

Algorithms Homework 1

Divide and Conquer

Due data: April 17, 2025 23:59

Sorting is one of the most common algorithms used in computer science. Most *efficient* sorting algorithms use the divide-and-conquer method, such as **merge sort** and **quick sort**.

The original designers of the above two sorting algorithms usually recursively called the function until only zero or one element remained. However, we may choose any arbitrary size as the terminal condition of our sorting algorithm. For example, when the size of the current sub-problem is less than 5, we can use the insertion sort (or any non-recursive method) to sort these elements.

Your job is to implement both the merge sort and the quick sort function and **observe how the different terminal conditions** affect the **actual running time**.

- Implement merge sort and quick sort.
- Test different terminal condition to trigger the non-recursive part. (e.g. $n < 5$, $n < 10$)
- Test your program with different input sizes: 1,000, 10,000, 100,000, ..., etc.

We do not require the program to output the result, but you may output it yourself to check the correctness of your program.

You need to submit a zip/rar file with

- A folder with name *code* includes all of your codes (.cpp, .h, and etc.)
 - Including all the versions you implemented.
- A pdf file with name report.pdf.

In your report, at least you need to explain

- The contains for each of your source code (what you implement)
- The terminal condition(s) you chose.
- The non-recursive method(s) you chose.
- The different input sizes you chose to test your program.
- The experiment results of the different combinations of the above 3 things.

You may also

- Try different things may affect the performance.
 - Different types/orders for the input data
- Try to compare your function with other sorting method / function

Note: you may see the following page to measure the running time of your program
<https://www.geeksforgeeks.org/measure-execution-time-function-cpp/>