Assignment 1: Question 5

Algorithms

November 9th, 2021

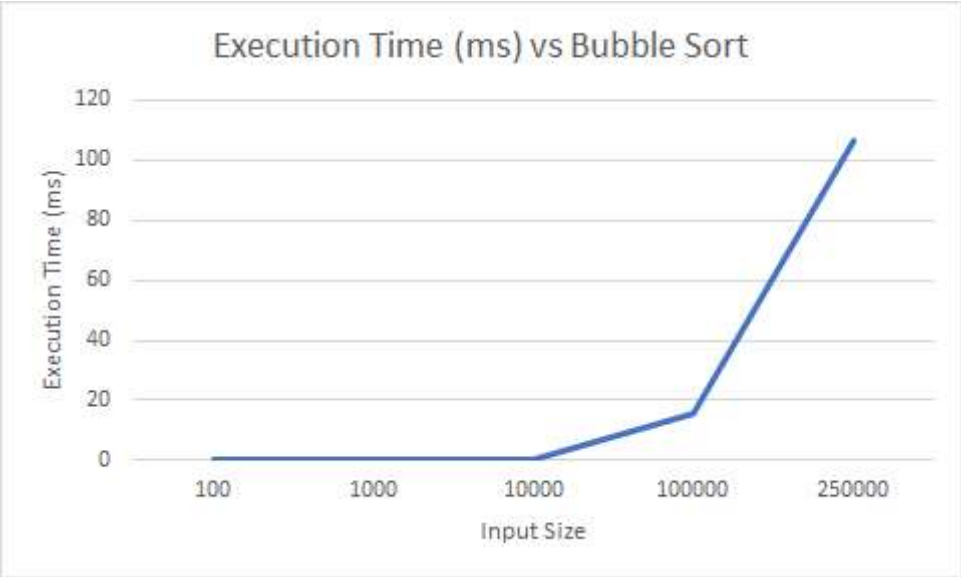
By Tate Donnelly

Comparing the Bubble Sort, Merge Sort, and Quick Sort Algorithms

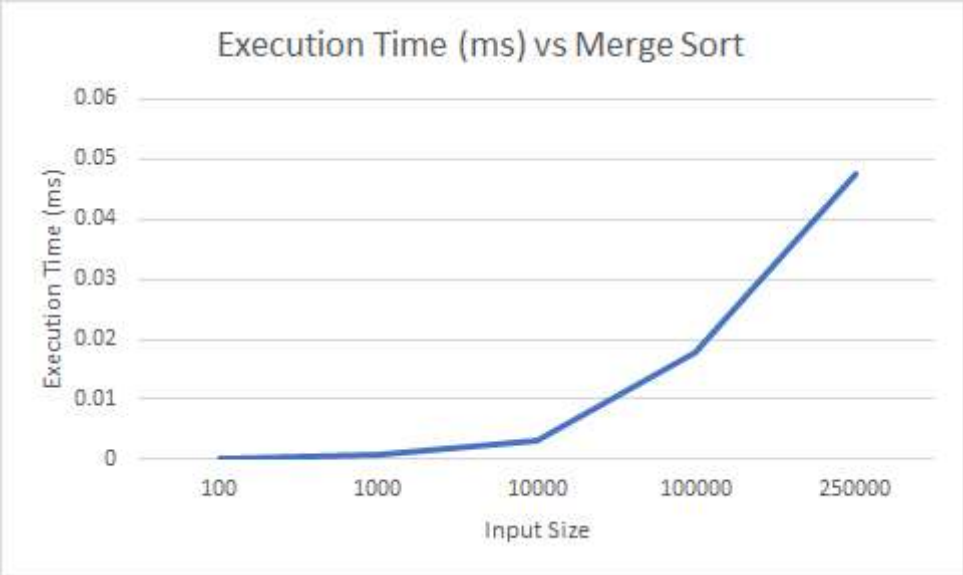
In order to compare the efficiency of bubble sort, merge sort, and quick sort algorithms, I ran a series of experiments where I tested how many milliseconds it took each algorithm to sort a randomly generated array. To get a good picture of how the execution time varied based on each algorithm, I tested each sorting algorithm with multiple input sizes(100, 1000, 10000, 100000, and 250000). I ran each function with each input size 3 times and averaged those trials to produce my graphs to account for variance in the array's data.

When the input size was 100, all of the sorting algorithms took 0 milliseconds. However, as the input size increased, the execution time range between the bubble sorting algorithm and the other algorithms began to increase rapidly as the execution time for the bubble sort algorithm was much longer than the other algorithms. This is best observed in the "Execution Time (ms) vs Bubble Sort, Merge Sort, and Quick Sort Algorithms" graph or by paying careful attention to the y-axis. Comparing each algorithm's graph side-by-side shows that the bubble sort algorithm's graph has a much steeper incline than the other two individual algorithm graphs. The range between the merge sort algorithm and the quick sort algorithm also grew to a lesser extent as the merge sort algorithm began to take longer than the quick sort algorithm. When the input sizes were 10000,100000, and 250000, the merge sort algorithm took about twice as long as the quicksort algorithm. This is seen with the slight difference in the "Execution Time (ms) vs Bubble Sort, Merge Sort, and Quick Sort Algorithms" graph and the slightly steeper incline in the merge sort algorithm's graph when compared with the quick sort algorithm graph. Therefore, the

