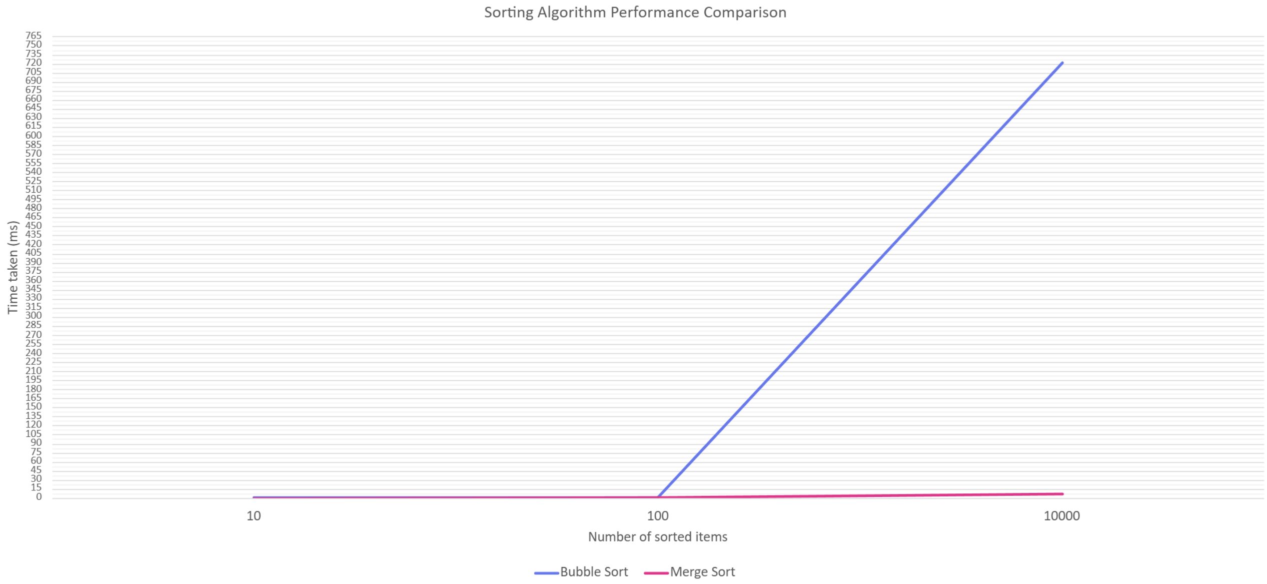
CS1IP Coursework 2

Sort Comparison

Freddie Taylor



The two sorts initially perform similarly - with the 10 and 100 item benchmarks lacking any easily perceivable speed differences, despite the different time complexities. This is because the $O(n \log_2 n)$ complexity of the merge sort and $O(n^2)$ of bubble result in an extremely similar number of worst-case comparison operations at lower item counts

	10 items	100 items	10000 items
Merge Sort	≈ 33 comparisons	≈ 664 comparisons	≈ 132,877
Bubble Sort	≈ 45 comparisons	≈4950 comparisons	≈ 49,995,000

The differences in comparison count where items ≤ 100 between the two sorts shown above are relatively inconsequential to modern CPUs unless performed frequently enough for the differences to compound, as shown by the results all rounding approximately to a millisecond. This suggests it's not always that beneficial to use a more complex, 'faster' sort when you are handling smaller arrays due to the marginal improvements found.

The major performance difference is observed when the number of items increases to 10,000. Due to merge sort's logarithmic time complexity, it performs, at most, 99,867,123 less comparison operations, showing that the performance difference of the two significantly increases as item count increases. This is because logarithmic time complexity's rate of growth decreases as item count grows, while the quadratic complexity of bubble sort has a linear rate of growth.

The practical effect of this is that merge sort is much faster for larger data sets.