PSA Assignment 6:

Name: Yash Pawar

NUID: 002747371

***In this assignment, your task is to determine--for sorting algorithms--what is the best predictor of total execution time: comparisons, swaps/copies, hits (array accesses), or something else. You will run the benchmarks for merge sort, (dual-pivot) quick sort, and heap sort. You will sort randomly generated arrays of between 10,000 and 256,000 elements (doubling the size each time). If you use the SortBenchmark, as I expect, the number of runs is chosen for you. So, you can ignore the instructions about setting the number of runs. For each experiment (a sort method of a given size), you will run it twice: once for the instrumentation, once (without instrumentation) for the timing.***

**Conclusion:**

A comparative analysis of Merge Sort, Quick Dual Pivot sort, and Heap Sort was conducted, using arrays of various lengths (N) and measuring the number of compares, swaps, and execution time (T). The results show that the ***number of compares*** is the best indicator of execution time.

I got to this conclusion after analyzing Log-Log graphs of N vs. T and N vs. compares, which indicated that the Comparison values are highly correlated with the execution time (T).

Conversely, the Log-Log graph of N vs. T and N vs. Swaps yielded significantly different results for all three sorting techniques. Thus, the evidence suggests that ***comparisons*** offer the best predictor of execution time among the three sorting techniques.

For Heap Sort:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Heap Sort 20 runs** | | | | | | | |
| **Array Size** | **Compares** | **Log(Compares)** | **Swaps** | **Log(Swaps)** | **Hits** | **Time (ns)** | **Log(Time)** |
| 10000 | 5611727 | 15.54036907 | 3010268 | 14.91753967 | 23264526 | 299.799294 | 5.70311323 |
| 20000 | 12151730 | 16.3129821 | 6485735 | 15.68511571 | 50246400 | 1295.6194 | 7.16674416 |
| 40000 | 26156587 | 17.07961161 | 13911666 | 16.44823833 | 107959838 | 5596.9936 | 8.62998488 |
| 80000 | 56017321 | 17.8411715 | 29678848 | 17.20594516 | 230750034 | 22901.0073 | 10.0389362 |
| 160000 | 76017433 | 18.14647325 | 55178848 | 17.82609025 | 510750034 | 45901.0073 | 10.7342423 |

Chart, line chart

Description automatically generated

For Merge Sort:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Merge Sort 20 runs** | | | | | | | |
| **Array Size** | **Compares** | **Log(Compares)** | **Swaps** | **Log(Swaps)** | **Hits** | **Time (ns)** | **Log(Time)** |
| 10000 | 1946347 | 14.48146484 | 9715 | 9.181426362 | 5223632 | 0.6358375 | -0.4528123 |
| 20000 | 4147890 | 15.23811033 | 19514 | 9.878887436 | 11327556 | 0.45551875 | -0.7863184 |
| 40000 | 8987454 | 16.01134016 | 39071 | 10.57313578 | 24415274 | 0.50321875 | -0.6867303 |
| 80000 | 19247359 | 16.77288441 | 78148 | 11.26635974 | 52350546 | 0.82983125 | -0.1865329 |
| 160000 | 41054999 | 17.53042316 | 156348 | 11.95983957 | 111740906 | 1.2005625 | 0.1827902 |

Chart, line chart

Description automatically generated

For QuickSort Dual Pivot:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Quick Sort Dual Pivot 20 runs** | | | | | | | |
| **Array Size** | **Compares** | **Log(Compares)** | **Swaps** | **Log(Swaps)** | **Hits** | **Time (ns)** | **Log(Time)** |
| 10000 | 18607 | 9.831293133 | 5861 | 8.676075516 | 42054 | 1.36486875 | 0.31105827 |
| 20000 | 38570 | 10.56023005 | 11688 | 9.366317953 | 85325 | 6.9749271 | 1.94232188 |
| 40000 | 58759 | 10.98119961 | 29901 | 10.3056472 | 178366 | 7.48161875 | 2.01244918 |
| 80000 | 117292 | 11.67242183 | 61434 | 11.02571871 | 363029 | 15.6632646 | 2.75131814 |
| 160000 | 203279 | 12.2223347 | 135547 | 11.81707372 | 745468 | 96.0446979 | 4.56481369 |