Program Structure and Algorithm

Assignment – 6

Name – Surbhi Soni

NUID - 002969578

1. With Instrumentation

Merge Sort:

|  |  |  |  |
| --- | --- | --- | --- |
| Array Size(N) | Compares | Swaps | Time(ms) |
| 10000 | 1435382 | 9825 | 19.740973 |
| 20000 | 3100617 | 19551 | 33.994712 |
| 40000 | 6661096 | 39068 | 130.925039 |
| 80000 | 14242292 | 78292 | 507.28352 |
| 160000 | 30324350 | 156487 | 7276.439012 |

QuickSort\_DualPivot:

|  |  |  |  |
| --- | --- | --- | --- |
| Array Size(N) | Compares | Swaps | Time(ms) |
| 10000 | 505767024 | 276874 | 31.178131 |
| 20000 | 1406588520 | 559201 | 71.962909 |
| 40000 | 8565748608 | 1179188 | 423.668675 |
| 80000 | 26047049879 | 2094233 | 2531.208043 |

Heap Sort:

|  |  |  |  |
| --- | --- | --- | --- |
| Array Size(N) | Compares | Swaps | Time(ms) |
| 10000 | 5613849 | 3015609 | 73.838712 |
| 20000 | 12151597 | 6488245 | 325.862054 |
| 40000 | 26153371 | 13912509 | 1573.259108 |
| 80000 | 56018178 | 29683444 | 7708.683166 |
| 160000 | 119464150 | 63012526 | 33311.911118 |

2.) Without Instrumentation

Merge Sort:

|  |  |
| --- | --- |
| Array Size(N) | Time(ms) |
| 10000 | 21.343862 |
| 20000 | 38.357029 |
| 40000 | 135.767793 |
| 80000 | 501.631524 |
| 160000 | 6535.926312 |

QuickSort\_DualPivot:

|  |  |
| --- | --- |
| Array Size(N) | Time(ms) |
| 10000 | 13.430804 |
| 20000 | 55.855717 |
| 40000 | 169.068197 |
| 80000 | 1756.000949 |

Heap Sort:

|  |  |
| --- | --- |
| Array Size(N) | Time(ms) |
| 10000 | 2.413729 |
| 20000 | 3.260404 |
| 40000 | 7.489908 |
| 80000 | 12.525499 |
| 160000 | 20.826627 |

Conclusion:

Based on the results, it appears that for Merge Sort and QuickSort\_DualPivot, the number of comparisons and swaps are the best predictors of total execution time. This is because as the size of the input array increases, the number of comparisons and swaps required to sort the array also increases, leading to an increase in execution time.

For Merge Sort, the number of comparisons and swaps are significantly larger than the other two algorithms for all input sizes, which suggests that these factors are the dominant predictors of execution time.

For QuickSort\_DualPivot, the number of comparisons and swaps also increase as the size of the input array increases, but they are not as dominant as in Merge Sort. This could be due to the fact that QuickSort\_DualPivot has a more efficient partitioning scheme than other quicksort implementations, which reduces the number of swaps and comparisons required.

Based on the provided results, we can say that for all three sorting algorithms (Merge Sort, QuickSort\_DualPivot, and Heap Sort), the time taken for sorting (measured without instrumentation) is the best predictor of total execution time. This is because as the size of the input array increases, the time taken for sorting also increases.