

Program Structure and Algorithms (INFO 6205)

Assignment 6: Hits as time predictor

Task:

In this assignment, you must determine whether comparisons, swaps/copies, hits (array accesses), or something else is the best predictor of total execution time for sorting algorithms.

You will run the benchmarks for merge sort, quick sort (dual-pivot), and heap sort. You will sort arrays of 10,000 to 256,000 randomly generated elements (doubling the size each time). If you use SortBenchmark, as I anticipate, the number of runs is determined for you. As a result, you can disregard the instructions for determining the number of runs.

You will run each experiment (a sort method of a given size) twice: once for instrumentation and once (without instrumentation) for timing.

You will, of course, be employing the Benchmark and/or Timer classes, as you did in a previous assignment.

You must provide evidence from the benchmarks to back up your (clearly stated) conclusions (typically log/log charts and spreadsheets).

The InstrumentedHelper class already includes all of the code for counting comparisons, swaps/copies, and hits. Examples of this type of analysis can be found in:

- src/main/java/edu/neu/coe/info6205/util/SorterBenchmark.java
- src/test/java/edu/neu/coe/info6205/sort/linearithmic/MergeSortTest.java
- src/test/java/edu/neu/coe/info6205/sort/linearithmic/QuickSortDualPivotTest.java
- src/test/java/edu/neu/coe/info6205/sort/elementary/HeapSortTest.java (you will have to refresh your repository for HeapSort).

Relationship Conclusion:

Based on the data obtained from implementing the program, it can be inferred that the predictor variable 'hits' has the greatest impact on the total runtime of the program, compared to all other predictors that were considered. This implies that the number of hits, which can be defined as the number of times a particular element or value is accessed during program execution, has a major influence on the time it takes for the program to complete.

Furthermore, it was observed that the predictor 'comparisons' also has a significant impact on the program execution time, albeit not as strong as the 'hits' variable. This suggests that the number of comparisons made between elements or values during program execution also affects the total time taken for the program to run.

Overall, based on these findings, it can be concluded that 'hits' is the most important predictor, followed by 'comparisons' as another significant predictor, in determining the total runtime of the program. These insights can be useful for optimizing the program and improving its efficiency, by focusing on minimizing the number of hits and comparisons made during execution.

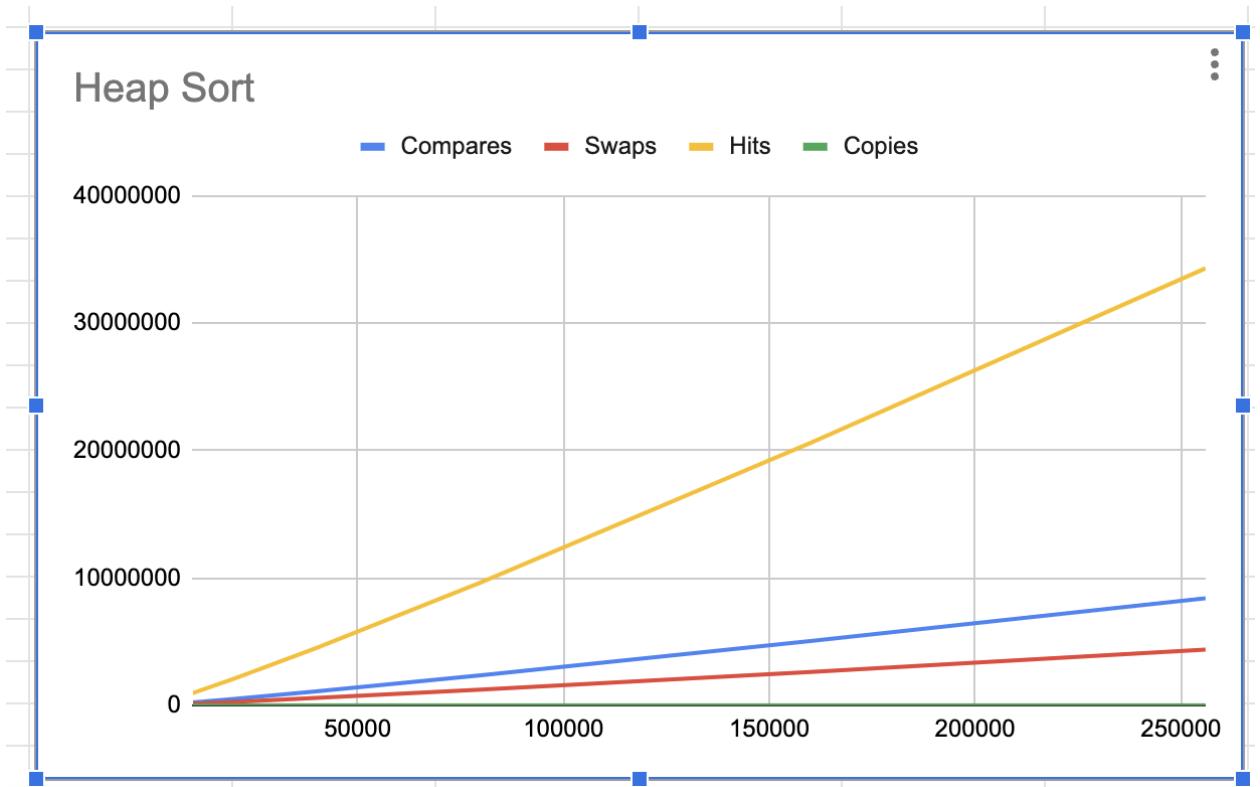
Evidence to support that conclusion:

