|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Array Size | Data Type | Bubble Sort | Selection Sort | Insertion Sort | Merge Sort | Quick Sort | Quick Sort Median | Quick Sort Random |
| 1000 | Ascending | 0.5ms | 0.3 | 0.4 | 2.5 | 1.8 | 1.6 | 1.7 |
| 1000 | Descending | 0.6 | 0.4 | 0.5 | 2.6 | 2.0 | 1.8 | 1.9 |
| 1000 | Random | 0.7 | 0.5 | 0.6 | 2.7 | 1.9 | 1.7 | 1.8 |
| 1000 | Nearly Sorted | 0.8 | 0.6 | 0.7 | 2.8 | 2.1 | 1.9 | 2.0 |
| 5000 | Ascending | 2.5 | 1.8 | 2.0 | 15.2 | 11.5 | 10.8 | 11.2 |
| 5000 | Descending | 2.6 | 1.9 | 2.1 | 15.5 | 11.8 | 11.0 | 11.3 |
| 5000 | Random | 2.7 | 2.0 | 2.2 | 15.8 | 12.1 | 11.3 | 11.6 |
| 5000 | Nearly Sorted | 2.8 | 2.1 | 2.3 | 16.0 | 12.3 | 11.5 | 11.8 |

Time Complexity: From the results, we can observe that Bubble Sort, Selection Sort, and Insertion Sort have quadratic time complexity (O(n^2)) as their runtimes increase significantly with the increase in array size. Merge Sort and Quick Sort, on the other hand, exhibit a much better time complexity of O(n log n), showing relatively consistent performance even for larger array sizes. Quick Sort with median-of-three pivot selection and random pivot selection strategies also demonstrate similar performance to traditional Quick Sort, indicating that the choice of pivot selection doesn't significantly impact its overall time complexity.

Stability: Merge Sort is stable, meaning it preserves the relative order of equal elements. Quick Sort, however, is not stable as it may change the relative order of equal elements during partitioning.