Project 1 - Report (Phase 2)

Due Apr 29 by 11pm **Points** 180 **Submitting** on paper **Available** after Apr 3 at 12am

CS 165 Project #1 — Analyzing Sorting Algorithms (Phase 2)

Project 1 involves implementing and testing various sorting algorithms experimentally to determine their real-world running times. In particular, for Phase 1 you implemented each of the following sorting algorithms:

1. **Insertion-sort**
2. **Merge-sort**
3. **Shellsort (4 versions)**
4. **Hybrid sort** (using merge-sort and insertion-sort, using 3 different cutoff points)

For Phase 2, you need to experimentally evaluate each of the above algorithms and report on your findings in a written report. For each implementation, you need to perform runtime experiments on different permutations, with multiple runs for each problem size, for increasing problem sizes. Specifically, you need to do a set of experiments for each of the following distributions:

- **Uniformly distributed permutations**, that is, permutations of the numbers, 1,2,3,...,n, where all permutations are equally likely.
- **Almost-sorted permutations**. These are generated by starting with a sorted array/vector of n numbers, say, the numbers 1,2,3,...,n, in this order. Then, independently choose 2log n pairs, (i,j), where i and j are uniformly-chosen random integers in the range from 0 to n-1, and swap the numbers at positions i and j in the array/vector.
- **Reverse sorted permutation.** This is the permutation (n,n-1,n-2,...,3,2,1).

Test 5 to 10 runs for each of a growing sequence of input sizes (e.g., 100, 500, 1000, 2500, 5000, ...), for each algorithm-input-sequence combination, average those running times, and then plot results on log-log plots. Produce a report write-up that explains the algorithms you implemented, and reports your findings, including the runtime performance you observed in your experiments. Your report will be graded based on the following rubric:

- 90 points. Correct English grammar and spelling

- 15 points. Plotting insertion-sort and merge-sort on a log-log plot and determining the slope of an asymptotic best-fit line to determine their running times experimentally, for each of the three input distributions.
- 15 points. Plotting the different Shellsort algorithms on a log-log plot and determining the slope of an asymptotic best-fit line to determine their running times experimentally, for each of the three input distributions.
- 15 points. Plotting the different Hybrid-sort algorithms on a log-log plot and determining the slope of an asymptotic best-fit line to determine their running times experimentally, for each of the three input distributions.
- 15 points. Comparing the different Shellsort algorithms to the different Hybrid-sort algorithms, to see which have similar running times and which ones are better than others.
- 15 points. Identifying which algorithms have very different running times for the different input distributions and which ones have similar running times for all the different input distributions.
- 15 points. Identifying the algorithm you think is best. Explain whether you think this algorithm is the best possible sorting algorithm or if there is a different algorithm that you think might be even better.

Turn in your written report as a PDF file via **GradeScope.**