For Bubble sort, the time it takes to sort the numbers exponential increased from 0 milliseconds at to 2729163 milliseconds at which is equivalent to about 45 minutes.

Insertion sort works the same where it increases exponentially from 0 milliseconds at to 453196 milliseconds at which is equivalent to about 7 and half minutes.

Selection sort decreases the time again from Bubble and Insertion sort where at it is 0 milliseconds and 281067 milliseconds at which is about 4 and half minutes.

Merge sort is where the time to sort really decreases and makes big numbers more reasonable to wait for where numbers is only 204 milliseconds compared to 45 minutes for Bubble sort.

Quick sort is the shortest out the five since it is a non-linear search. While loading the program the first search took 848 milliseconds which is longer then merge sort but then decreases to 135 milliseconds at . After the first run when the JVM warms up it takes a much shorter time to sort then Bubble, Selection and Insertion and Merge Sort.

In Java it was not possible to get the range 1 - for the random number generator even using long arrays. To get the input for that range I took the average for all array inputs. For example to obtain the range 1 - for Bubble sort at numbers where the results where 216 milliseconds, 220 milliseconds and 216 milliseconds I would take the three inputs and divide by three to get the average time. The only one that I did different was Quicksort. Since Quicksort takes a little longer to start up I took the second and third range and took the average of those two to get the 1 - input.