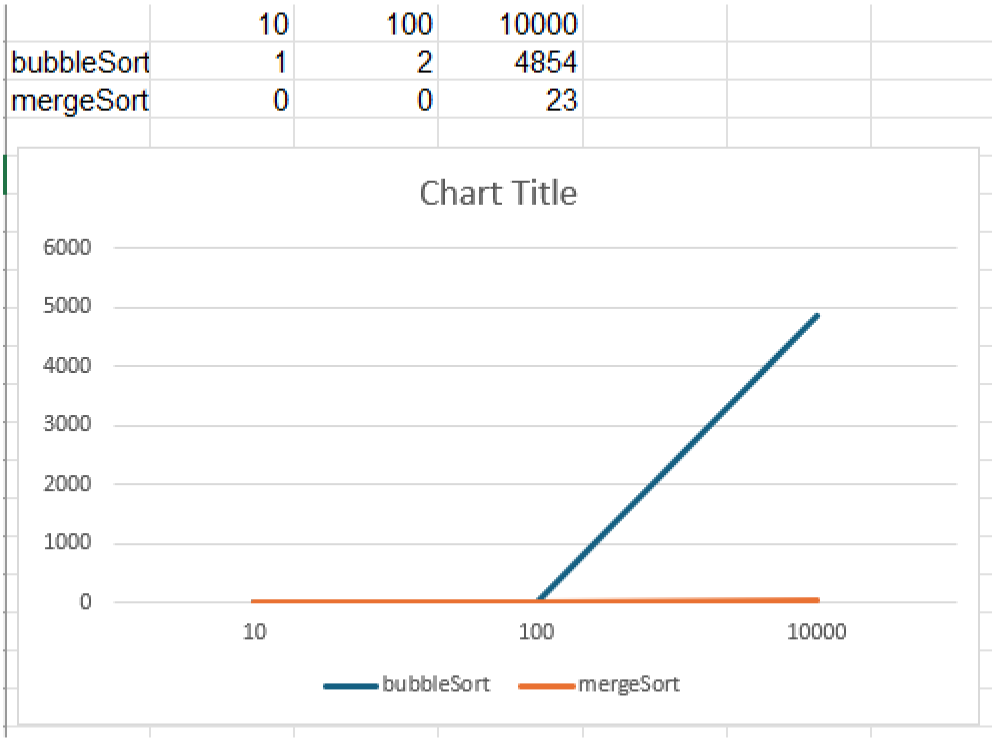
BubbleSort and MergeSort Complexity Reflection:



	10	100	10000
bubbleSort	1	2	4854
mergeSort	0	0	23

This graph represents the time taken (in ms) that it took each bubble and merge sort to complete on 3 different data sets of 10, 100 and 10000 items respectively. We see the expected result that as the amount of data that each of the sorts has to perform on increases, so does the amount of time that it takes them to complete the sort, however most of the times are not very significant due to the relatively small amount of data to be sorted for most of it. The only instance with a time over a 10th of a second (100ms) is bubble sort for 10000 items which is about 4.8 seconds (4800ms).

This is due to bubble sort's poor time complexity, in its best case scenario it has a time complexity of O(n), but both its worst case and average cases are O(n^2). We see bubble sort perform less efficiently than merge on all cases because it is a less efficient algorithm, but this really comes into show for 10000 items where the poor average time complexity of O(n^2) makes it scale poorly with the large increase in the data size, which made it take so much longer than every other sort.

On the other hand, merge sort is a far more efficient sorting algorithm with a time complexity of O(n log n) for all 3 cases of its worst, best and average case. This means that it is able to perform better than bubble sort at all of the given amounts of data, because it is more efficient in general, but it is also far more efficient at scaling with an increase in the data size that needs to be sorted, which is why we see it increase a bit but not nearly by as much when it goes from 100 items to 10000.