Program Structures and Algorithms
Spring 2023(SEC – 08)

NAME: Daiming Yang
NUID: 002771605

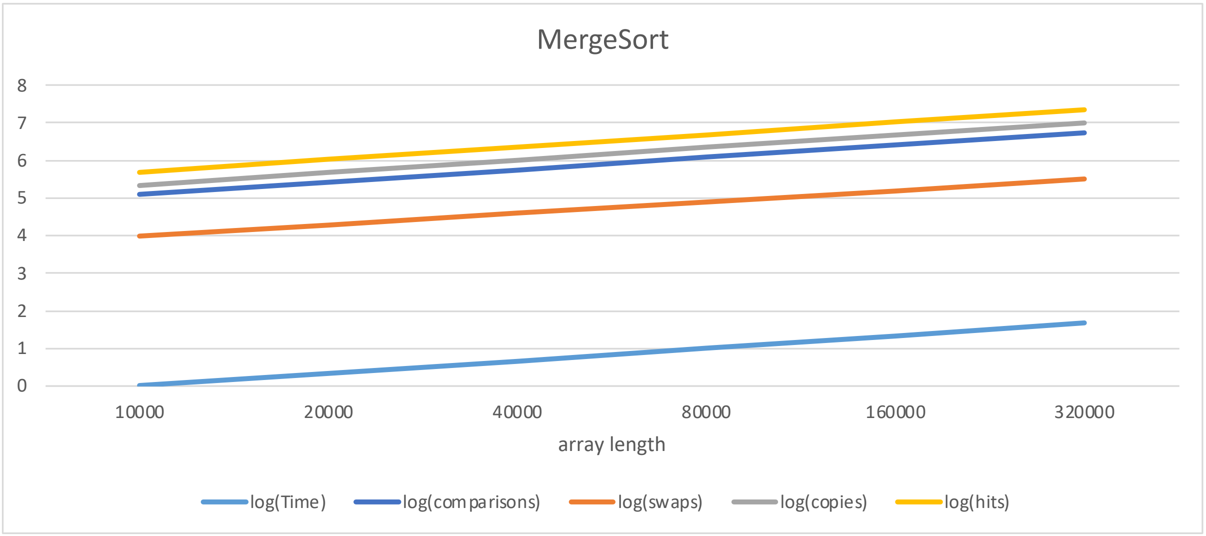
Task: Assignment6 (Hits as time predictor)

Relationship Conclusion:

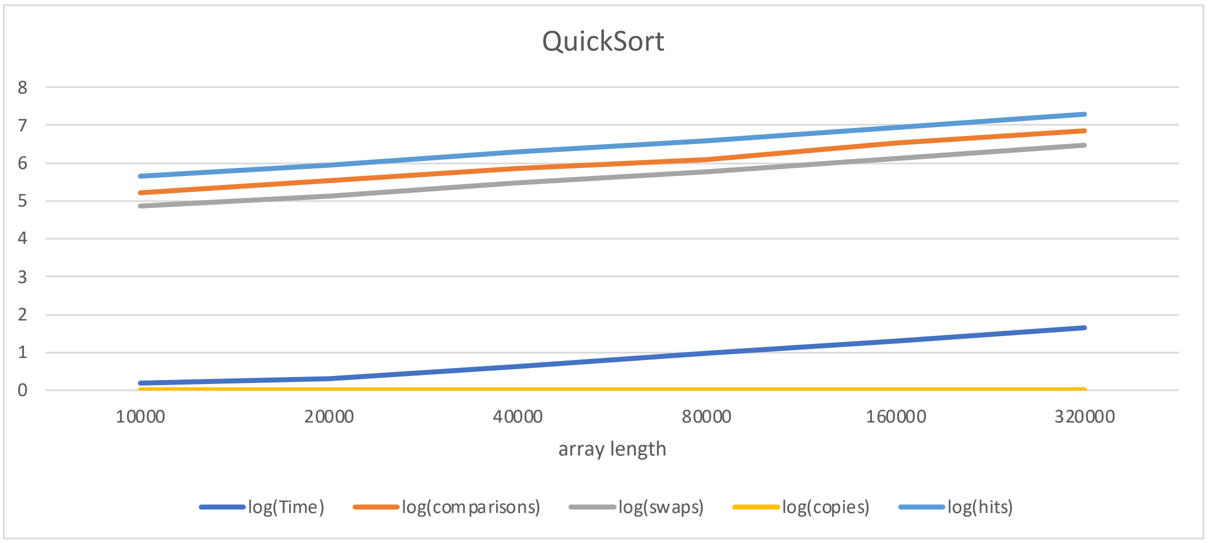
According to the data, swap and copy are more expensive than the comparison and hits. So swaps/copies can be a good predictor.

