

CS 4833 Homework 1

Due Monday, Feb. 13

There can be many algorithms for solving the same problem. One algorithm may have better asymptotic complexity than another. But for inputs below a certain size, it may not run as fast in practice because of constant factors that are ignored in asymptotic complexity analysis.

For this assignment, you will demonstrate this experimentally for a pair of sorting algorithms: selection sort and a divide-and-conquer sorting algorithm, which can be either quicksort or merge sort. You will also show it for a pair of integer multiplication algorithms: grade-school multiplication and Karatsuba's divide-and-conquer multiplication algorithm.

You may use freely-available code for these algorithms that you find on the Internet. However, you cannot use the code of another student or collaborate with other students on this assignment. Moreover, the code you use should be relatively simple and easy to read, without any fancy optimizations that could affect the timing comparison.

For your two sorting algorithms, and for your two integer multiplication algorithms, perform timing experiments with different input sizes, and show the results in either a table or a graph. Your objective is to identify the crossover point where the asymptotically superior algorithm is also more efficient in practice.

Your report should include:

- A description (of no more than one page) of your experiments and results, including a brief description of the computer system you performed your experiments on.
- Source code for each algorithm and where you got it, including any code for timing the algorithms.

You are free to discuss this assignment as much as you want with the instructor, the TA, and even your fellow classmates, and I recommend doing so if it will help you better understand how to perform an experimental timing analysis. But the experiments you report must be your own. Note that results often vary depending on the computer platform you use, input data, and implementation of the algorithms.