|  |  |
| --- | --- |
|  | Computer Science 211  *Data Structures* Spring 2018 |

## Assignment– Analysis of Common Sorting Algorithms

### Part 1 — Correct Sorting Algorithms

Create Java methods for each of the following, based on sorting arrays of integers:

1. Bubble Sort
2. Selection Sort
3. Insertion Sort
4. Quick Sort
5. Merge Sort

### Part 2 — Time Trials for Sorting Algorithms

Create a testing shell for a sorting method that that times the method.

The testing shell should:

* Generate a random array of integers of a prescribed length.
* Start a timer.
* Sort the array.
* Stop the timer and calculate the elapsed time.

You should test the five methods created for part 1 above, with increasing larger data sets --

100, 1,000, 10,000, 100,000, 1,000,000 and 10,000,000 values

First, determine which of the sizes is appropriate for each sorting method, based on the time it takes to sort. If a method takes more than one minute, then you do not need to test the method with that size data set. For example, a merge sort might work with 1,000,000 items in a few seconds, but a selection sort will take much longer, so it might be okay to do the insertion sort up to 100,000 items, but not with larger data sets. You can determine exactly which size data sets you use with each soring method – you have some flexibility, but the basic idea is not to run sorts that take more than about a minute or two to complete.

Put your results into an Excel Spreadsheet.

### Part 3 — Temporal Efficiency Analysis

Write a report on the results of your work in this assignment. How does the time it takes for each algorithm to run compare to the size of the data set? What are the best case, worst case and average case times for each method for each array length? How does it fit with functions like T = f(n), T= f(n log(n) ) or T= f( n^2)? What does our book tell us about the time efficiency of these algorithms? Based on this, what did you expect the results to look like before running the time trials? How did the results compare with this? Which algorithm was most efficient? Why was this more efficient than the others?

Your report should include:

* a brief description of the topic – what is this experiment all about? What do we hope to learn from it? Why is this important?
* a brief discussion of what you would expect to see in the experiment based on the information in the textbook.
* a description of your experiment – what did you do? What equipment and software did you use? Include the clock speed of your computer, the specific compiler you used, and a listing of the source code for your methods.
* a presentation of your results – including tables and graphs of results. Create graphs that illuminate key points about your data, it is not necessary to graph everything.
* a commentary or analysis of results. What does the data show? How does this compare to what you expected?
* a bibliography. Cite books or Web sites that you looked at, along with some that have links to further information on the topic.