|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Time taken for sorting algorithms to sort arrays of different sizes (in seconds) | | | | | | | | | | | | | | | | |
| Array to Sort | Descending Array | | | | Random Array | | | | Nearly Sorted Array | | | | Average | | | |
| Size: | 10 | 50 | 100 | 1000 | 10 | 50 | 100 | 1000 | 10 | 50 | 100 | 1000 | 10 | 50 | 100 | 1000 |
| MergeSort | 0.00144 | 0.00408 | 0.00864 | 0.08783 | 0.00393 | 0.01271 | 0.02089 | 0.20551 | 0.00292 | 0.00661 | 0.00594 | 0.05345 | 0.00276333 | 0.0078 | 0.011823 | 0.115597 |
| QuickSort | 0.00177 | 0.00407 | 0.00491 | 0.06662 | 0.00314 | 0.01264 | 0.0259 | 0.24824 | 0.00232 | 0.00573 | 0.00566 | 0.04422 | 0.00241 | 0.00748 | 0.012157 | 0.119693 |
| QuickSort3 | 0.00273 | 0.00403 | 0.00749 | 0.04654 | 0.00408 | 0.01268 | 0.02194 | 0.23956 | 0.00208 | 0.00704 | 0.00706 | 0.06276 | 0.00296333 | 0.007917 | 0.012163 | 0.116287 |
| QuickSortR | 0.00255 | 0.00503 | 0.00642 | 0.04511 | 0.00474 | 0.01319 | 0.01765 | 0.22908 | 0.00128 | 0.00708 | 0.00792 | 0.04421 | 0.00285667 | 0.008433 | 0.010663 | 0.106133 |

From comparison, all the sorting methods and their time to complete sorting is very similar suggesting that they share a similar time complexity.

* **Merge Sort:**  On average merge sort was the most consistent sorting method across all sizes of arrays being the second best on average for each size (10, 50, 100, 1000).
* **Quick Sort:** The fastest for small size arrays but the worst for large arrays. Therefore, it is not as scalable as the other methods.
* **Quick Sort Random Pivot:** The fastest for large size arrays but the worst for small arrays. Therefore, it is better at a larger scale and can be considered as more scalable than the other methods.
* **Quick Sort Median of 3:** On average this sort was worst of the sorting method across all sizes of arrays being the slowest or second slowest on average for each size (10, 50, 100, 1000).