**Project Report 5**

**Problem Statement:**

The primary goal of Homework 5 is to familiarize myself with the various properties and intricacies of sorting methods such as insertion, selection, quick, and merge sort. In addition to creating the sorting algorithms, each will be analyzed and compared by their run times. Each sorting method will sort by various attributes a list of almost ten thousand books. The program inputs are numbers 1 to 5, which allow for menu navigation. The program outputs are several .csv files which are examples of the various sorting methods executed in the code and the runtimes of each algorithm. As the inputs for this assignment were simply single-digit integer values, there was little error handling needed. The switch case needed to identify if a user input was an integer between 1 and 5.

**Design:**

To get each sorting algorithm to work seamlessly, there was little room for any deviations from what was required of the assignment. To read in the data from the .csv file and rearrange them, it was imperative that the object used to store it was a vector data structure. The algorithms used in the program were mainly the various sorting methods outlined in the Assignment 5 problem statement but also swap, partition, and merge. The program would not have executed according to the problem statement were it not to have all these data structures and algorithms. Even obscure algorithms like swap and partition held an important place in some of the algorithms.

**Implementation:**

Like the many assignments that came before this, I started by implementing the Book class into the program. This involved creating a read file function that takes in six attributes from the .csv file and filling a vector object with this organized data. Then I wrote the write file function which utilizes the getter methods from the book class to create a text file with the sorted data. Finally came the implementation of each sorting algorithm, using Dr. Gauches’ slides as a general guide, with slight caveats and nuances, I was able to implement each sorting algorithm.

**Testing:**

The main way I tested my program was by running many trials on each sorting algorithm and modifying how and what was being sorted between them. I would start with insertion sort and try sorting each attribute in ascending order. If each .csv file output correctly, I would move onto testing the selection sorting algorithm. I did this for both other sorting methods and verified they worked accordingly. Additionally, I took into account the efficacy of each respective sorting algorithm and based my confidence on the runtimes around my general knowledge on how fast each sorting method runs. For navigating the menu, I input numbers 1 to 5 and made sure each sorting algorithm executed properly. Special cases I tested included numbers out of the range of 1 to 5 and other data types. Everything worked as expected.

**Conclusion:**

The result of the assignment outlined the efficiency and speed of several sorting algorithms. The project was a success ignoring the fact I submitted it late. I would once again start on the assignment sooner and summon the help of \ professors and TA’s to help me overcome issues that took too much time out of the week to fix. The project took a combined five hours to reach its completed form.