Programming assignment #1

Sorting

Objective

1. To understand how to design an algorithm.
2. To study the concept of recursion.

Problem Definition

1. Implement insertion sort, merge sort, heap sort, and quick sort
2. Using recursion for merge sort and a quick sort, implement the program in C++
3. Analyze the runtimes for these algorithms
4. Compare the runtimes with the asymptotics you expect

Experiments

Adar needs to sort some lists and he knows that sometimes the input list is often very close to already being sorted. He wants you to help him by implementing three sorting algorithms and comparing their speed on three types of lists. The three sorting algorithms she wants you to test are insertion sort, merge sort, and quick sort. The three types of lists you will be testing against are generated in the following manners:

1. A uniformly random sorted list
2. An arranged sorted list(min to max)
3. A sorted list where one percent of list randomly chosen elements were swapped
4. A sorted list where one percent of list randomly chosen adjacent elements were swapped

I/O Format

Your program must be able to read an input file. The I/O file names are arguments to your program; in other words, those file name can NOT be fixed. In command line, your program is invoked by:

```
./a.out  1  inputfile  outputfile  //1:insertion sort  2:merge sort  3:heap sort  4:quick sort
```

Input file example

```
2 55 62 68 82 112 118 147 154 209                    //array 1
```

Program Submission

1. Please use C/C++ language and your program must be written in only one source file.
2. Your source file must be named as “Student_ID_number_hw1.cpp” and please make sure that all characters of the filename are in **lower case**. For example, if your student number is 9711592, the name of your program file should be “9711592_hw1.cpp”.

Report

1. No more than 3 pages.
2. Your report must contain:
   a. The flow chart or the pseudo code of you program.
   b. The experimental results and analysis.
3. The report file name must be “Student_ID_number_hw1.doc(x)” or “Student_ID_number_hw1.pdf” and please make sure that all characters of the filename are in lower case. For example, if your student number is 9711592, the name of your program file should be “9711592_hw1.doc” or “9711592_hw1.pdf”.

Grading

You need to submit both your source code and report. Remember the submission rules mentioned above, or you will be punished on your grades.

●	Unique and compilable source code	30 %
●	Performance	20 %
●	Report	40 %
●	Questions in demo	10 %

Due Date

Upload your report and program to the course website and there will be a **demo** of this assignment. All of your files must be archived to only one file named “**Student_ID_number_hw1.zip**” or “**Student_ID_number_hw1.rar**”. You have also to handed in the report when demo.

The upload dead line would be **at pm 11:59 on October 23, 2016**. The demo schedule would be released before Oct. 23.