CSCI 377 Computer Algorithms

Assignment 1

Part 1

Due Date: 22 Feb 2019

1) Introduction:

Sorting algorithms are widely used in practice. In this assignment, student will practice implementing various sorting algorithms learned in class: insertion sort, merge sort, heap sort, quick sort.

2) Resources:

You can find insertion sort and merge sort in chapter 2, heap sort in chapter 6 and quick sort in chapter 7.

3) Description:

All programs must be written in C or C++. As part of your code, you will include two counters that count the number of data comparisons and data moves. Also include a timer to see how long it takes the program to execute.

The following examples show when to include your counters:

```
Dummy_sort(A, n, counter1, counter1)
for i <- 1 to n-1 do
  for j <-i+1 to n do
     counter1<-counter1+1
     if(A[i]>A[j]) then
        swap(A[i], A[j])
        counter2 <- counter2+1
```

You will find on CUNY blackboard a main.cpp, a makefile and 11 data files. You will use the main.cpp to run your programs. You can either make four separate cpp files for each sort and run together with main.cpp or code all your sorting algorithms in main.cpp.

You will use the 11 data files to test your code and observe the behavior of four sorts. These files all contain shuffled lists of integers, with one integer listed per line. The files are:

Filename	# items	lowest	highest	Description
Num8.txt	2^3	1	8	no omissions, no duplicates
Num16.txt	2^4	1	16	no omissions, no duplicates
Num32.txt	2^5	1	32	no omissions, no duplicates
Num64.txt	2^6	1	64	no omissions, no duplicates
Num128.txt	2^7	1	128	omissions/duplicates possible
Num256.txt	2^8	1	256	omissions/duplicates possible
Num512.txt	2^9	1	512	omissions/duplicates possible
Num1024.txt	2^{10}	1	1024	omissions/duplicates possible
Num2048.txt	2^{11}	1	2048	omissions/duplicates possible
Num4096.txt	2^{12}	1	4096	omissions/duplicates possible
Num8192.txt	2^{13}	1	8192	omissions/duplicates possible

4) **Analysis**:

Analyze and discuss the results of your experiment. You may want to plot your results using Excel or a graphing program to help you visualize the results better. At the very least answer the following questions:

a) Discuss what your results mean regarding the theoretical run-time of the different algorithms.

b) Do the sorts really take $O(n^2)$ and $O(n \lg n)$ steps to run?

c) Explain how you got your answer to this question.

d) Which of the sorts takes the most steps?

e) Which of the $O(n \lg n)$ sorts takes the most steps?

f) Why?

g) Under what circumstances might you prefer to use one of the sorts versus others?

h) In general, which sort seems preferable?

i) Why?

Academic Honesty:

Please do your own work on this assignment. No collaboration in coding or writing the laboratory report is allowed. Downloading code from the web is also NOT permitted for this assignment.

How to turn in:

1. By the due date, submit your lab report and source code as well as an instruction how to compile and run your code to the instructor via email at nlu@jjay.cuny.edu.
2. If you are submitting multiple files, compress and submit a single zip file.
3. Submit before 3pm on Feb 22.