

CS 3307

Assignment 1

Due September 14th, 2025, at 11:59pm CT

Instructions

1. Implement the following sorting algorithms in C++:

- Insertion Sort
- Merge Sort
- Quick Sort

2. Evaluate each algorithm's performance on random data.

- Generate vectors of various sizes n , filled with random integer values. For example, you could test $n = 10, 100, 1000, 10000, \dots$ up to a value where sorting takes 10s of seconds.
- Run each of your sorting algorithms on these vectors and measure their runtime performance.
- Create a plot where:
 - The x-axis represents n (vector size).
 - The y-axis represents runtime (in seconds).
 - Three lines are plotted, one for each sorting algorithm.
- Ensure that your timing measurements include **only the sorting time** (do not include time to create or initialize vectors).

3. Evaluate each algorithm's performance on sorted data.

- Repeat Step 2, but this time use **already sorted vectors** (e.g., initialized as $[0, 1, 2, \dots, n-1]$).

4. Write a 1-page, single-spaced report (12-point font) discussing the following:

- Runtime comparisons between the algorithms.
- How the observed performance relates to their theoretical Big-O time complexities.
- The differences in performance when sorting random vs. sorted input.

- The space complexity of each algorithm and how it affects practical performance (i.e. using large amounts of memory can make code slower due to hurting cache performance).

Deliverables

- C++ Source Code implementing the three sorting algorithms.
- A report which includes:
 - Two plots (one for random unsorted vectors, one for sorted vectors) showing runtime vs. input size.
 - A 1-page writeup that accomplishes point 4 of the instructions.