Lab 2: Sorting

Each student chooses either of the following sets for further requirements.

- **Set 1** (7 algorithms): Selection Sort, Insertion Sort, Bubble Sort, Heap Sort, Merge Sort, Quick Sort, and Radix Sort.
- **Set 2** (12 algorithms): Selection Sort, Insertion Sort, Binary-Insertion Sort, Bubble Sort, Shaker Sort, Shell Sort, Heap Sort, Merge Sort, Quick Sort, Counting Sort, Radix Sort, and Flash Sort.

1 Programming

1.1 Algorithms

- Students are required to implement all sorting algorithms (for **ascending order** only) from the chosen set using *C/C++* language.

1.2 Experiments

```
for each Data Order S1:
  for each Data Size S2:
      for each Sorting Algorithm S3:
          1. Create an array with Data Order S1 and Data Size S2
          2. Sort the created array using the Sorting Algorithm S3
             and measure the running time (millisecs) of the implementation
          3. Take note S1, S2, S3 and the running time
```

1.2.1 Input Data Order

- Examine the chosen sorting algorithms with data of different arrangements, including:
  - Random Order Data
  - Nearly Sorted Data
  - Sorted Data
  - Reverse Data

  See `DataGenerator.cpp` for more information.

1.2.2 Input Size

- Examine the chosen sorting algorithms with data of the following sizes: `3,000`; `10,000`; `30,000`; `100,000`; `300,000`.

2 Report

Include the following contents:

1. Presentation on installed algorithms: ideas, algorithms (step-by-step), algorithm's review (time complexity, space complexity if possible).
2. Presentation on experimental results and comments.
   - How to present experimental results: instead of giving runtime numbers, we should visualize graphs; thus, it will be easier to observe and comment. You will draw 4 graphs corresponding to the 4 input data states. In particular, each graph has a vertical axis as the data size, the horizontal axis is the run time (as shown in Figure 1)

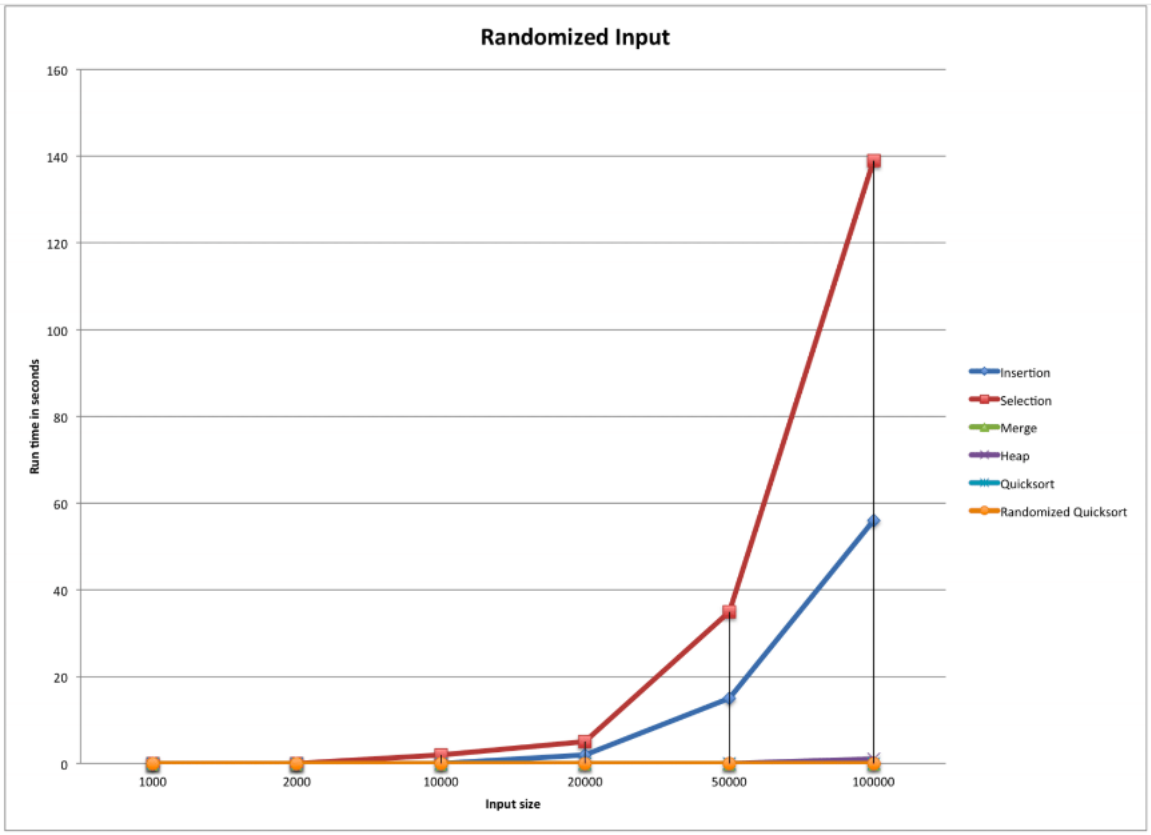

Figure 1: Illustration Graph

   - Comments on the graphs which are drawn (the fastest / slowest algorithm(s) in each case, time acceleration of algorithms, etc.). Explain.
   - Overall comments of algorithms on all Data order and all Data size (the fastest / slowest algorithms overall, grouping the stable/unstable algorithms, etc.)

Submission regulation

- Students create a folder <Student's ID> containing the contents following:
  - <Code> folder: contains the whole project. (Only files with **.cpp** and **.h** extension is required.)
  - <Report.pdf> file: is the file containing the report.
- Compress the above folder into Student's ID.rar(.zip) for submission.
- Submission with wrong regulation will result in a "0" (zero).
- Plagiarism and Cheating will result in a "0" (zero) for the entire course and will be subject to appropriate referral to the Management Board for further action.