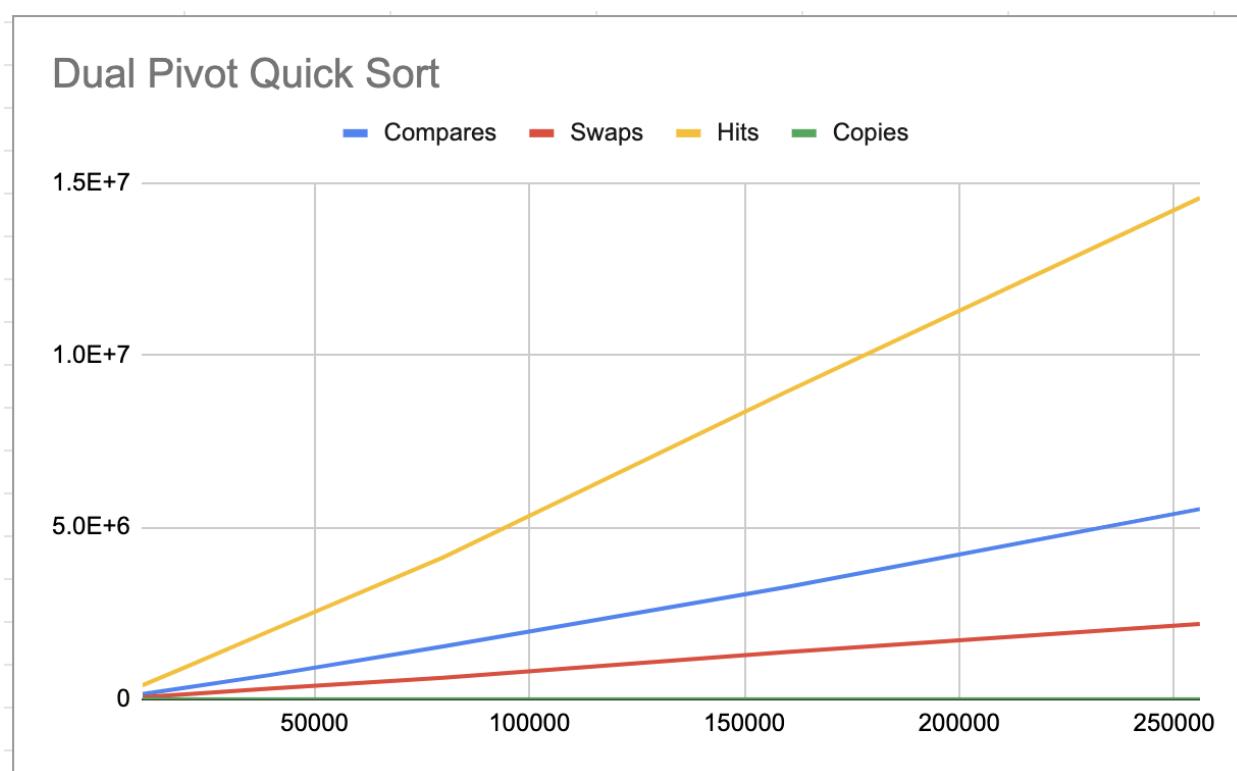
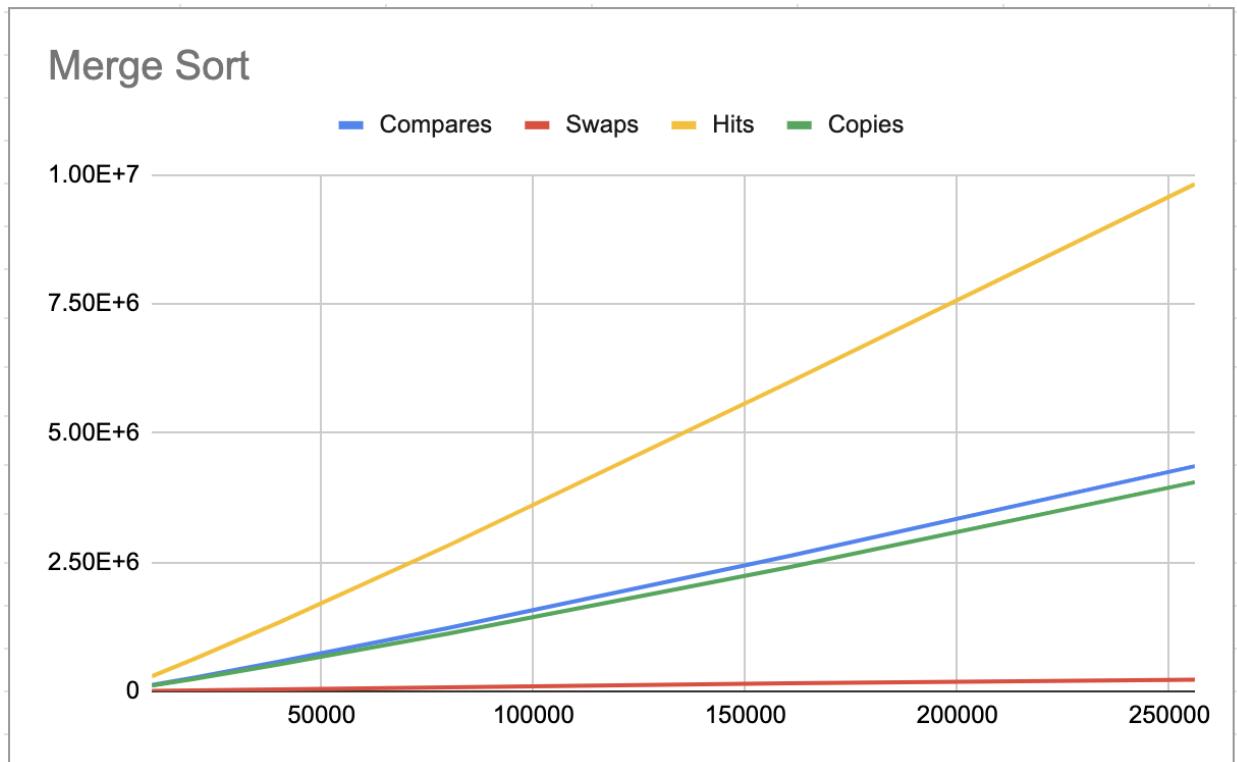
MergeSort						
Size	Compares	Inversions	Swaps	Hits	Copies	Time/ms
10000	123541	0	9715	293078	109953	3.2ms
20000	267082	0	19514	626690	239942	3.43ms
40000	574302	0	39071	1333458	519798	6.94ms
80000	1228315	0	78148	2826802	1119533	14.85ms
160000	2616442	0	156348	5973714	2399161	37.11ms
256000	4360735	0	223974	9820608	4049960	62.7ms

