

Programming Assignment II

Sorting Algorithms

Due Date: 2024/09/25 (30)

1 Description of Assignment

Implement at least the following sorting algorithms and compare their performances.

1. Insertion sort
2. Selection sort
3. Quick sort
4. Merge sort
5. Heap sort

The first set of input data to be sorted should be randomly generated. Use the same random number generator and the same seed for each of the sorting algorithm. Then, if you still have time, try using other types of input data (for example, sorted or reversed data) to analyze the performances of the sorting algorithms.

Print out the CPU time required to sort n elements for each sorting algorithm, and include them in the report of your assignment. The number of elements to be sorted should be from $n = 10000$ to $n = 100000$, or even larger. If the number of elements is too small, it may not be possible to distinguish the performances of the algorithms. You can also report the maximum number of elements that can be sorted by these algorithms within a given CPU time (e.g. 5 minutes).

Before comparing the performances, make sure that all sorting algorithms are implemented correctly. Therefore, your program should provide [debug mode](#) to check the results of sorting. Do not include the time for checking the correctness of each sorting algorithm in the CPU time of that sorting algorithm.

Notes

The format of the report of the assignment should be close to a technical research report, and include at least the following items:

1. Title and Author.
This section should include [assignment number](#), *your name*, *student number* and *email address* on the *first* part of the *first* page of your report.

2. Description of the Problem.

A “formal” description of the problem in this assignment.

In addition to the basic requirements specified in the assignment, highlight additional functions or features that you have implemented.

Do not copy assignment instructions directly into this section.

3. Main results.

This section should include at least the following items.

(a) The design of your program.

If the design or part of the design was obtained by reference to other sources, discussions, etc., appropriate citations should be given.

(b) The data structures used in the program to improve the efficiency of the program.

These data structures should be implemented by you and appear in the first part of your program.

(c) List of your program with comments.

i. If your program is very long, list only the main parts of the program here and the entire program in appendix.

ii. Additional comments can be added manually to explain the design of the program.

(d) Outputs of the compilation and the executions of your program.

This section should use [screen dumps](#) whenever possible.

4. The performance evaluation of your program.

(a) List the execution time of your program with various input sizes, such as $n = 100, 200, \dots, 1000$.

(b) The maximum input size that your program can run in a reasonable amount of time, such as 1 minute, 5 minutes, or 10 minutes.

5. Conclusions

(a) Summarize what you did and the interesting things you learned from this assignment.

(b) Describe the difficulties you encountered during program development and how you overcame them. (This is strong evidence that you did the work yourself.)

Additional notes:

1. Turn in your report on or before due day.

2. The output of the program execution should indicate the correctness of your program. In other words, a set of [comprehensive](#) (but not necessarily exhaustive) annotated test data for the problem should be provided to show that

your program is indeed correct. This can be done by carefully selecting a set of test data.

3. Print or write the report on A4 paper and staple it to the upper left corner.