Programming Assignment II
Sorting Algorithms

Due Date: 2025/09/24 (30 points)

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 each assignment report should be cloase to a technical research report and must include at least the following sections:

1. **Title and Author**
   On the first part of the first page, clearly include:

- Assignment number
- Your name
- Student number
- Email address

2. **Description of the Problem**
   Provide a clear and formal description of the problem of this assignment.
   In addition to the basic requirements given in the assignment, highlight any extra functions or features you have implemented. Do not simply copy the assignment instructions into this section.

3. **Main Results**
   This section should include at least the following:

   (a) Program Design.
   Explain the overall design of your program. If any part of the design was obtained from references, discussions with other people, or other sources, proper citations must be provided.

   (b) Data Structures.
   Describe the data structures you implemented to improve efficiency. These should be your own implementations and appear in the first part of your program.

   (c) Program Listing with Comments.

   i. If the program is long, include only the main parts in the report body, and place the complete program in the appendix.

   ii. Add explanatory comments where appropriate to clarify your design.

   (d) Program Outputs.
   Include compilation results and execution outputs. Whenever possible, provide screen dumps.

4. **Performance Evaluation**

   (a) Report execution times of your program with various input sizes, such as $n = 100, 200, \ldots, 1000$.

   (b) Indicate the maximum input size your program can handle within a reasonable time limit (e.g., 1, 5, or 10 minutes).

5. **Conclusions**

   (a) Summarize what you accomplished and any interesting insights gained from this assignment.

   (b) Describe the challenges you encountered during development and how you overcame them. (This provides strong evidence that you completed the work independently.)

Additional Notes

1. Submit your report on or before the due date.
2. The program output should clearly demonstrate correctness. That is, provide a set of comprehensive (but not necessarily exhaustive) annotated test cases to show that your program works correctly.
3. Print the report on A4 paper and staple it in the upper-left corner.