

## Data

	Time to Sort (ms)			
Size	Bubble Sort	Insertion Sort	Quick Sort	Selection Sort
100	44	10	20	28
500	706	158	52	404
1000	2543	632	99	1557

(fastest per size is highlighted)

## Analysis

For smaller data sets, insertion sort seems to be the quickest sorting algorithm at 10 ms with an array size of 100. Despite this, the difference between the algorithms is close, with the next closest, Quick Sort, at 20 ms and the slowest, Bubble Sort, at 44 ms. Although Insertion Sort is shown as the fastest, all times are relatively small, indicating that unless very intensive computing is running, any of the sorting algorithms will work just fine.

For higher data sets, Quick Sort starts to prove that it is the quickest sorting algorithm. I did not expect such high differences between the algorithms. I was aware that Quick Sort would be the fastest because it has the quickest Big O runtime, but I did not realize how drastic this difference was. For data sets of 500 and 1000, Quick Sort remained quicker than any other sorting algorithm, with the margin becoming much greater as the data set increases. By size 1000, Quick Sort does not even break 100ms, while all others show above 500 ms and even above 2000 ms with Bubble Sort. This data shows that for large, intensive sets of data, Quick Sort is the most efficient algorithm. The tradeoffs between the algorithms do not become apparent until you reach large data sets. This is seen very clearly in the size 1000 times.