CS 456: Advanced Algorithms

Programming Assignment #01

Assigned Date	: Thursday, September 18, 2014
Due Date	: Monday, September 29, 2014 @ 12:29:59 p.m.

Overview

Your first programming assignment is to **implement Merge Sort and Insertion Sort**. The objective of this assignment is to **validate** runtime complexity analysis and asymptotic runtime complexity analysis by comparing best case, worst case, and average case run times **using an actual computer**. More specifically, you are expected to think about and address the following questions:

a. At what size n_0 does your implementation start to exhibit asymptotic complexity?
b. At what input size n does Insertion Sort *beat* Merge Sort?
c. What inputs are required to generate *average* complexity. How about best case and worst case?
d. How does the measured run time correspond to the abstract complexity analysis using operation counting (as discussed in class)?
e. How to create your test driver so that it exercises your sort programs.
f. How to create the sorting class so that it will be extensible and reusable for future projects.

Instructions

- This is an individual assignment. **Do your own work**.
- **Start early!!**
- **Take backups of your code often!!**.
- The report part of your solution must be produced using a word processor. Any figures, graphs, plots, etc., should also be produced using appropriate computer applications. Graphs/plots should be properly labeled.
- You may use any programming language of your choice. However, you should make sure that your code compiles and runs on a typical Linux machine.
- Follow a good coding standard. Use the Google C++ coding standard found here `http://goo.gl/1rC1o`, if you don't already follow one.
- Total points: [**100 points**]

Deliverables

The due date of this assignment is **Monday, September 29, 2014 @ 12:29:59 p.m.** A dropbox will be opened for submission on Moodle before the due date. A complete solution comprises of:

- A report that includes the followings:
  - Motivation and background of the experiment [**5 points**]
  - Pseudocode with Invariants and pre/post conditions [**10 points**]
  - Testing Plan and Test Results [**20 points**]
  - A correctness proof of your programs [**20 points**]
  - Problems Encountered/Key insights [**5 points**]
  - Conclusion and performance comparisons [**20 points**]
  - Program listing
    - Good programming structure (headers, variable names, code re-use, functional decomposition, object-oriented design, and comments) [**5 points**]
    - Implemented pre/post conditions [**5 points**]
    - Implemented Invariants in program [**10 points**]
- A compressed tarball of the directory containing your source codes. Do not include executables in this tarball. To create a compressed tarball of the directory source, use the following command: `tar -zcvf name-111-pr1.tar.gz source/`. Obviously, change the name to your last name and 111 to the last three digits of your SIUE ID.