Program Structures and Algorithms
Spring 2023(SEC 1)

NAME: Harshil Shah
NUID: 002780887
Assignment: 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.

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.

Evidence:

- **MergeSort**

Size	Compares	Swaps	Hits	Copies	Normalized Time
10000	1.319	0.106	5.318	2.389	3.43
20000	1.328	0.099	5.35	2.423	3.44
40000	1.335	0.092	5.377	2.454	3.38
80000	1.342	0.086	5.401	2.48	4.09
160000	1.348	0.081	5.423	2.504	4.24
256000	1.35	0.07	5.448	2.543	4.31

- **Dual Pivot QuickSort**

Size	Compares	Swaps	Hits	Copies	Normalized Time
10000	1.695	0.719	4.589	0	3.45
20000	1.718	0.716	4.607	0	3.4
40000	1.743	0.704	4.598	0	3.46
80000	1.759	0.726	4.686	0	3.75
160000	1.751	0.706	4.636	0	3.86
256000	1.765	0.736	4.729	0	4.01

- **HeapSort**

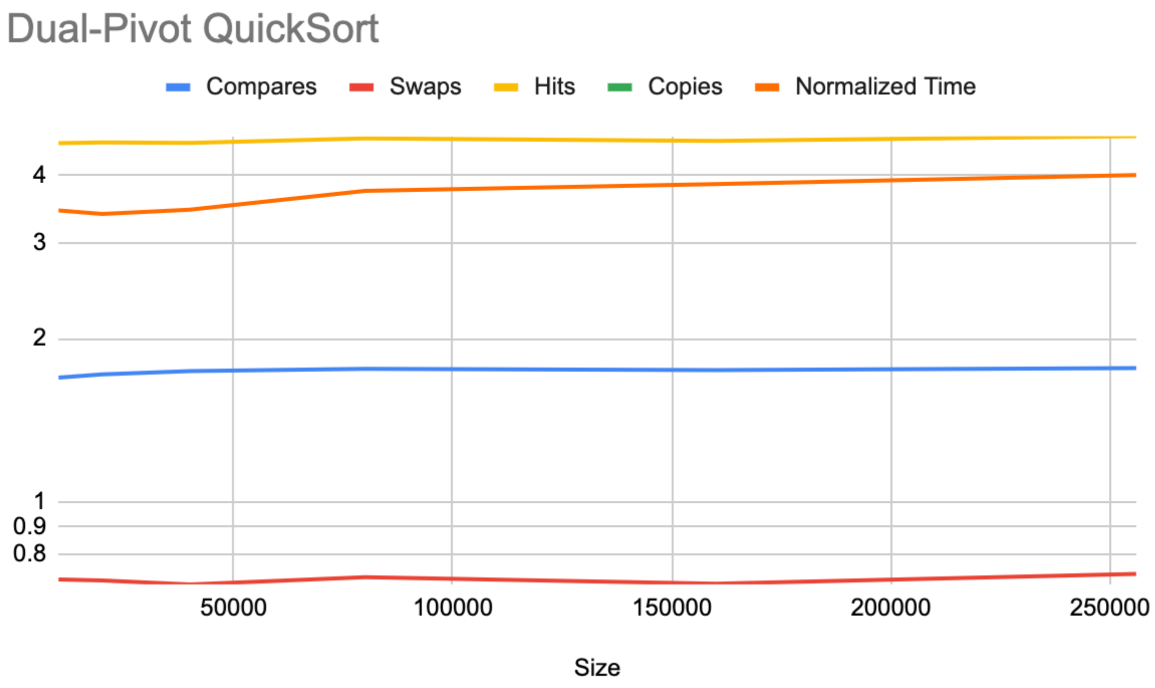
Size	Compares	Swaps	Hits	Copies	Normalized Time

