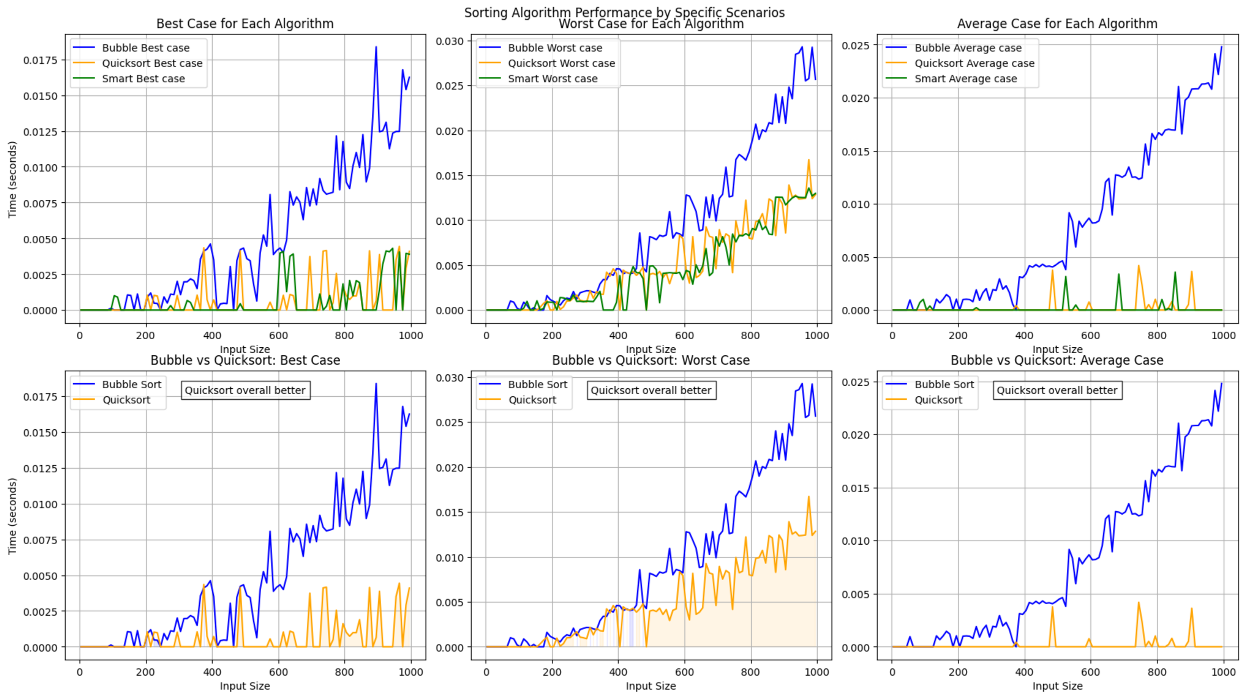
Question 3



Question 4

In the plots above, the threshold for an input as "small" is determined by examining the performance curves of the algorithms. The threshold is set to ***125***, as justified by the observed data trends.

- For the 'best case' scenario, it's observed that Bubble Sort and Quicksort perform nearly identically up to an input size of 125, beyond which Quicksort maintains almost a flat performance curve, efficiently scaling with larger inputs. Bubble Sort, in contrast, shows a noticeable increase in time to do the same sorts. This suggests that Bubble Sort handles "small" inputs efficiently, but as the input size crosses the 125 mark, its performance degrades significantly compared to Quicksort.
- Examining the 'worst case' scenario, even at smaller input sizes, the performance of Bubble Sort is worst past input size of 80. But 80 shouldn't be considered "small" as the difference between the Bubble sort and Quicksort is not that much.
- Lastly, the 'average case' plot reinforces the threshold choice. Although Bubble Sort is performing well up to the 125 mark, beyond this point, its time complexity curve begins rise up relative to Quicksort. Quicksort's time complexity remains relatively stable, reinforcing the idea that 125 is a tipping point from where using a more complex but efficient algorithm like Quicksort becomes apparent.

In conclusion, the input size threshold of 125 effectively captures the transition point in all cases. Below this size, the simpler Bubble Sort algorithm competes closely with Quicksort, beyond this threshold, Quicksort's design allows it to outperform significantly. Validating the choice of 125 as the demarcation line for "small" inputs.