
Lab Report – Week 5 – *Recursive Sorting Research Assignment*

Issam Ahmed
CSCI 112 900 Summer 2019

Assignment Analysis and Design

This assignment was a research test to compare the processing time of two recursive sorting methods, Quick Sort and Merge Sort. To facilitate this, a program was made to for both sorting methods. The methods that performed the actual sorting was provided, but the rest of the code was written to test these given methods.

First, a menu interface was created in the main method to let the user choose between the two sorting methods. Whichever sorting method is chosen, the testing methodology is the same. First, a for loop was made to run each method 100 times. Within the loop, an array was created with 10 million random 6-digit integers. This unsorted array was then fed into the specified sorting algorithm. Before the sorting method is called, the time is recorded, and after the method was completed, the time was recorded again. Subtracting these two values gave the total time it took to complete the sorting algorithm. This is then shown on the console and all the time data was exported to a CSV file.

Assignment Code

The assignment program was submitted as a zip file.

Assignment Testing

For this assignment, I first tested out each sorting method to see if it produced a sorted list of integers. This was done with only an array size of 100 to save time. Once confident of the sorting methods, I tested the menu interface with test stubs instead of the actual sorting algorithms. Once satisfied by both the input interface and sorting methods, I tested out the output section. This was done by running the algorithm 100 times but with a small array to see if the CSV and Console exported the processing time.

Assignment Evaluation

The assignment programming was straight forward since the hardest part, the recursive sorting algorithm, was given to us. This makes sense since the actual code is complicated and would take time to recreate, even if given pseudo code. Creating the test program around the sorting methods was also easy since it was a basic program. The hardest part was trying to figure out how to call the sorting methods and understanding how they functioned. This was not necessary for the assignment but understanding it would help in the future with more complex sorting requirements.

One way to improve the test program would be to give the user an option to choose the number of iterations or even the size of the data set. Another way to improve this would be the ability to test different types of data sets. For example, an object array or random strings. However, for the purpose of our research this program was more than adequate.