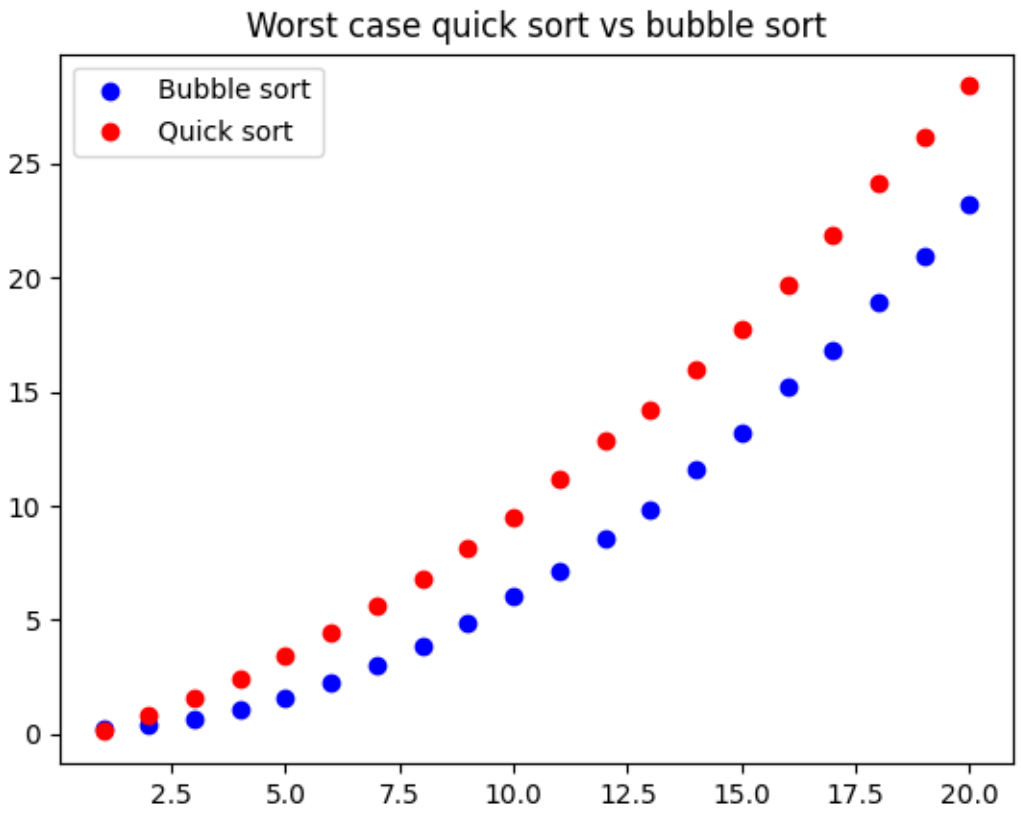
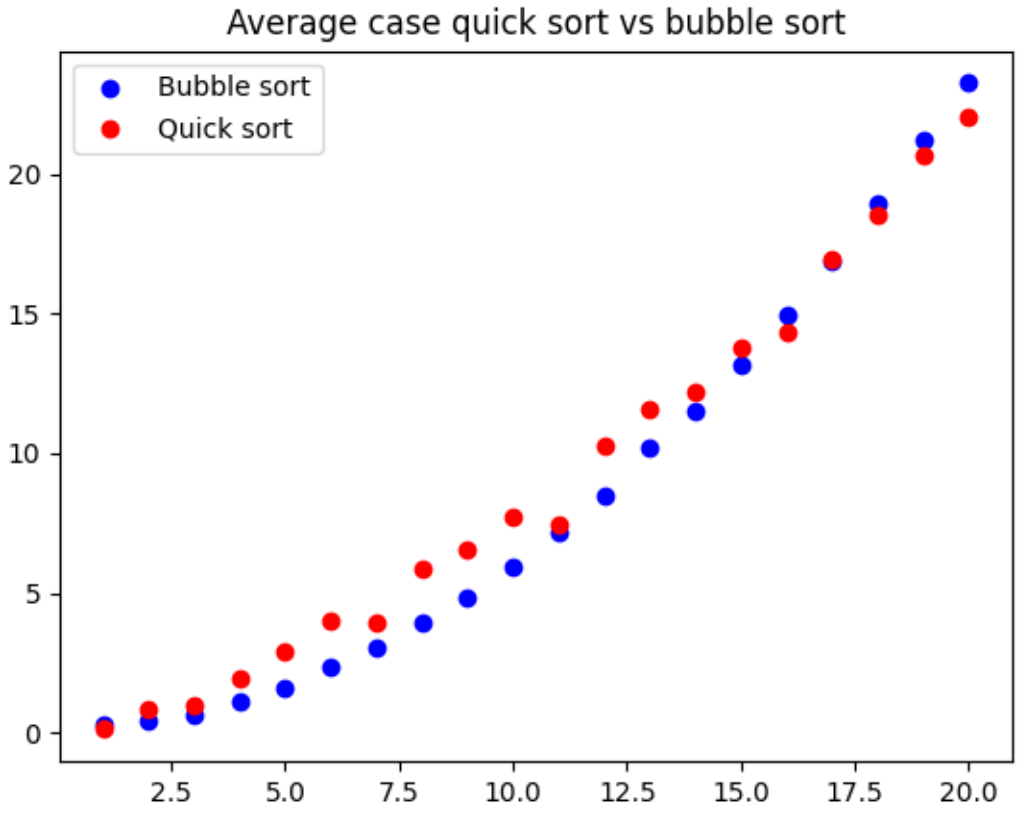
3.

