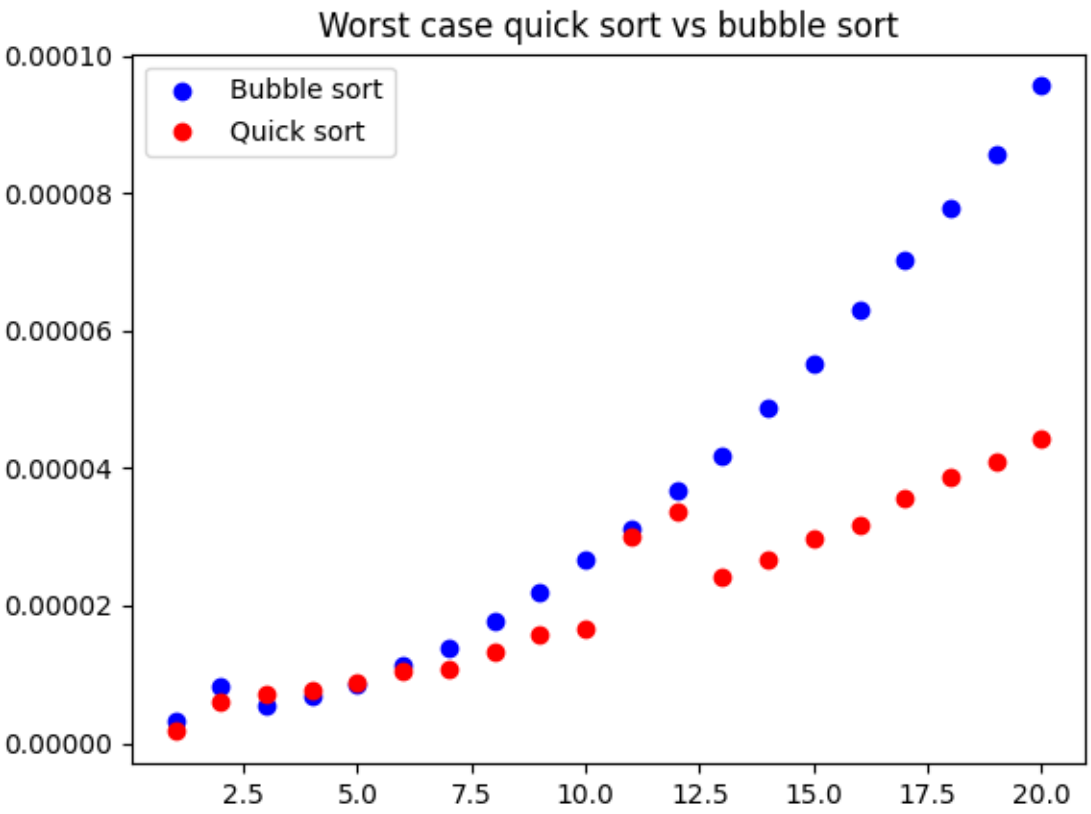
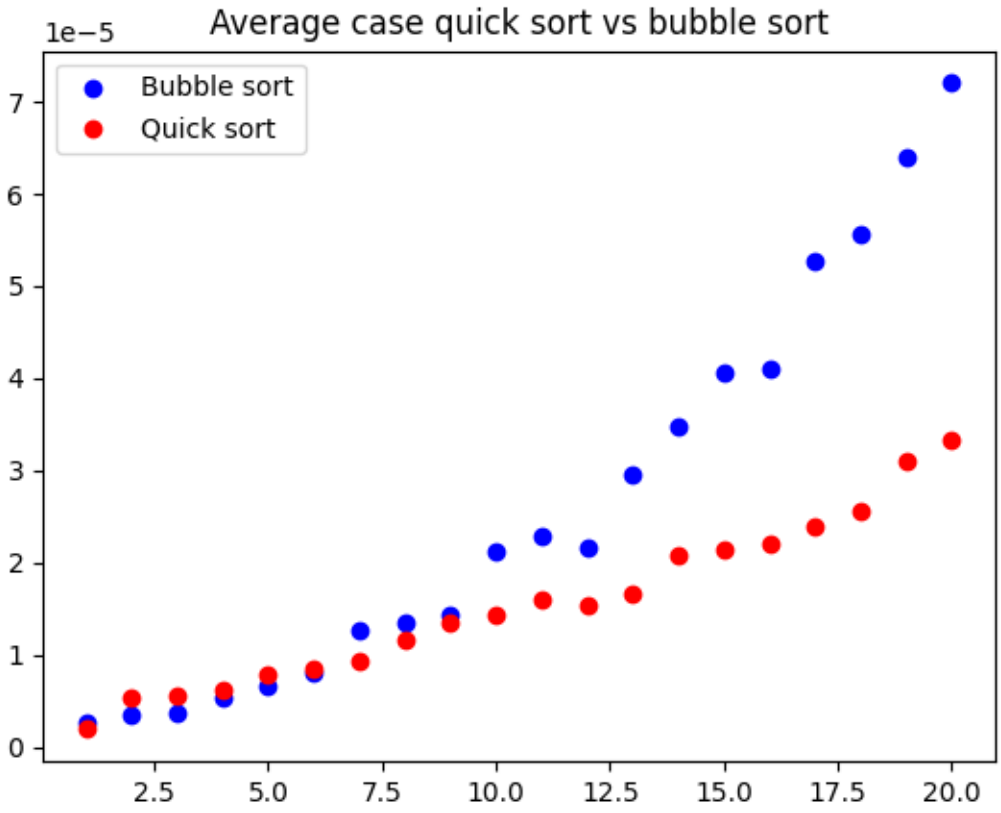
3.

