Sorting algorithms

Cadre Carrigan

05/15/2019

For this sorting project, I tested BubbleSort, InsertionSort, SelectionSort, MergeSort, and QuickSort. After running all of the algorithms, time differences were fairly similar, but, because the files and the amount of data being sorted is so small, it is hard to see significant differences. But, with all things considered, there are different reasons for using different algorithms such as...

- Implementation
- Size of data
- Complexity of program

Depending on the program, one may choose to use a more simple method as the data is not very big or the computer does not have as strong of processing power/RAM. On the other hand, with large amount of data, the user would want to go with a more complex, but efficient algorithm to lower run time.

Unfortunately, while we can see using mathmematical analysis which programs are faster and better suited for specific scenarios, empirical analysis is hard to use in this instance due to the size of the data we are using. The small data size does not allow us to see a significant difference between the algorithms. This makes it hard to use just empirical analysis alone in this instance. If I was able to have access to larger data, it would be much easier to see a difference as well as compare it to runtimes with smaller data to see which algorithms are more effecient for each size of data.