Worst case:



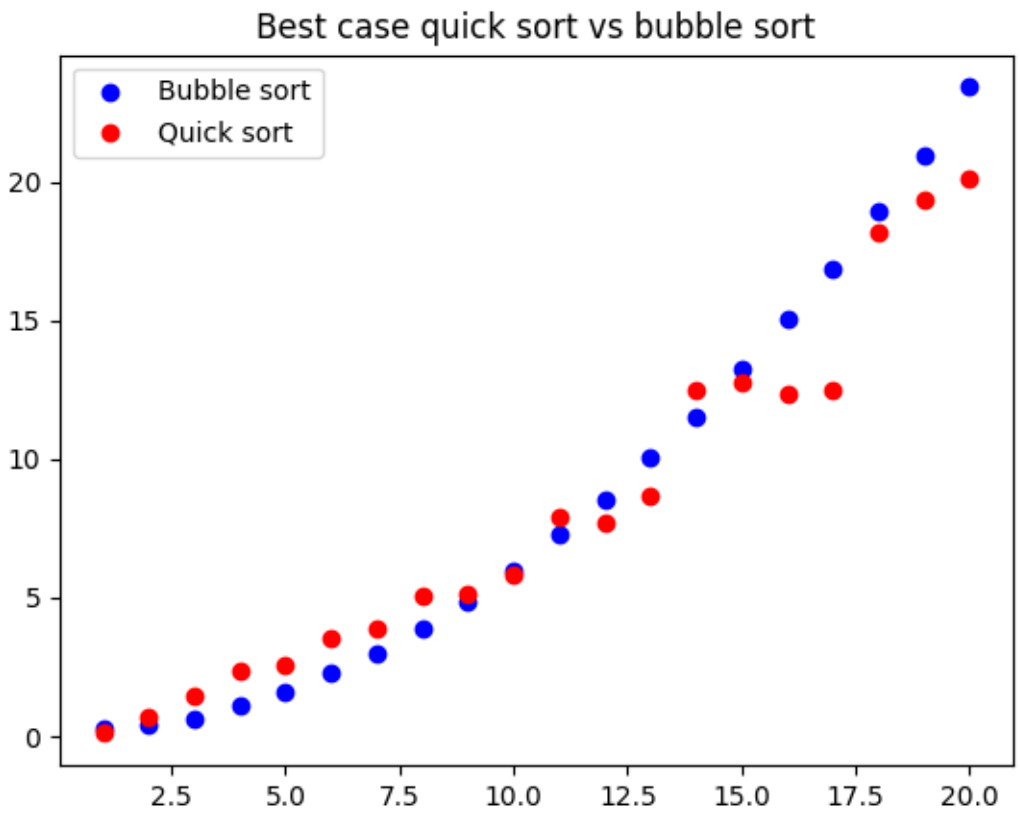
In the worst case for both algorithms bubble sort consistently outperforms quick sort since quick sort has more setup to do than bubble sort and has the same time complexity of O(n^2) (only in worst case).

Average case:



In the average case, quick sort starts to consistently outperform bubble sort at arrays of size 18 and larger. With arrays of smaller sizes, bubble sort consistently outperforms quick sort.

Best case:



In the best case, quick sort starts consistently performing bubble sort at arrays of size 15 and larger. Before then, bubble sort is generally more efficient with a few outliers.

4.

Threshold for "small" input: n = 18.

Justification:

I chose 18 because that is the threshold at which quick sort began outperforming bubble sort in the average case. 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). On arrays smaller than 18, bubble sort pretty consistently outperformed quick sort with the exception of the best case for quick sort, which would be rather infrequent.