Evidence to support that conclusion:

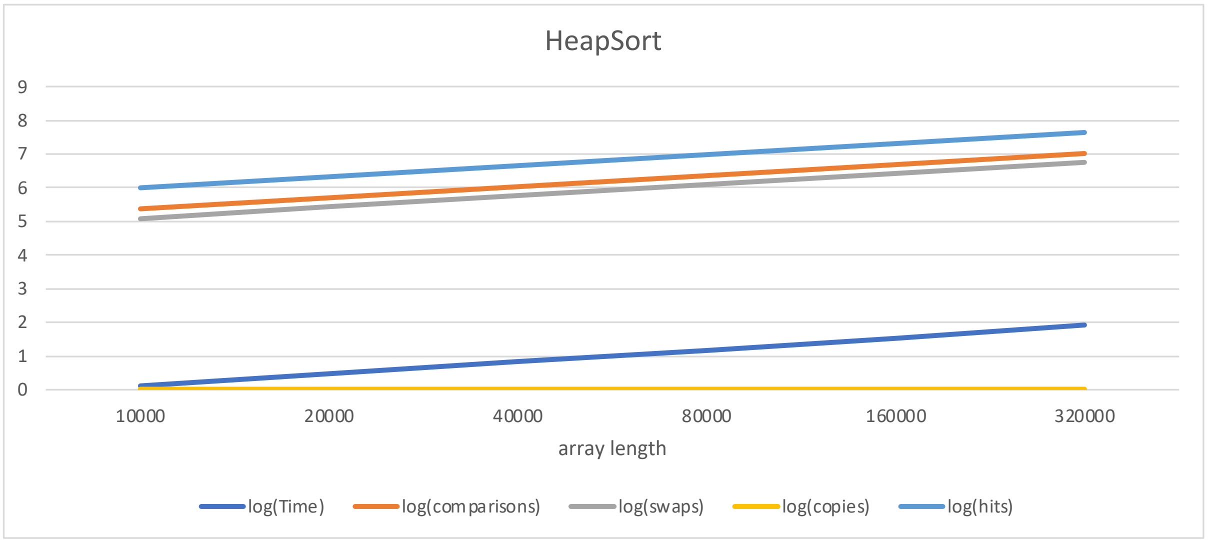
MergeSort	ArrayLength	log(Time)	log(comparisons)	log(swaps)	log(copies)	log(hits)
	10000	0.004321374	5.084517653	3.989738954	5.342422681	5.690013106
	20000	0.320146286	5.419868055	4.28868509	5.681241237	6.025009379
	40000	0.666517981	5.752850805	4.592526496	6.017033339	6.357824621
	80000	0.994317153	6.083509571	4.893191758	6.350248018	6.688298042
	160000	1.331629718	6.412283501	5.193022973	6.681241237	7.02E+00
	320000	1.686010291	6.739432208	5.494718115	7.01E+00	7.34E+00



QuickSort	ArrayLength	log(Time)	log(comparisons)	log(swaps)	log(copies)	log(hits)
	10000	0.201397124	5.201855759	4.856148638	0	5.652263751
	20000	0.307496038	5.527888854	5.133497393	0	5.947788463
	40000	0.639486489	5.857261315	5.477464212	0	6.285982394
	80000	0.972202838	6.083509571	5.78319513	0	6.59816958
	160000	1.302763708	6.52396161	6.12796508	0	6.94E+00
	320000	1.652343055	6.848502066	6.468636764	0	7.28E+00