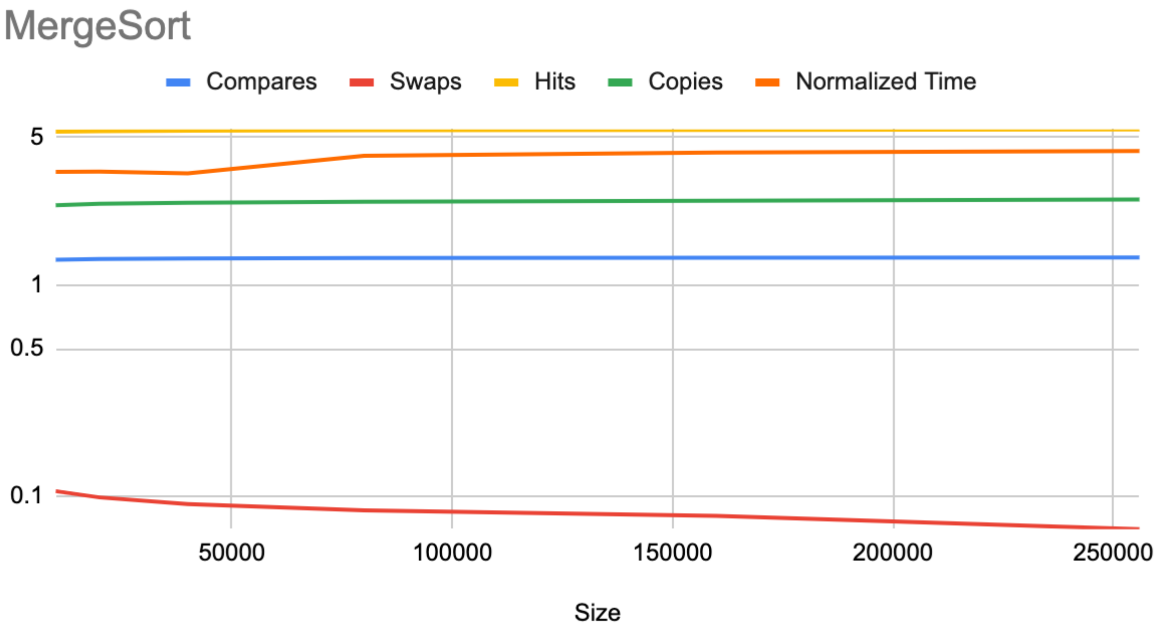
10000	2.556	1.349	10.505	0	3.97
20000	2.579	1.355	10.578	0	4.08
40000	2.599	1.361	10.641	0	4.16
80000	2.616	1.366	10.696	0	4.74
160000	2.632	1.37	10.745	0	4.93
256000	2.638	1.371	10.762	0	5.14