Worst case:



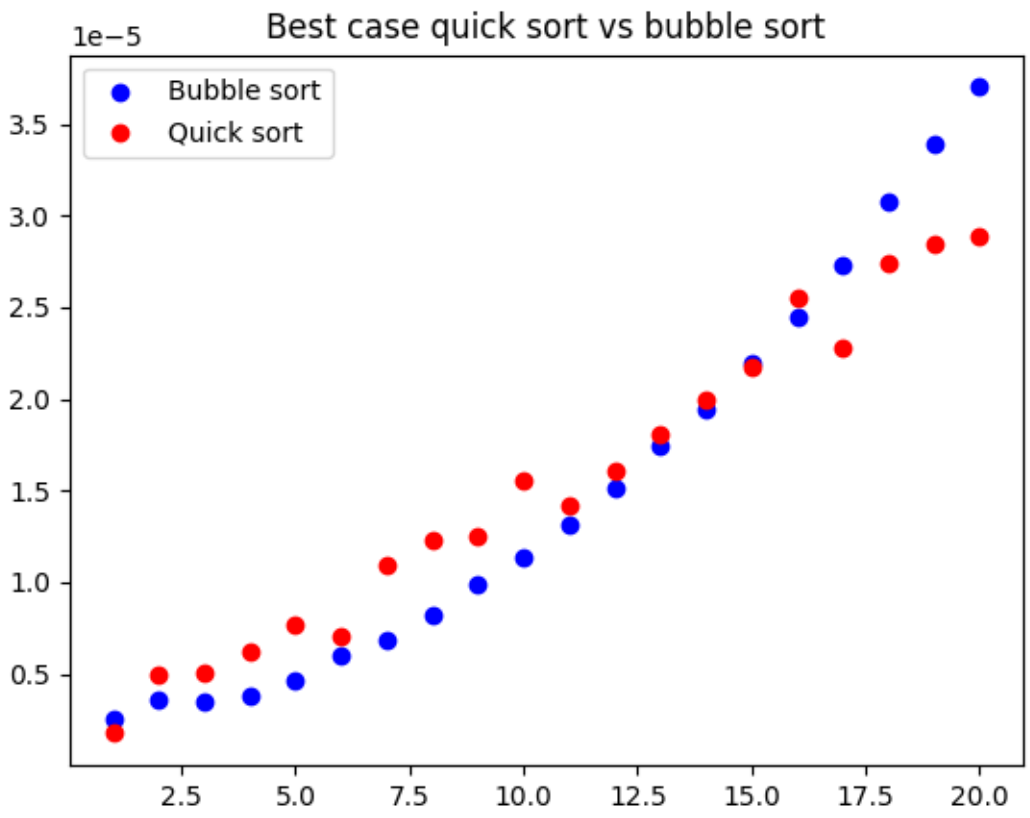
In the worst case for both algorithms, bubble sort and quick sort perform quite similarily up until an input size of around 7, at which point quick sort starts pulling ahead. This is likely due to how bubble sort is in general slower than quick sort and the worst case of a generally slower algorithm will have a more negative effect on the time complexity than a relatively fast algorithm like quick sort.

Average case:



In the average case, bubble sort starts off faster than quick sort up to an input size of 6, at which point the differences are negligible. Afterwards, quick sort starts to very consistently outperform bubble sort. Bubble sort starts off faster due to the additional setup quick sort requires which is not worth it at lower array sizes compared to the simple setup of bubble sort. As the array size increases, this setup starts to become more and more worth it, as bubble sort scales quadratically whereas quick sort scales in log linear time.

Best case:



In the best case for both algorithms, bubble sort consistently outperforms quick sort up until an input size of around 14, after which quick sort scales slower relative to the number of elements in the array and begins significantly outperforming bubble sort. Best case bubble sort only has to iterate through and compare elements without the need to do any swaps. This, in addition to the relative simplicity of the algorithm compared to quick sort, enables it to scale really well on smaller input sizes. On larger inputs, however, quick sort's log linear time complexity begins to significantly outperform bubble sort despite the additional setup required.

4.

Threshold for "small" input: n = 6.

Justification:

I chose 6 because that is the threshold at which quick sort began either outperforming or matching bubble sort in the average and worst cases. The average case is the most important one for our purposes, since it is far more common that a sort function would be called on an unsorted array (average or best case) than a sorted array (worst case). Even in the worst case, quick sort is still generally outperforms bubble sort after the sixth element. On arrays smaller than 6, bubble sort pretty consistently matched or even outperformed quick sort, especially in the best case. It's also more likely for a smaller array to already be sorted than a larger array, so although bubble sort has stellar performance in the best case, the likelihood of that performance being utilized shrinks exponentially as the size of the array increases.