HeapSort	ArrayLength	log(Time)	log(comparisons)	log(swaps)	log(copies)	log(hits)
	10000	0.117271296	5.371754798	5.093677283	0	5.985442642
	20000	0.475671188	5.708154803	5.428879586	0	6.321228387
	40000	0.820857989	6.041875273	5.758688635	0	6.654105855
	80000	1.164055292	6.373464824	6.091120845	0	6.984968315
	160000	1.51295108	6.70295155	6.419479911	0	7.31E+00
	320000	1.915505362	7.030696041	6.746211607	0	7.64E+00



MergeSort Data:

MergeSort with instrumented:

```
ArrayLength: 10000
2023-03-12 19:18:48 INFO  SorterBenchmark - run: sort 10,000 elements using
SorterBenchmark on class java.lang.Integer from 10,000 total elements and 10 runs using
sorter: mergeSort
2023-03-12 19:18:48 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
mergeSort with 10,000 elements with 10 runs
2023-03-12 19:18:48 INFO  TimeLogger - Raw time per run (mSec):  5.20
2023-03-12 19:18:48 INFO  TimeLogger - Normalized time per run (n log n):  7.32
number of comparisons, swaps, copies, hits: 121483.6, 9766.5, 220000.0, 489793.6
ArrayLength: 20000
```

```
2023-03-12 19:18:48 INFO  SorterBenchmark - run: sort 20,000 elements using
SorterBenchmark on class java.lang.Integer from 20,000 total elements and 10 runs using
sorter: mergeSort
2023-03-12 19:18:48 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
mergeSort with 20,000 elements with 10 runs
2023-03-12 19:18:48 INFO  TimeLogger - Raw time per run (mSec):  3.00
2023-03-12 19:18:48 INFO  TimeLogger - Normalized time per run (n log n):  1.95
number of comparisons, swaps, copies, hits: 262946.9, 19439.5, 480000.0, 1059276.6
ArrayLength: 40000
2023-03-12 19:18:48 INFO  SorterBenchmark - run: sort 40,000 elements using
SorterBenchmark on class java.lang.Integer from 40,000 total elements and 10 runs using
sorter: mergeSort
2023-03-12 19:18:48 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
mergeSort with 40,000 elements with 10 runs
2023-03-12 19:18:48 INFO  TimeLogger - Raw time per run (mSec):  7.10
2023-03-12 19:18:48 INFO  TimeLogger - Normalized time per run (n log n):  2.14
number of comparisons, swaps, copies, hits: 566044.8, 39131.5, 1040000.0, 2279421.4
ArrayLength: 80000
2023-03-12 19:18:48 INFO  SorterBenchmark - run: sort 80,000 elements using
SorterBenchmark on class java.lang.Integer from 80,000 total elements and 10 runs using
sorter: mergeSort
2023-03-12 19:18:48 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
mergeSort with 80,000 elements with 10 runs
2023-03-12 19:18:49 INFO  TimeLogger - Raw time per run (mSec):  14.50
2023-03-12 19:18:49 INFO  TimeLogger - Normalized time per run (n log n):  2.03
number of comparisons, swaps, copies, hits: 1212019.4, 78197.3, 2240000.0, 4878631.8
ArrayLength: 160000
2023-03-12 19:18:49 INFO  SorterBenchmark - run: sort 160,000 elements using
SorterBenchmark on class java.lang.Integer from 160,000 total elements and 10 runs using
sorter: mergeSort
2023-03-12 19:18:49 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
mergeSort with 160,000 elements with 10 runs
2023-03-12 19:18:49 INFO  TimeLogger - Raw time per run (mSec):  31.30
2023-03-12 19:18:49 INFO  TimeLogger - Normalized time per run (n log n):  2.06
number of comparisons, swaps, copies, hits: 2583946.4, 155963.5, 4800000.0,
1.03959282E7
ArrayLength: 320000
2023-03-12 19:18:49 INFO  SorterBenchmark - run: sort 320,000 elements using
SorterBenchmark on class java.lang.Integer from 320,000 total elements and 10 runs using
sorter: mergeSort
2023-03-12 19:18:49 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
mergeSort with 320,000 elements with 10 runs
2023-03-12 19:18:50 INFO  TimeLogger - Raw time per run (mSec):  74.00
2023-03-12 19:18:50 INFO  TimeLogger - Normalized time per run (n log n):  2.29
number of comparisons, swaps, copies, hits: 5488228.8, 312405.1, 1.024E7, 2.20733498E7
```

