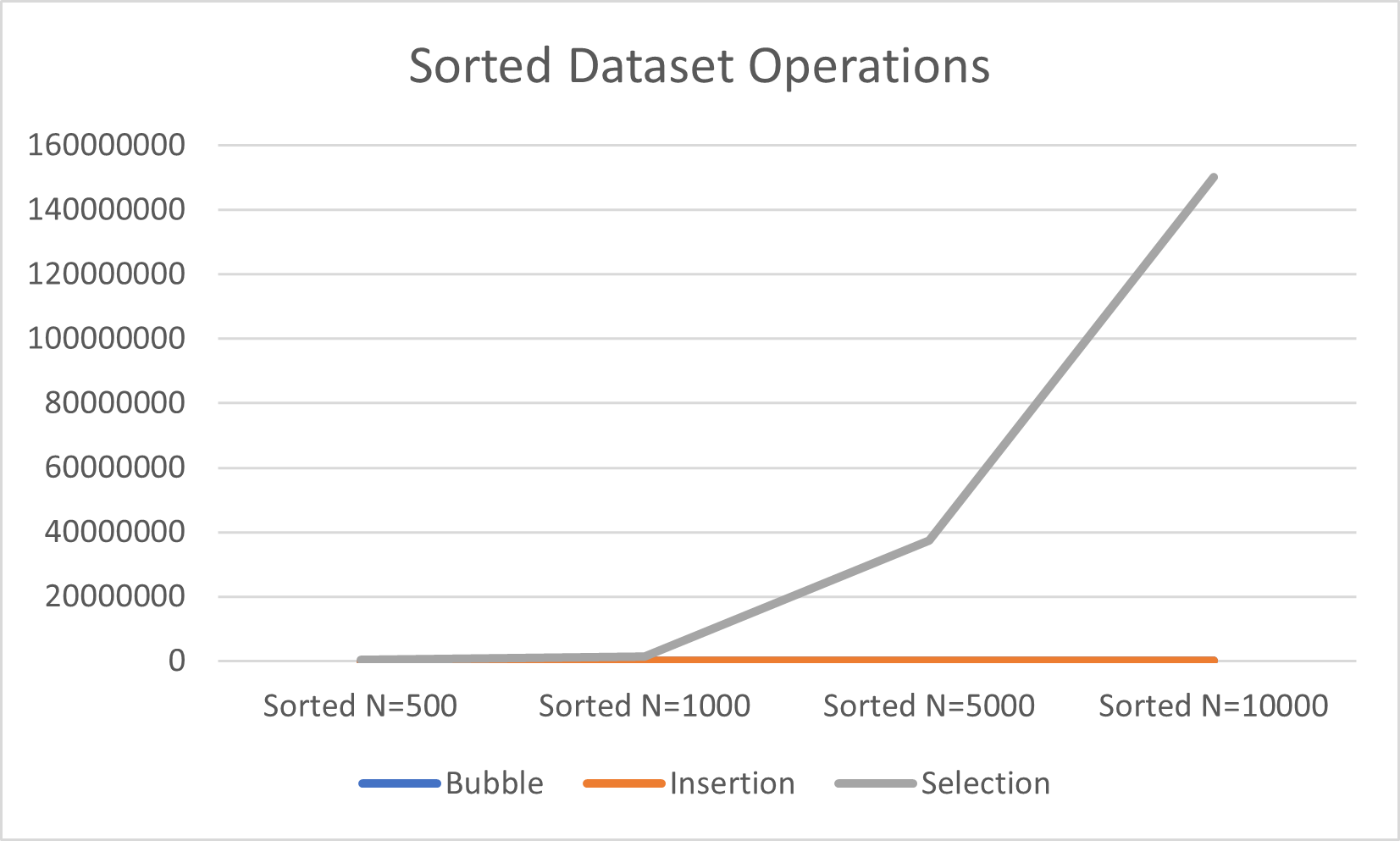
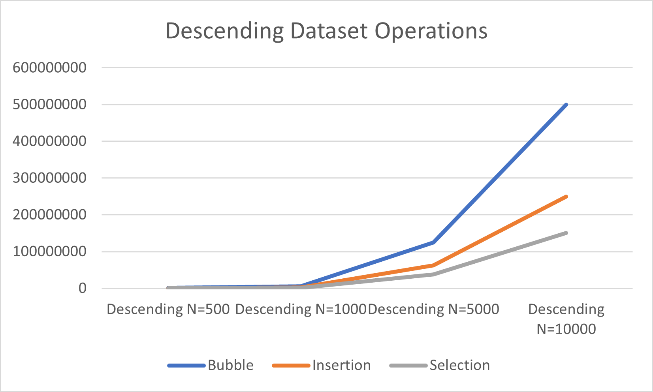
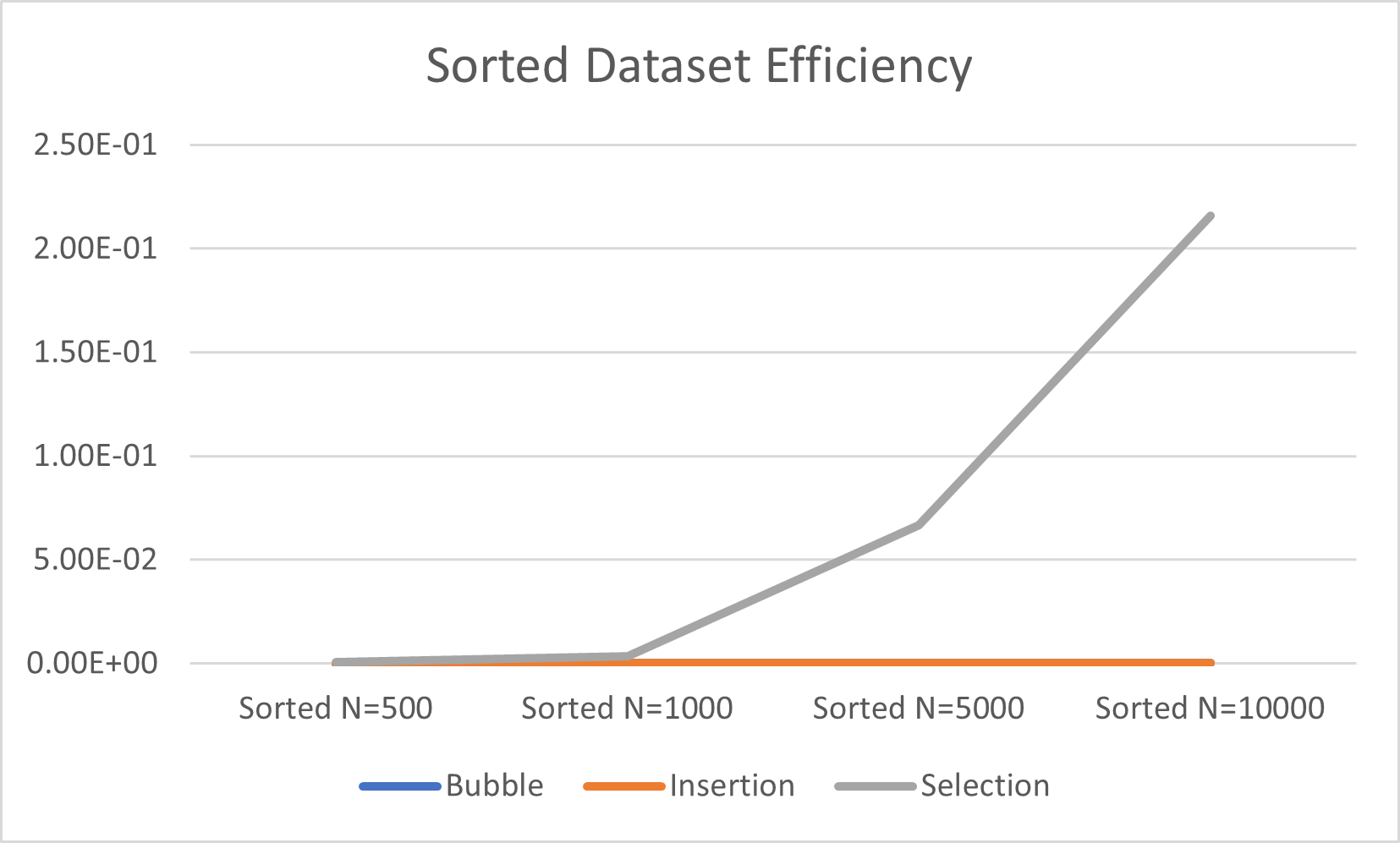
Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

As we can see, the time efficiency and number of operations both follow the same trends, with slight variations as to how fast they climb up. With an already sorted dataset, insertion and bubble are able to end super-fast because they detect it’s all sorted while selection keeps going. However, when it comes to the worst-case scenario (biggest numbers at front, smallest at back), bubble and insertion have to repeat so many more times while selection actually becomes the most efficient. With a random dataset, insertion is more efficient because it brings numbers down the dataset but only to where it fits, it’s like selection sort but shrinks the searching size by more than one every loop, but acts like bubble sort code-wise.