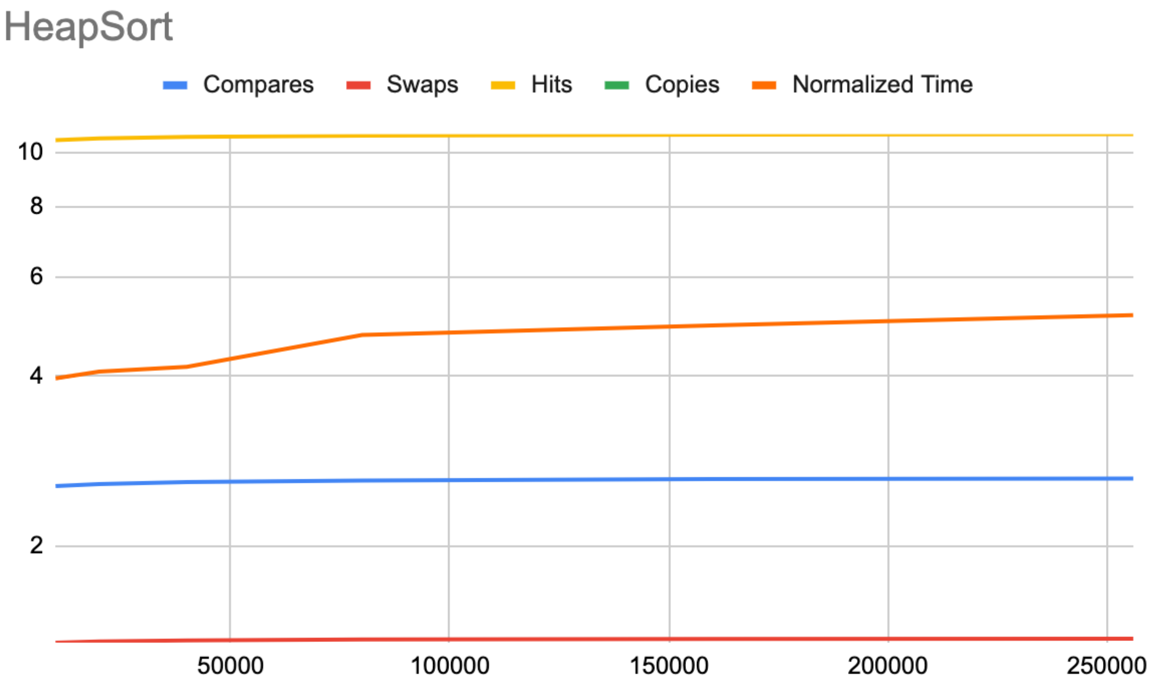
Observation:

1. **Dual Pivot Quicksort** -: It appears that the **number of swaps** has the most significant impact on the algorithm's performance. As the input size increases, the number of swaps also increases, and the normalized time increases accordingly. This suggests that the number of swaps is a good predictor of total execution time for dual pivot quicksort. Comparisons and hits also show a slight increase in execution time with respect to the input size, but the effect is not as significant as the number of swaps.
2. **MergeSort** -: The evidence shows that as the size of the input data increases, the number of copies and hits also increase, which is expected because more copies are needed to create temporary arrays for larger input sizes and more data needs to be accessed during the sorting process. Therefore, the number of **copies and hits** can serve as reliable predictors for the execution time of mergesort algorithm.
3. **HeapSort** -: We can see that the **number of comparisons** increases as the input size increases. The other metrics, such as swaps, hits, and copies, remain relatively constant at 1.36 or lower. Therefore, the number of comparisons is the most significant factor affecting the total execution time, as the other metrics do not increase significantly with increasing input size. Furthermore, we can see that the normalized time, which is a measure of the total execution time per input element, increases with input size. This is expected since the number of comparisons also increases with input size. Therefore, we can conclude that the number of comparisons is the best predictor of total execution time for heapsort.

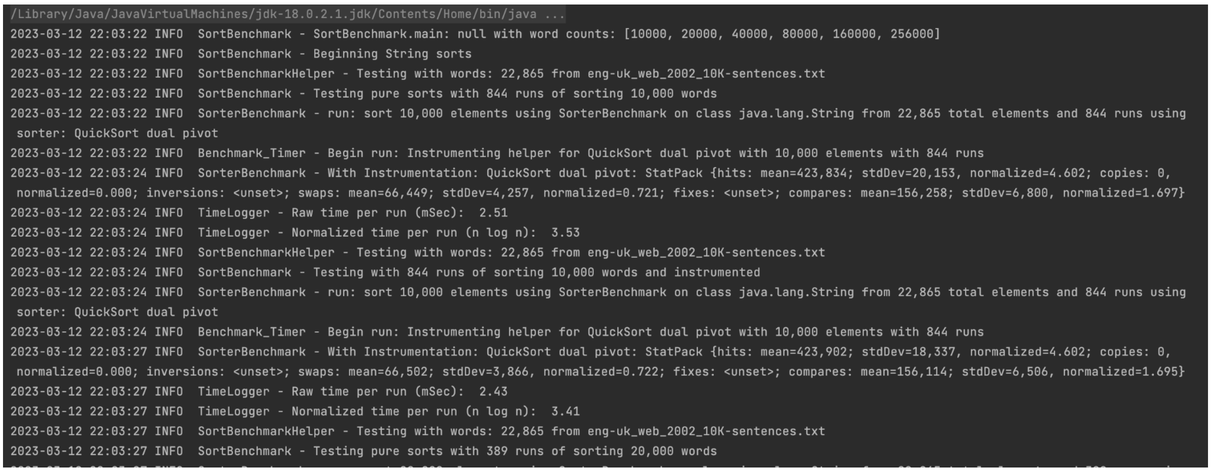
Graphical Representation:


Dual-Pivot QuickSort
Compares
Swaps
Hits
Copies
Normalized Time
4
3
2
1
0.9
0.8
50000
100000
150000
200000
250000
Size

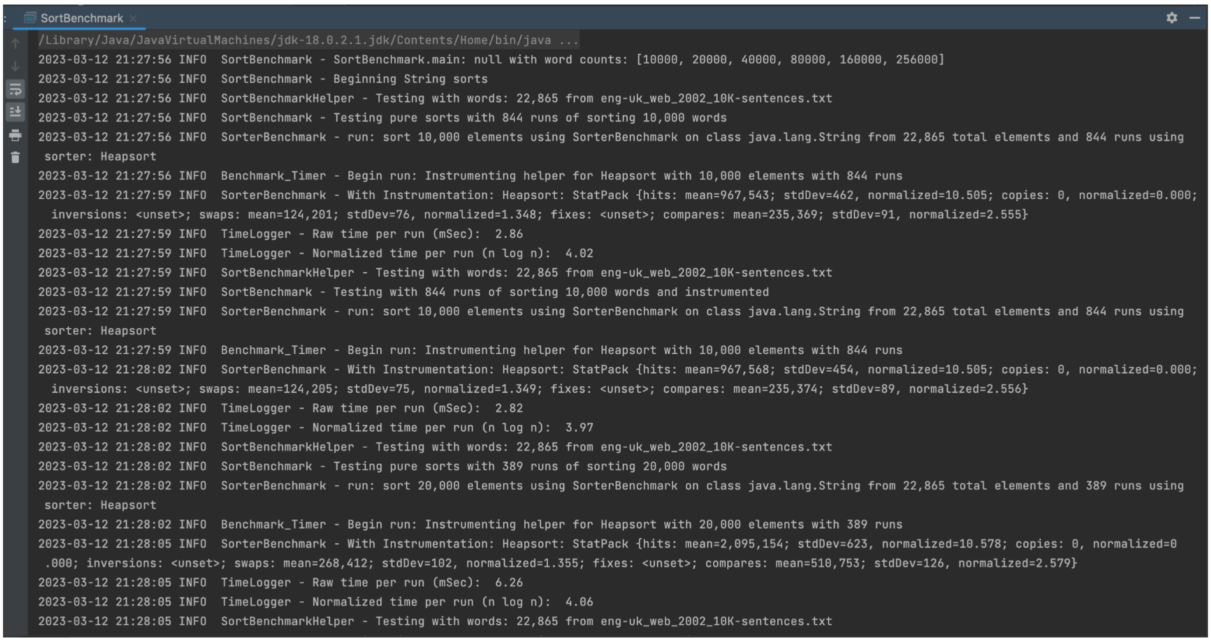

MergeSort
Compares
Swaps
Hits
Copies
Normalized Time
5
1
0.5
0.1
50000
100000
150000
200000
250000
Size