MergeSort without instrumented:

```
ArrayLength: 10000
```

```
2023-03-12 22:06:05 INFO  SorterBenchmark - run: sort 10,000 elements using
SorterBenchmark on class java.lang.Integer from 10,000 total elements and 100 runs using
sorter: mergeSort
2023-03-12 22:06:05 INFO  Benchmark_Timer - Begin run: Helper for mergeSort with 10000
elements with 100 runs
2023-03-12 22:06:05 INFO  TimeLogger - Raw time per run (mSec):  1.01
2023-03-12 22:06:05 INFO  TimeLogger - Normalized time per run (n log n):  1.42
ArrayLength: 20000
2023-03-12 22:06:05 INFO  SorterBenchmark - run: sort 20,000 elements using
SorterBenchmark on class java.lang.Integer from 20,000 total elements and 100 runs using
sorter: mergeSort
2023-03-12 22:06:05 INFO  Benchmark_Timer - Begin run: Helper for mergeSort with 20000
elements with 100 runs
2023-03-12 22:06:05 INFO  TimeLogger - Raw time per run (mSec):  2.09
2023-03-12 22:06:05 INFO  TimeLogger - Normalized time per run (n log n):  1.36
ArrayLength: 40000
2023-03-12 22:06:05 INFO  SorterBenchmark - run: sort 40,000 elements using
SorterBenchmark on class java.lang.Integer from 40,000 total elements and 100 runs using
sorter: mergeSort
2023-03-12 22:06:05 INFO  Benchmark_Timer - Begin run: Helper for mergeSort with 40000
elements with 100 runs
2023-03-12 22:06:06 INFO  TimeLogger - Raw time per run (mSec):  4.64
2023-03-12 22:06:06 INFO  TimeLogger - Normalized time per run (n log n):  1.40
ArrayLength: 80000
2023-03-12 22:06:06 INFO  SorterBenchmark - run: sort 80,000 elements using
SorterBenchmark on class java.lang.Integer from 80,000 total elements and 100 runs using
sorter: mergeSort
2023-03-12 22:06:06 INFO  Benchmark_Timer - Begin run: Helper for mergeSort with 80000
elements with 100 runs
2023-03-12 22:06:07 INFO  TimeLogger - Raw time per run (mSec):  9.87
2023-03-12 22:06:07 INFO  TimeLogger - Normalized time per run (n log n):  1.39
ArrayLength: 160000
2023-03-12 22:06:07 INFO  SorterBenchmark - run: sort 160,000 elements using
SorterBenchmark on class java.lang.Integer from 160,000 total elements and 100 runs using
sorter: mergeSort
2023-03-12 22:06:07 INFO  Benchmark_Timer - Begin run: Helper for mergeSort with
160000 elements with 100 runs
2023-03-12 22:06:09 INFO  TimeLogger - Raw time per run (mSec):  21.46
2023-03-12 22:06:09 INFO  TimeLogger - Normalized time per run (n log n):  1.41
ArrayLength: 320000
2023-03-12 22:06:09 INFO  SorterBenchmark - run: sort 320,000 elements using
SorterBenchmark on class java.lang.Integer from 320,000 total elements and 100 runs using
sorter: mergeSort
2023-03-12 22:06:09 INFO  Benchmark_Timer - Begin run: Helper for mergeSort with
320000 elements with 100 runs
2023-03-12 22:06:16 INFO  TimeLogger - Raw time per run (mSec):  48.53
2023-03-12 22:06:16 INFO  TimeLogger - Normalized time per run (n log n):  1.50
```

QuickSort Data:

QuickSort with instrumented:

```
ArrayLength: 10000
2023-03-12 21:16:46 INFO  SorterBenchmark - run: sort 10,000 elements using
SorterBenchmark on class java.lang.Integer from 10,000 total elements and 1 runs using
sorter: quickSort
2023-03-12 21:16:46 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
quickSort with 10,000 elements with 1 runs
2023-03-12 21:16:47 INFO  TimeLogger - Raw time per run (mSec):  308.00
2023-03-12 21:16:47 INFO  TimeLogger - Normalized time per run (n log n):  433.35
quickSort: StatPack {hits: 449,018, normalized=4.875; copies: 0, normalized=0.000;
inversions: <unset>; swaps: 71,804, normalized=0.780; fixes: 36,045,790,
normalized=391.362; compares: 159,168, normalized=1.728}
number of comparisons, swaps, copies, hits: 159168.0, 71804.0, 0.0, 449018.0
ArrayLength: 20000
2023-03-12 21:16:47 INFO  SorterBenchmark - run: sort 20,000 elements using
SorterBenchmark on class java.lang.Integer from 20,000 total elements and 1 runs using
sorter: quickSort
2023-03-12 21:16:47 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
quickSort with 20,000 elements with 1 runs
2023-03-12 21:16:49 INFO  TimeLogger - Raw time per run (mSec):  692.00
2023-03-12 21:16:49 INFO  TimeLogger - Normalized time per run (n log n):  448.93
quickSort: StatPack {hits: 886,724, normalized=4.477; copies: 0, normalized=0.000;
inversions: <unset>; swaps: 135,987, normalized=0.687; fixes: 105,220,997,
normalized=531.232; compares: 337,201, normalized=1.702}
number of comparisons, swaps, copies, hits: 337201.0, 135987.0, 0.0, 886724.0
ArrayLength: 40000
2023-03-12 21:16:49 INFO  SorterBenchmark - run: sort 40,000 elements using
SorterBenchmark on class java.lang.Integer from 40,000 total elements and 1 runs using
sorter: quickSort
2023-03-12 21:16:49 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
quickSort with 40,000 elements with 1 runs
2023-03-12 21:17:04 INFO  TimeLogger - Raw time per run (mSec):  4151.00
2023-03-12 21:17:04 INFO  TimeLogger - Normalized time per run (n log n):  1249.23
quickSort: StatPack {hits: 1,931,890, normalized=4.558; copies: 0, normalized=0.000;
inversions: <unset>; swaps: 300,237, normalized=0.708; fixes: 448,707,697,
normalized=1058.609; compares: 719,882, normalized=1.698}
number of comparisons, swaps, copies, hits: 719882.0, 300237.0, 0.0, 1931890.0
ArrayLength: 80000
2023-03-12 21:17:04 INFO  SorterBenchmark - run: sort 80,000 elements using
SorterBenchmark on class java.lang.Integer from 80,000 total elements and 1 runs using
sorter: quickSort
2023-03-12 21:17:04 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
quickSort with 80,000 elements with 1 runs
2023-03-12 21:17:58 INFO  TimeLogger - Raw time per run (mSec):  13120.00
2023-03-12 21:17:58 INFO  TimeLogger - Normalized time per run (n log n):  1841.23
```

```
quickSort: StatPack {hits: 3,964,328, normalized=4.389; copies: 0, normalized=0.000;
inversions: <unset>; swaps: 607,009, normalized=0.672; fixes: 1,681,772,247,
normalized=1862.051; compares: 1,513,587, normalized=1.676}
number of comparisons, swaps, copies, hits: 1513587.0, 607009.0, 0.0, 3964328.0
ArrayLength: 160000
2023-03-12 21:17:58 INFO  SorterBenchmark - run: sort 160,000 elements using
SorterBenchmark on class java.lang.Integer from 160,000 total elements and 1 runs using
sorter: quickSort
2023-03-12 21:17:58 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
quickSort with 160,000 elements with 1 runs
2023-03-12 21:20:39 INFO  TimeLogger - Raw time per run (mSec):  42910.00
2023-03-12 21:20:39 INFO  TimeLogger - Normalized time per run (n log n):  2820.93
quickSort: StatPack {hits: 8,755,791, normalized=4.567; copies: 0, normalized=0.000;
inversions: <unset>; swaps: 1,342,657, normalized=0.700; fixes: -1,943,421,698,
normalized=-1013.641; compares: 3,341,655, normalized=1.743}
number of comparisons, swaps, copies, hits: 3341655.0, 1342657.0, 0.0, 8755791.0
ArrayLength: 320000
2023-03-12 21:36:03 INFO  SorterBenchmark - run: sort 320,000 elements using
SorterBenchmark on class java.lang.Integer from 320,000 total elements and 1 runs using
sorter: quickSort
2023-03-12 21:36:03 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
quickSort with 320,000 elements with 1 runs
2023-03-12 21:46:04 INFO  TimeLogger - Raw time per run (mSec):  149315.00
2023-03-12 21:46:04 INFO  TimeLogger - Normalized time per run (n log n):  4616.68
quickSort: StatPack {hits: 18,911,587, normalized=4.662; copies: 0, normalized=0.000;
inversions: <unset>; swaps: 2,941,960, normalized=0.725; fixes: 2,113,628,825,
normalized=521.067; compares: 7,055,082, normalized=1.739}
number of comparisons, swaps, copies, hits: 7055082.0, 2941960.0, 0.0, 1.8911587E7
```