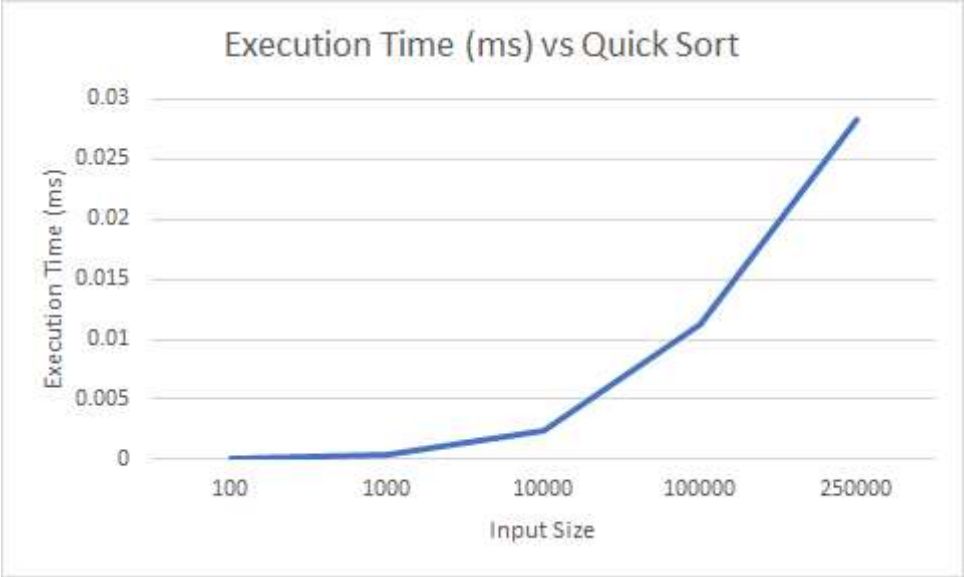
most time-efficient sorting algorithm is the quick sort algorithm, followed closely by the merge sort algorithm, and the least efficient algorithm is the bubble sort algorithm.



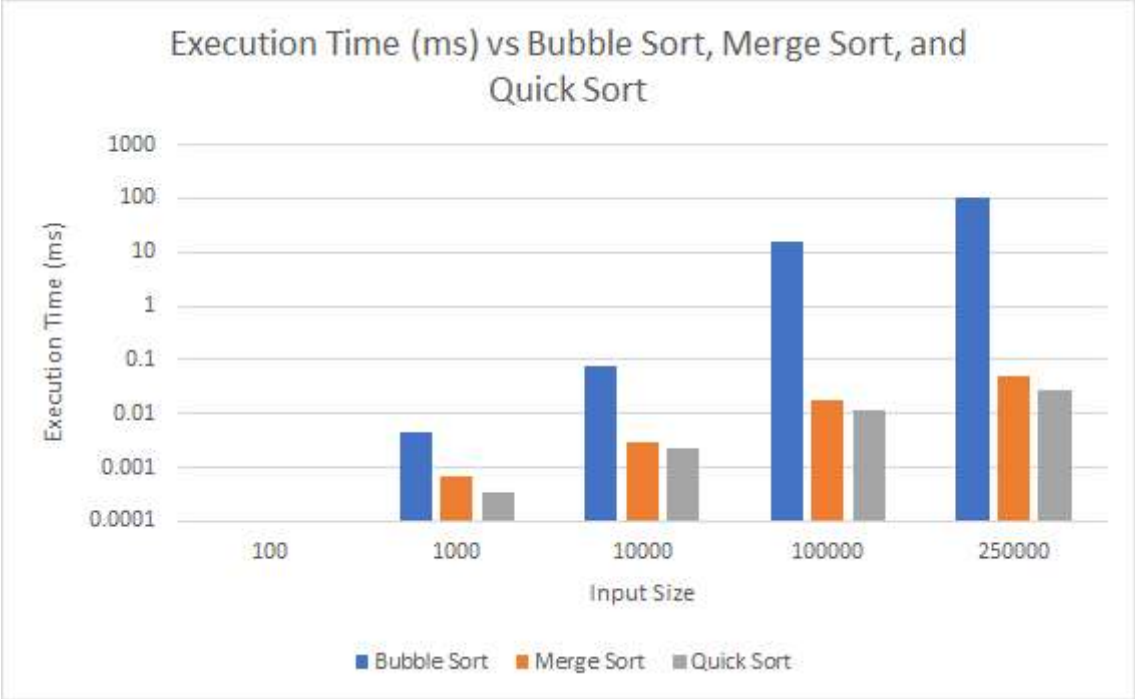
The bubble sort algorithm goes through every element in the array, swapping an element with the one next to it if the one next to it is lower. It does this until every element is ordered.



The merge sort algorithm divides the array into two sections recursively until it each section only has one element. At that point, it begins to combine elements into ordered arrays and then combines those arrays.



The quicksort algorithm shuffles the array and chooses an element to be the partition. It then examines the other elements in the array and attempts to order them so that elements larger than it are on its right and elements smaller than it are on its left.



	Bubble Sort	Merge Sort	Quick Sort
100	0	0	0
1000	0.00433333	0.000666667	0.0003333
10000	0.077	0.003	0.0023333
100000	15.339	0.017666667	0.0113333
250000	106.474	0.047666667	0.0283333