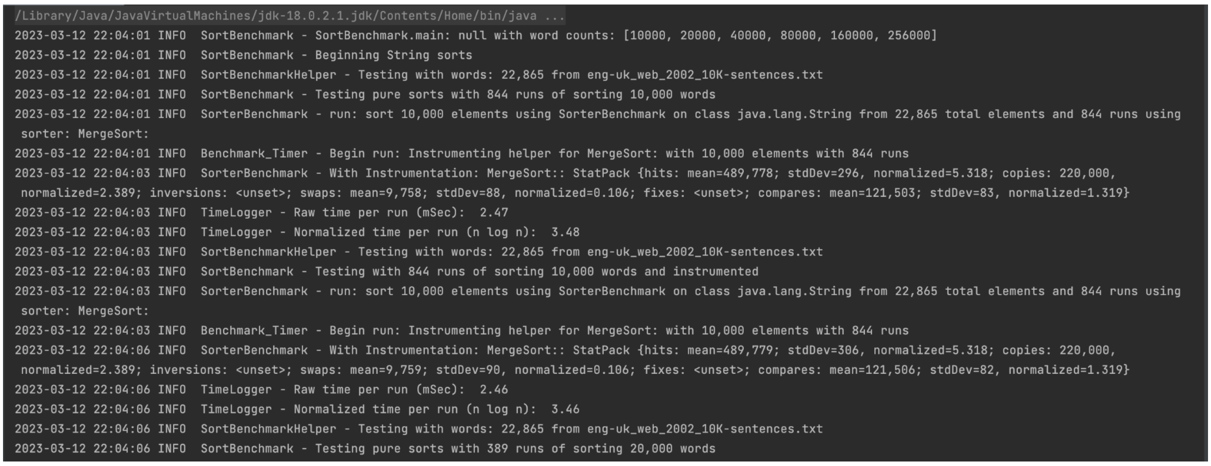
Program Execution Screenshots:



```
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
2023-03-12 22:03:22 INFO  SortBenchmark - SortBenchmark.main: null with word counts: [10000, 20000, 40000, 80000, 160000, 256000]
2023-03-12 22:03:22 INFO  SortBenchmark - Beginning String sorts
2023-03-12 22:03:22 INFO  SortBenchmarkHelper - Testing with words: 22,865 from eng-uk_web_2002_10K-sentences.txt
2023-03-12 22:03:22 INFO  SortBenchmark - Testing pure sorts with 844 runs of sorting 10,000 words
2023-03-12 22:03:22 INFO  SorterBenchmark - run: sort 10,000 elements using SorterBenchmark on class java.lang.String from 22,865 total elements and 844 runs using
 sorter: QuickSort dual pivot
2023-03-12 22:03:22 INFO  Benchmark_Timer - Begin run: Instrumenting helper for QuickSort dual pivot with 10,000 elements with 844 runs
2023-03-12 22:03:24 INFO  SorterBenchmark - With Instrumentation: QuickSort dual pivot: StatPack {hits: mean=423,834; stdDev=20,153, normalized=4.602; copies: 0,
 normalized=0.000; inversions: <unset>; swaps: mean=66,449; stdDev=4,257, normalized=0.721; fixes: <unset>; compares: mean=156,258; stdDev=6,800, normalized=1.697}
2023-03-12 22:03:24 INFO  TimeLogger - Raw time per run (mSec):  2.51
2023-03-12 22:03:24 INFO  TimeLogger - Normalized time per run (n log n):  3.53
2023-03-12 22:03:24 INFO  SortBenchmarkHelper - Testing with words: 22,865 from eng-uk_web_2002_10K-sentences.txt
2023-03-12 22:03:24 INFO  SortBenchmark - Testing with 844 runs of sorting 10,000 words and instrumented
2023-03-12 22:03:24 INFO  SorterBenchmark - run: sort 10,000 elements using SorterBenchmark on class java.lang.String from 22,865 total elements and 844 runs using
 sorter: QuickSort dual pivot
2023-03-12 22:03:24 INFO  Benchmark_Timer - Begin run: Instrumenting helper for QuickSort dual pivot with 10,000 elements with 844 runs
2023-03-12 22:03:27 INFO  SorterBenchmark - With Instrumentation: QuickSort dual pivot: StatPack {hits: mean=423,902; stdDev=18,337, normalized=4.602; copies: 0,
 normalized=0.000; inversions: <unset>; swaps: mean=66,502; stdDev=3,866, normalized=0.722; fixes: <unset>; compares: mean=156,114; stdDev=6,506, normalized=1.695}
2023-03-12 22:03:27 INFO  TimeLogger - Raw time per run (mSec):  2.43
2023-03-12 22:03:27 INFO  TimeLogger - Normalized time per run (n log n):  3.41
2023-03-12 22:03:27 INFO  SortBenchmarkHelper - Testing with words: 22,865 from eng-uk_web_2002_10K-sentences.txt
2023-03-12 22:03:27 INFO  SortBenchmark - Testing pure sorts with 389 runs of sorting 20,000 words
```



```
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
2023-03-12 21:27:56 INFO  SortBenchmark - SortBenchmark.main: null with word counts: [10000, 20000, 40000, 80000, 160000, 256000]