==

QuickSort without instrumented:

```
ArrayLength: 10000
2023-03-12 22:06:16 INFO  SorterBenchmark - run: sort 10,000 elements using
SorterBenchmark on class java.lang.Integer from 10,000 total elements and 100 runs using
sorter: quickSort
2023-03-12 22:06:16 INFO  Benchmark_Timer - Begin run: Helper for quickSort with 10000
elements with 100 runs
2023-03-12 22:06:16 INFO  TimeLogger - Raw time per run (mSec):  1.59
2023-03-12 22:06:16 INFO  TimeLogger - Normalized time per run (n log n):  2.24

ArrayLength: 20000
2023-03-12 22:06:16 INFO  SorterBenchmark - run: sort 20,000 elements using
SorterBenchmark on class java.lang.Integer from 20,000 total elements and 100 runs using
sorter: quickSort
2023-03-12 22:06:16 INFO  Benchmark_Timer - Begin run: Helper for quickSort with 20000
elements with 100 runs
2023-03-12 22:06:16 INFO  TimeLogger - Raw time per run (mSec):  2.03
2023-03-12 22:06:16 INFO  TimeLogger - Normalized time per run (n log n):  1.32
```

```
ArrayLength: 40000
2023-03-12 22:06:16 INFO  SorterBenchmark - run: sort 40,000 elements using
SorterBenchmark on class java.lang.Integer from 40,000 total elements and 100 runs using
sorter: quickSort
2023-03-12 22:06:16 INFO  Benchmark_Timer - Begin run: Helper for quickSort with 40000
elements with 100 runs
2023-03-12 22:06:16 INFO  TimeLogger - Raw time per run (mSec):  4.36
2023-03-12 22:06:16 INFO  TimeLogger - Normalized time per run (n log n):  1.31

ArrayLength: 80000
2023-03-12 22:06:16 INFO  SorterBenchmark - run: sort 80,000 elements using
SorterBenchmark on class java.lang.Integer from 80,000 total elements and 100 runs using
sorter: quickSort
2023-03-12 22:06:16 INFO  Benchmark_Timer - Begin run: Helper for quickSort with 80000
elements with 100 runs
2023-03-12 22:06:18 INFO  TimeLogger - Raw time per run (mSec):  9.38
2023-03-12 22:06:18 INFO  TimeLogger - Normalized time per run (n log n):  1.32

ArrayLength: 160000
2023-03-12 22:06:18 INFO  SorterBenchmark - run: sort 160,000 elements using
SorterBenchmark on class java.lang.Integer from 160,000 total elements and 100 runs using
sorter: quickSort
2023-03-12 22:06:18 INFO  Benchmark_Timer - Begin run: Helper for quickSort with 160000
elements with 100 runs
2023-03-12 22:06:20 INFO  TimeLogger - Raw time per run (mSec):  20.08
2023-03-12 22:06:20 INFO  TimeLogger - Normalized time per run (n log n):  1.32

ArrayLength: 320000
2023-03-12 22:06:20 INFO  SorterBenchmark - run: sort 320,000 elements using
SorterBenchmark on class java.lang.Integer from 320,000 total elements and 100 runs using
sorter: quickSort
2023-03-12 22:06:20 INFO  Benchmark_Timer - Begin run: Helper for quickSort with 320000
elements with 100 runs
2023-03-12 22:06:26 INFO  TimeLogger - Raw time per run (mSec):  44.91
2023-03-12 22:06:26 INFO  TimeLogger - Normalized time per run (n log n):  1.39
```

