

Isaiah Jenkins

Array Size - 100,000

Run sizes for MySort

100,000 run size - 5917 milliseconds

50,000 run size - 2944 milliseconds

25,000 run size - 1071 milliseconds

MergeSort

26 milliseconds

Array Size - 50,000

Run sizes for MySort

50,000 run size - 1454 milliseconds

25,000 run size - 737 milliseconds

15000 run size - 360 milliseconds

Merge Sort

11 milliseconds

SUMMARY

Firstly, I was deciding between using bubble sort and Insertion Sort for sorting the runs. Although both have the same running time complexities for best, average and worst case, I decided to use Insertion Sort because if the array is mostly sorted, it will perform better.

Overall, through different tests of smaller and larger array sizes, Merge Sort performed better. I tested my sort with different run sizes from its approximate smallest run size to the maximum run size. I observed that going smaller than the approximate smallest run sizes resulted in a Negative Array Size Exception. Although timsort and merge sort have the same worst-case complexity ($O(n \log n)$), merge sort was faster in this case because merge sort with a divide and conquer algorithm, it's better for larger arrays.