Sorting Algorithms

*Comparison of Merge, Bubble, Selection, and Insertion Sorts

Sean Robbins

Chapman Student
Orange, United States
srobbins@chapman.edu

***Abstract*—This is a write up about a comparison experiment in the uses of different sorting algorithms. The algorithms in question are Merge Sort, Selection Sort, Insertion Sort, and Bubble Sort.**

***Index Terms*—Sorting Algorithms**

I. Introduction

These algorithms were tested against each other on the same sets of randomly generated data to see how they would perform.

II. Implementation

A. Context

The algorithms were all run against one another on the same computer. Whether or not this specific devise was optimized to perform one of these algorithms is unknown. All the algorithms were run on the exact same data, that were stored as numbers in the form of the type Double. In addition to this these randomly generated numbers were read in from a text file that was passed into the program as a command line argument.

III. Results

When all these algorithms were run against each other Merge Sort proved to run the quickest of all of them. Averaging about 3/1000 of a second to run while insertions and selection both averaged around 12/1000 of a second and bubble sort lagged behind greatly at 54/1000 of a second. From the comparison of these numbers it was fairly obvious that merge sort was far more efficient at sorting the data given than insertion, selection, and bubble sort were. One thing of note however, was that due to mereg sorts unique method of dividing the data to be sorted down to its smallest parts and reintegrating from the ground up sorting as it goes that the amount of memory consumed by merge sort when it is running is head and shoulders above the other sorting algorithms

Identify applicable funding agency here. If none, delete this.