Akash Bharadwaj Karthik (002787011)

**Program Structures & Algorithms**

**Spring 2023 (Sec -8)**

**Assignment No. 6**

**Task**

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.

1. 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.
2. 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.
3. You must support your (clearly stated) conclusions with evidence from the benchmarks (you should provide log/log charts and spreadsheets typically).

**Relationship Conclusion:**

**Swaps are the best predictor for execution time for Merge Sort:**

Highest value observed in swaps for Merge sort both with and without instrumentation;

Highest value of 0.9486 observed with instrumentation and 0.9351 without instrumentation

**Hits and swaps** are good predictors for execution time of **Heap sort** **with instrumentation**, and **compares** are good predictors when **not using instrumentation.**

Highest value of **0.975 observed with instrumentation** and **0.9888 without instrumentation**

**Hits and Swaps** are good predictors for execution time for **Quick Sort with instrumentation, and swaps and compares** are good predictors **without instrumentation.**

Highest value of **0.9772 observed with instrumentation** and **0.9716 without instrumentation**

**Metrics used for benchmarking:**

**Array sizes:** 10,000 to 160,000 using doubling method.

**Sort algorithms**: **QuickSortDualPivot** , **BasicMergeSort** and **HeapSort**

**Evidence used to support conclusion:**

***BasicMergeSort Table:***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| N | lg(compares) | lg(Swaps) | lg(Hits) | lg(Copies) | lg(Time w/o instrumenting) | lg(Time w instrumenting) |
| 10000 | 16.89065675 | 13.251491 | 18.9017 | 17.747144 | 2.570462931 | 3.042644337 |
| 20000 | 18.00466306 | 14.251219 | 20.0149 | 18.87267488 | 4.063502942 | 3.584962501 |
| 40000 | 19.1104969 | 15.252797 | 21.1201 | 19.9881521 | 4.146492307 | 3.937344392 |
| 80000 | 20.2090035 | 16.253436 | 22.218 | 21.0950673 | 4.960233672 | 5.094658343 |
| 160000 | 21.30122112 | 17.252595 | 23.3096 | 22.19460298 | 6.372081484 | 6.238022518 |

***DualPivotQuickSort Table:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | lg(compares) | lg(Swaps) | lg(Hits) | lg(Time w/o instrumenting) | lg(Time w instrumenting) |
| 10000 | 17.25395484 | 15.987986 | 18.6767 | 1.726831217 | 2.176322773 |
| 20000 | 18.36776615 | 17.100775 | 19.7892 | 2.321928095 | 2.599317794 |
| 40000 | 19.48965891 | 18.215648 | 20.906 | 3.313245852 | 3.798050515 |
| 80000 | 20.59370942 | 19.304238 | 21.9999 | 4.523561956 | 4.925999419 |
| 160000 | 21.69093276 | 20.414587 | 23.1047 | 6.098874287 | 6.19061486 |

***HeapSort Table***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N | lg(compares) | lg(Swaps) | lg(Hits) | lg(Time w/o instrumenting) | lg(Time w instrumenting) |
| 10000 | 17.84457258 | 16.922223 | 19.8839 | 1.744161096 | 1.887525271 |
| 20000 | 18.96225681 | 18.034021 | 20.9986 | 2.853995647 | 3.898208353 |
| 40000 | 20.07103212 | 19.137749 | 22.1048 | 4.193771743 | 4.544732656 |
| 80000 | 21.17217892 | 20.234435 | 23.2036 | 5.13996057 | 5.792074462 |
| 160000 | 22.26669099 | 21.325099 | 24.2962 | 6.990387652 | 7.676027579 |

***Graphical Representation***

**Highest**  value observed in **swaps for Merge sort** both with and without instrumentation;

Highest value of **0.9486 observed with instrumentation** and **0.9351 without instrumentation**

**Highest**  value observed in **hits**/**swaps for Quick Sort** **with instrumentation** and **swaps/compare** **without instrumentation**;

Highest value of **0.9772 observed with instrumentation** and **0.9716 without instrumentation**

**Highest**  value observed in **hits**,**swaps and compares equally for Heap Sort** **with instrumentation** and **compares** **without instrumentation**;

Highest value of **0.975 observed with instrumentation** and **0.9888 without instrumentation**

***Console Output:***

***Text

Description automatically generated***

***A computer screen capture

Description automatically generated with low confidence***

***A computer screen capture

Description automatically generated with low confidence***

***A computer screen capture

Description automatically generated with low confidence***

***A picture containing text, outdoor

Description automatically generated***