your code (algorithms and test programs), a copy of your presentation, and any instructions that I need to be able to run your tests. Be sure the names of all team members are clearly visible on the hard copy. In addition, submit all of your code and test data on `ada` using the `dropoff` command. Note that you will also be filling out an in-class team peer evaluation as part of your project (which will be used as part of the overall grade as appropriate).