HeapSort Data:

HeapSort with instrumented:

```
ArrayLength: 10000
2023-03-12 21:20:39 INFO  SorterBenchmark - run: sort 10,000 elements using
SorterBenchmark on class java.lang.Integer from 10,000 total elements and 1 runs using
sorter: heapSort
```

```
2023-03-12 21:20:39 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
heapSort with 10,000 elements with 1 runs
2023-03-12 21:20:40 INFO  TimeLogger - Raw time per run (mSec):  263.00
2023-03-12 21:20:40 INFO  TimeLogger - Normalized time per run (n log n):  370.03
number of comparisons, swaps, copies, hits: 235372.0, 124073.0, 0.0, 967036.0
ArrayLength: 20000
2023-03-12 21:20:40 INFO  SorterBenchmark - run: sort 20,000 elements using
SorterBenchmark on class java.lang.Integer from 20,000 total elements and 1 runs using
sorter: heapSort
2023-03-12 21:20:40 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
heapSort with 20,000 elements with 1 runs
2023-03-12 21:20:43 INFO  TimeLogger - Raw time per run (mSec):  1109.00
2023-03-12 21:20:43 INFO  TimeLogger - Normalized time per run (n log n):  719.45
number of comparisons, swaps, copies, hits: 510687.0, 268460.0, 0.0, 2095214.0
ArrayLength: 40000
2023-03-12 21:20:43 INFO  SorterBenchmark - run: sort 40,000 elements using
SorterBenchmark on class java.lang.Integer from 40,000 total elements and 1 runs using
sorter: heapSort
2023-03-12 21:20:43 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
heapSort with 40,000 elements with 1 runs
2023-03-12 21:20:57 INFO  TimeLogger - Raw time per run (mSec):  4819.00
2023-03-12 21:20:57 INFO  TimeLogger - Normalized time per run (n log n):  1450.26
number of comparisons, swaps, copies, hits: 1101223.0, 576705.0, 0.0, 4509266.0
ArrayLength: 80000
2023-03-12 21:20:57 INFO  SorterBenchmark - run: sort 80,000 elements using
SorterBenchmark on class java.lang.Integer from 80,000 total elements and 1 runs using
sorter: heapSort
2023-03-12 21:20:57 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
heapSort with 80,000 elements with 1 runs
2023-03-12 21:21:56 INFO  TimeLogger - Raw time per run (mSec):  19611.00
2023-03-12 21:21:56 INFO  TimeLogger - Normalized time per run (n log n):  2752.16
number of comparisons, swaps, copies, hits: 2363006.0, 1233448.0, 0.0, 9659804.0
ArrayLength: 160000
2023-03-12 21:21:56 INFO  SorterBenchmark - run: sort 160,000 elements using
SorterBenchmark on class java.lang.Integer from 160,000 total elements and 1 runs using
sorter: heapSort
2023-03-12 21:21:56 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
heapSort with 160,000 elements with 1 runs
2023-03-12 21:25:55 INFO  TimeLogger - Raw time per run (mSec):  79502.00
2023-03-12 21:25:55 INFO  TimeLogger - Normalized time per run (n log n):  5226.52
number of comparisons, swaps, copies, hits: 5046050.0, 2627120.0, 0.0, 2.060058E7
ArrayLength: 320000
2023-03-12 21:46:04 INFO  SorterBenchmark - run: sort 320,000 elements using
SorterBenchmark on class java.lang.Integer from 320,000 total elements and 1 runs using
sorter: heapSort
2023-03-12 21:46:04 INFO  Benchmark_Timer - Begin run: Instrumenting helper for
heapSort with 320,000 elements with 1 runs
2023-03-12 21:58:20 INFO  TimeLogger - Raw time per run (mSec):  248752.00
```

```
2023-03-12 21:58:20 INFO  TimeLogger - Normalized time per run (n log n):  7691.18
number of comparisons, swaps, copies, hits: 1.073238E7, 5574573.0, 0.0, 4.3763052E7
```