2023-03-12 21:27:56 INFO  SortBenchmark - Beginning String sorts
2023-03-12 21:27:56 INFO  SortBenchmarkHelper - Testing with words: 22,865 from eng-uk_web_2002_10K-sentences.txt
2023-03-12 21:27:56 INFO  SortBenchmark - Testing pure sorts with 844 runs of sorting 10,000 words
2023-03-12 21:27:56 INFO  SorterBenchmark - run: sort 10,000 elements using SorterBenchmark on class java.lang.String from 22,865 total elements and 844 runs using
 sorter: Heapsort
2023-03-12 21:27:56 INFO  Benchmark_Timer - Begin run: Instrumenting helper for Heapsort with 10,000 elements with 844 runs
2023-03-12 21:27:59 INFO  SorterBenchmark - With Instrumentation: Heapsort: StatPack {hits: mean=967,543; stdDev=462, normalized=10.505; copies: 0, normalized=0.000;
  inversions: <unset>; swaps: mean=124,201; stdDev=76, normalized=1.348; fixes: <unset>; compares: mean=235,369; stdDev=91, normalized=2.555}
2023-03-12 21:27:59 INFO  TimeLogger - Raw time per run (mSec):  2.86
2023-03-12 21:27:59 INFO  TimeLogger - Normalized time per run (n log n):  4.02
2023-03-12 21:27:59 INFO  SortBenchmarkHelper - Testing with words: 22,865 from eng-uk_web_2002_10K-sentences.txt
2023-03-12 21:27:59 INFO  SortBenchmark - Testing with 844 runs of sorting 10,000 words and instrumented
2023-03-12 21:27:59 INFO  SorterBenchmark - run: sort 10,000 elements using SorterBenchmark on class java.lang.String from 22,865 total elements and 844 runs using
 sorter: Heapsort
2023-03-12 21:27:59 INFO  Benchmark_Timer - Begin run: Instrumenting helper for Heapsort with 10,000 elements with 844 runs
2023-03-12 21:28:02 INFO  SorterBenchmark - With Instrumentation: Heapsort: StatPack {hits: mean=967,568; stdDev=454, normalized=10.505; copies: 0, normalized=0.000;
  inversions: <unset>; swaps: mean=124,205; stdDev=75, normalized=1.349; fixes: <unset>; compares: mean=235,374; stdDev=89, normalized=2.556}
2023-03-12 21:28:02 INFO  TimeLogger - Raw time per run (mSec):  2.82
2023-03-12 21:28:02 INFO  TimeLogger - Normalized time per run (n log n):  3.97
2023-03-12 21:28:02 INFO  SortBenchmarkHelper - Testing with words: 22,865 from eng-uk_web_2002_10K-sentences.txt
2023-03-12 21:28:02 INFO  SortBenchmark - Testing pure sorts with 389 runs of sorting 20,000 words
2023-03-12 21:28:02 INFO  SorterBenchmark - run: sort 20,000 elements using SorterBenchmark on class java.lang.String from 22,865 total elements and 389 runs using
 sorter: Heapsort
2023-03-12 21:28:02 INFO  Benchmark_Timer - Begin run: Instrumenting helper for Heapsort with 20,000 elements with 389 runs
2023-03-12 21:28:05 INFO  SorterBenchmark - With Instrumentation: Heapsort: StatPack {hits: mean=2,095,154; stdDev=623, normalized=10.578; copies: 0, normalized=0
 .000; inversions: <unset>; swaps: mean=268,412; stdDev=102, normalized=1.355; fixes: <unset>; compares: mean=510,753; stdDev=126, normalized=2.579}
