

# CSCI 36200: Data Structures

## Programming Assignment 3

Instructor: Dr. Snehasis Mukhopadhyay

Due date: April 11, 2019

The objective of this assignment is to analyze the time complexities of four different sorting algorithms. The project has the following components:

1. Implement a random number generator function that can generate integer numbers between 1 and 20000. Duplicate numbers should not be allowed. You can make use of any of the random number generating available in a programming language library, or you can write your own pseudo random number generating code.
2. Implement algorithms for **insertion sort, quick sort, heap sort, and merge sort**. Given a choice between a recursive and an iterative version for a particular algorithm, you may choose either one.
3. Execute each sorting algorithm for the following different values of  $n$ , where  $n$  is the size of the list to be sorted,

$$n = 100, 500, 1000, 2000, 5000, 8000, 10000.$$

In each case, the list consists of random numbers generated in step 1. For a particular value of  $n$ , the same input data (*i.e.*, list to be sorted) should be given to all the sorting algorithms.

4. For each value of  $n$ , and for each sorting algorithm, obtain the actual execution time of the algorithm for the machine you are using for the cases when
  - a) an unsorted list is presented as the input data, and
  - b) a sorted list is presented as the input data

Make sure your sorting algorithms work properly by examining the sorted data. However, do not submit any sorted data as output. No output files need to be submitted. You are required to submit the source code and an input file for size  $n = 100$ . You are required to submit a project report consisting of the following components:

- (i) Statement of the objectives of the project.
- (ii) Brief description of how each algorithm works (do not write them as algorithms).
- (iii) Two plots: one showing data size against execution time for each algorithm on a single graph for unsorted data, and the second one showing similar plot for sorted data. For plotting purposes, use a log scale for data size.
- (iv) Comments on the actual execution times you recorded against the complexities discussed in the class.

(v) Any other comments and conclusions.

*There is no limit on the length of the report.* A suggested maximum length (including the plots) is 5 pages.

As usual, you will be graded for the project on the following points: (i) correctness of the program, (ii) organization of the program, (iii) readability of the program, and (iv) your report. Your code will be tested to verify correctness.