===

HeapSort without instrumented:

```
ArrayLength: 10000
2023-03-12 22:06:26 INFO  SorterBenchmark - run: sort 10,000 elements using
SorterBenchmark on class java.lang.Integer from 10,000 total elements and 100 runs using
sorter: heapSort
2023-03-12 22:06:26 INFO  Benchmark_Timer - Begin run: Helper for heapSort with 10000
elements with 100 runs
2023-03-12 22:06:26 INFO  TimeLogger - Raw time per run (mSec):  1.31
2023-03-12 22:06:26 INFO  TimeLogger - Normalized time per run (n log n):  1.84
ArrayLength: 20000
2023-03-12 22:06:26 INFO  SorterBenchmark - run: sort 20,000 elements using
SorterBenchmark on class java.lang.Integer from 20,000 total elements and 100 runs using
sorter: heapSort
2023-03-12 22:06:26 INFO  Benchmark_Timer - Begin run: Helper for heapSort with 20000
elements with 100 runs
2023-03-12 22:06:26 INFO  TimeLogger - Raw time per run (mSec):  2.99
2023-03-12 22:06:26 INFO  TimeLogger - Normalized time per run (n log n):  1.94
ArrayLength: 40000
2023-03-12 22:06:26 INFO  SorterBenchmark - run: sort 40,000 elements using
SorterBenchmark on class java.lang.Integer from 40,000 total elements and 100 runs using
sorter: heapSort
2023-03-12 22:06:26 INFO  Benchmark_Timer - Begin run: Helper for heapSort with 40000
elements with 100 runs
2023-03-12 22:06:27 INFO  TimeLogger - Raw time per run (mSec):  6.62
2023-03-12 22:06:27 INFO  TimeLogger - Normalized time per run (n log n):  1.99
ArrayLength: 80000
2023-03-12 22:06:27 INFO  SorterBenchmark - run: sort 80,000 elements using
SorterBenchmark on class java.lang.Integer from 80,000 total elements and 100 runs using
sorter: heapSort
2023-03-12 22:06:27 INFO  Benchmark_Timer - Begin run: Helper for heapSort with 80000
elements with 100 runs
2023-03-12 22:06:29 INFO  TimeLogger - Raw time per run (mSec):  14.59
2023-03-12 22:06:29 INFO  TimeLogger - Normalized time per run (n log n):  2.05
ArrayLength: 160000
2023-03-12 22:06:29 INFO  SorterBenchmark - run: sort 160,000 elements using
SorterBenchmark on class java.lang.Integer from 160,000 total elements and 100 runs using
sorter: heapSort
2023-03-12 22:06:29 INFO  Benchmark_Timer - Begin run: Helper for heapSort with 160000
elements with 100 runs
2023-03-12 22:06:33 INFO  TimeLogger - Raw time per run (mSec):  32.58
2023-03-12 22:06:33 INFO  TimeLogger - Normalized time per run (n log n):  2.14
ArrayLength: 320000
```

```
2023-03-12 22:06:33 INFO  SorterBenchmark - run: sort 320,000 elements using
SorterBenchmark on class java.lang.Integer from 320,000 total elements and 100 runs using
sorter: heapSort
2023-03-12 22:06:33 INFO  Benchmark_Timer - Begin run: Helper for heapSort with 320000
elements with 100 runs
2023-03-12 22:06:43 INFO  TimeLogger - Raw time per run (mSec):  82.32
2023-03-12 22:06:43 INFO  TimeLogger - Normalized time per run (n log n):  2.55
```