2023-03-12 21:28:05 INFO  TimeLogger - Raw time per run (mSec):  6.26
2023-03-12 21:28:05 INFO  TimeLogger - Normalized time per run (n log n):  4.06
2023-03-12 21:28:05 INFO  SortBenchmarkHelper - Testing with words: 22,865 from eng-uk_web_2002_10K-sentences.txt
```



```
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
2023-03-12 22:04:01 INFO  SortBenchmark - SortBenchmark.main: null with word counts: [10000, 20000, 40000, 80000, 160000, 256000]
2023-03-12 22:04:01 INFO  SortBenchmark - Beginning String sorts
2023-03-12 22:04:01 INFO  SortBenchmarkHelper - Testing with words: 22,865 from eng-uk_web_2002_10K-sentences.txt
2023-03-12 22:04:01 INFO  SortBenchmark - Testing pure sorts with 844 runs of sorting 10,000 words
2023-03-12 22:04:01 INFO  SorterBenchmark - run: sort 10,000 elements using SorterBenchmark on class java.lang.String from 22,865 total elements and 844 runs using
 sorter: MergeSort:
2023-03-12 22:04:01 INFO  Benchmark_Timer - Begin run: Instrumenting helper for MergeSort: with 10,000 elements with 844 runs
2023-03-12 22:04:03 INFO  SorterBenchmark - With Instrumentation: MergeSort:: StatPack {hits: mean=489,778; stdDev=296, normalized=5.318; copies: 220,000,
 normalized=2.389; inversions: <unset>; swaps: mean=9,758; stdDev=88, normalized=0.106; fixes: <unset>; compares: mean=121,503; stdDev=83, normalized=1.319}
2023-03-12 22:04:03 INFO  TimeLogger - Raw time per run (mSec):  2.47
2023-03-12 22:04:03 INFO  TimeLogger - Normalized time per run (n log n):  3.48
2023-03-12 22:04:03 INFO  SortBenchmarkHelper - Testing with words: 22,865 from eng-uk_web_2002_10K-sentences.txt
2023-03-12 22:04:03 INFO  SortBenchmark - Testing with 844 runs of sorting 10,000 words and instrumented
2023-03-12 22:04:03 INFO  SorterBenchmark - run: sort 10,000 elements using SorterBenchmark on class java.lang.String from 22,865 total elements and 844 runs using
 sorter: MergeSort:
2023-03-12 22:04:03 INFO  Benchmark_Timer - Begin run: Instrumenting helper for MergeSort: with 10,000 elements with 844 runs
2023-03-12 22:04:06 INFO  SorterBenchmark - With Instrumentation: MergeSort:: StatPack {hits: mean=489,779; stdDev=306, normalized=5.318; copies: 220,000,
 normalized=2.389; inversions: <unset>; swaps: mean=9,759; stdDev=90, normalized=0.106; fixes: <unset>; compares: mean=121,506; stdDev=82, normalized=1.319}
2023-03-12 22:04:06 INFO  TimeLogger - Raw time per run (mSec):  2.46
2023-03-12 22:04:06 INFO  TimeLogger - Normalized time per run (n log n):  3.46
2023-03-12 22:04:06 INFO  SortBenchmarkHelper - Testing with words: 22,865 from eng-uk_web_2002_10K-sentences.txt
2023-03-12 22:04:06 INFO  SortBenchmark - Testing pure sorts with 389 runs of sorting 20,000 words
```