**CS2500 Programming Project #1 SS2017**

Teams: You may work in a team of not more than 2 students (both of which must be enrolled in this class).

Overview: In this project you will use an actual machine to validate our run-time complexity analysis and asymptotic run-time complexity analysis by comparing worst case, best case, and average case run times for Insertion Sort, Bubble Sort, and Merge Sort. You should implement all three algorithms using the same language which should then be executed on the same machine. Part of your analyses should include plotting their time behavior as a function of input size (n) and compare this with plots of the expected behavior for the given complexity classes.

Submission: Your submission will be a written report comparing the algorithms’ performance and perhaps including discussion of some of the items below. You are additionally required to submit the source code used in the analyses, but the grading focus is on the written report and thorough analysis of the algorithms. Feel free to include pseudocode in the report and/or specific details concerning implementations which may lead to differences in performance.

Some additional questions to think about and discuss within the report:

What size n0 is required to begin to exhibit asymptotic complexity?

What inputs are required to generate “average” complexity? How about “best case and worst case”? Are the inputs the same for all algorithms? If not, is this a fair comparison?

How does measured run time correspond to operation counts that we use in abstract complexity analysis? Are you counting the same thing?