

Assignment #7 (Efficiency of sortings)

Goal: To compare the efficiency of five different sorting algorithms

Guide :

I. Each student has to implement three sorting algorithms by himself and directly call two sorting functions of C / C++.

To compare the efficiency of various sorting algorithms, your task is to implement three different sorting algorithms (one in type A, one in type B and one in type C) and directly to call two sorting functions of C/C++(type D and type E). You can refer to the source code provided in the text book for your implementation. In addition, to compare these five algorithms, you have to write a main program for performing the testing. The execution time for various data sizes should be included in the report.

Type A:

1. selection sort
2. insertion sort

Type B:

1. heap sort
2. merge sort

Type C:

1. quick sort
2. radix sort

Type D:

qsort() function in C library stdlib.h or cstdlib

Type E:

sort() function in C++ library (STL)

The rule for determining what sorting algorithms you should implement is as follows. Suppose the last two digits of your student ID is $n = 10x + y$, then $(n \bmod 2) + 1$, $(x + y \bmod 2) + 1$, and $((x + y * 2) \bmod 2) + 1$ denote the algorithms you have to implement for types A, B, and C, respectively. For example, for $n = 15$, you will obtain the numbers 2, 1, and 2 for types A, B, and C, respectively. For $n = 25$, the numbers are 2, 2, and 1.

II. Program Format

To facilitate the testing, you have to follow these formats when you write your program:

- (1) Please combine your source code into a single file. However, in this file, you should implement three sorting algorithms as three different functions.
- (2) The input file name is file “input.txt”, whose format is defined as follows.

```
N
X1
X2
.
.
XN
```

The number N in the first row represents the number of integers to be sorted. The following $X_1 \sim X_N$ are the N integers (one per row) to be sorted. The sorted results should be output in five different files, which are “outputA.txt”, “outputB.txt”, “outputC.txt”, “outputD.txt”, and “outputE.txt” (corresponding to the five types). Each output file must have the name of the sorting algorithm in its first row (e.g. Insertion Sort, Quick sort, C `qsort()`, C++ `sort()` and so on), followed by N numbers sorted by ascending order in the next N rows. Do not forget to close your file pointer after the output is done. Though you have to edit your “input.txt”, please note that TA will always test your program with his own “input.txt”.

- (3) In your submission, besides your source code and your testing report, you have to prepare an input.txt file of your own.

(4) Auto-testing for various data sizes:

In this assignment, you are asked to compare these five sorting algorithms with different data sizes, and write a report for your results. The data sizes to be tested are 100, 500, 1000, 5000, 10000, 50000, 100000, 500000..... For each data size, you have to do the experiment 10 times for 10 different data sets. Please use the random function to create your own dataset. Note that in this auto-test, to save your time, it is not necessary to output the sorted files. For each fixed data size, please record the average execution time of 10 data sets. To give a fair comparison, you should adopt the same data sets for each algorithm. The execution time is

measured in seconds. If it takes too much time (e.g. time > 5 minutes), then record it as “TLE”, which stands for “Time Limit Exceed”. The following table is an example for your experiments.

data size	Insertion	Heap	Radix	C qsort	C++ sort
100	0.001				
500	0.025				
1000					
5000					
10000					
50000					
100000					
500000	TLE				

note : The execution time is measured in seconds

TLE stands for “Time Limit Exceed” (>5 minutes)

III. **Report**

Report should be written in a Word file, which contains:

- (1) The environment: your hardware spec (CPU speed, capacity of main memory), your operating system, name of compiler and version of compiler.
- (2) Your experimental results, as the table given above.

IV. **Additional Notes**

- (1) Please do the auto-test experiment only if your implementations are correct. Your report will not be accepted if your implementations are incorrect.
- (2) In this assignment, using source codes from the text book or from the Internet will not be considered plagiarism. However, please complete the main testing program, generate random numbers, do the statistics of sorting time, do the experiment, and write the report by yourself. With this assignment, you are supposed to understand the concepts of and differences between these sorting algorithms.