Stephanie Gremillion

Charlyne Walker

COP 3337 - RVBB

24 July 2022

Project 5 – Written Component

The merge sort algorithm works by separating the values into two halves recursively until each half is of minimum size. Then, it sends each half to be sorted, returns the sorted halves to be sorted with the other sorted halves. It tackles the problem by creating smaller and smaller sorts, more easily handling the bigger sort (as everything is already sorted, it just needs to figure out which one goes first).

The first method in the merge sort algorithm determines the mid-point. It separates the input into the first half and the second half, using these halves to call itself recursively until the halves are small enough. The second method is the merge method that takes the two halves, determines the order they should be in, and sends them back. The smaller it is, the easier and faster it is to sort. It is essential that the first method breaks down the values into smaller groups it is easier to sort smaller values and it is also easier to sort already sorted values.

The insertion sort algorithm takes each value, compares it to the other values, finds where it belongs, and inserts it into that location. The selection sort algorithm finds the smallest item, compares it to the unsorted items, and puts the smaller first. Both algorithms loop through each value but only insertion compares against the already sorted elements.

By breaking down the problem into smaller problems, merge sort works faster on larger sets of data than insertion of selection sort. For the former to sorting algorithms, increasing the size of the sets of data significantly increases the time it takes to sort. Timewise, insertion and selection sort take O(n^2) amount of time to sort. As the number of values increases, the amount of time increases significantly. However, merge sort uses O(n log(n)) amount of time to sort. While the time increases as the values increase, the proportion compared to the other two sorts is significantly lower.

HORSTMANN, CAY. *Big Java: Early Objects*. 7th ed., JOHN WILEY, 2020.

Rowell, Eric. “Know Thy Complexities!” *Big-O Cheat Sheet*, https://www.bigocheatsheet.com/.