ANALYSIS OF TIME TAKEN

algorithm	tmax/tmin	n ratio	nlog(n) ratio	n^2 ratio	behaviour
SC	9866.15876	100	150	10000	n^2
SS	10019.6936	100	150	10000	n^2
SR	9877.44448	100	150	10000	n^2
IC	10.0136986	10	11.25	100	n
IS	10.5068493	10	11.25	100	n
IR	23117.5555	150	231.6034222	22500	n^2
MC	1493.54488	1000	1500	1000000	nlog(n)
MS	1524.79941	1000	1500	1000000	nlog(n)
MR	1517.27408	1000	1500	1000000	nlog(n)
QC	10247.6867	100	150	10000	n^2
QS	1458.45094	1000	1500	1000000	nlog(n)
QR	1490.53309	1000	1500	1000000	nlog(n)

SELECTION SORT:

For the first algorithm with selection sort on constant input it shows the expected behavior of n^2 ratio matches the tmax/tmin ratio. For the second run on selection sort with sorted input too it shows the expected behavior of n^2. For the third run on the selection sort on random input too it shows the behavior of n^2 which is expected as the time complexity for the best and the worst cases are all n^2 for selection sort.

INSERTION SORT:

For the first run for the insertion sort with constant inputs it runs for O(n) which is expected as the IC only has to traverse the array once. For sorted input on insertion sort too it shows an expected behavior of O(n). For the third run, too on random input it runs for n^2 which is expected as for insertion sort it shows n^2 complexity in worst case complexity.

MERGE SORT:

For the first run on merge sort with constant input mergesort shows a behavior of o(nlogn) which is expected for the best case of mergesort. For the second run with sorted input to the mergesort shows the behavior of o(nlogn) as it still must divide all the elements into a singular size array. For the third run with random input, it still shows the behavior of o(nlogn) as the worst case time complexity of merge sort is also o(nlogn).

QUICK SORT:

For the first run on quicksort with the constant input the behavior shown is n^2 as it is the worst-case time complexity of quicksort probably due to more comparisons. For the second run with sorted input the quicksort it shows the complexity of o(nlogn) which is expected. For the third run with random input Quicksort shows expected behavior of o(nlogn) which is expected as the worst case time complexity for this program.