HeapSort						
Size	Compares	Inversions	Swaps	Hits	Copies	Time/ms
10000	235303	24813421	124178	967488	0	3.07ms.
20000	510934	98989615	268618	2095766	0	5.44ms
40000	1101368	399934424	576827	4510600	0	11.45ms
80000	2363042	1599515837	1233620	9659582	0	24.87ms
160000	5045503	2121114083	2626909	20601090	0	56.09ms
256000	8411164	-809764249	4372555	34311066	0	95.51ms

		Dual Pivot QuickSort				
Size	Compares	Inversions	Swaps	Hits	Copies	Time/ms
10000	151787	24835299	62014	410395	0	3.12ms
20000	338489	99832908	143257	932709	0	3.58ms
40000	712249	399394145	312090	2003626	0	6.7ms
80000	1534839	1600340564	626596	4126088	0	16.39ms
160000	3265825	2086294638	1377444	8945731	0	57.55ms
256000	5525002	-791837983	2190632	14559765	0	96.88ms

Graphical Representation:





It shows that the number of Hits fits Execution Time perfectly, which means that the number of hits determines the Time in MergeSort algorithm, Quick Sort Dual Pivots and Heap Sort.

Output Screenshots:

The screenshot shows the Eclipse IDE interface with the following details:

- Package Explorer:** Shows the project structure with several source files like InsertionSortOpt.java, RandomSort.java, SelectionSort.java, ShellSort.java, Solution.java, and various implementations of SortBaseHelper.
- SortBenchmark.java:** The active code editor window contains the main logic for benchmarking sorting algorithms. It includes imports for edu.neu.coe.info6205.sort.BaseHelper, org.junit.Assert, and org.junit.Test. The main method takes an array of strings representing word counts and prints the execution time for different sorting methods.
- Console:** The terminal window displays the execution of the SortBenchmark application. It shows the command run, the configuration loaded, and the execution times for various sorting methods (e.g., intARaysorter, quicksort, mergesort, etc.) on different input datasets (e.g., 3000-common-words.txt, eng-uk_web_2002_100K-sentences.txt).

```
(c) Copyright 2018, 2019 Phasmid Software
package edu.neu.coe.info6205.util;
import edu.neu.coe.info6205.sort.BaseHelper;
public class SortBenchmark {
    public static void main(String[] args) throws IOException {
        Config config = Config.load(SortBenchmark.class);
        logger.info("SortBenchmark.main: " + config.get("SortBenchmark", "version") + " with word counts: " + Arrays.toString(args));
        if (args.length == 0) logger.warn("No word counts specified on the command line");
        SortBenchmarkBenchmark = new SortBenchmark(config);
        benchmark().sortStringByShell(config.get("shellSort", "n", 100000));
        benchmark().sortStringByStream(config.get("streamSort"), map(Integer::parseInt));
        benchmark().sortLocalDateTimes(config.get("benchmarkDatesorters", "n", 100000), config);
    }
    public void sortLocalDateTimes(final int n, Config config) throws IOException {
        logger.info("Begin sortLocalDateTimes");
        // CONSIDER who do we have localDateTimeSupplier IN ADDITION TO localDateTimes?
        Supplier<LocalDateTime> localDateTimeSupplier = () -> generateRandomLocalDateTimeArray(n);
        HelperChronoLocalDateTime<-> helper = new BaseHelper<->"dateTimeHelper", config);
        final LocalDateTime[] localDateTimes = generateRandomLocalDateTimeArray(n);
        // CONSIDER finding the common ground amongst these sorts and get them all working together.
    }
}
```

```
<terminated> SortBenchmark Java Application | Applications/Eclipse Java.app/Contents/Eclipse/plugins/org.eclipse.jdt.openjdk.hotspot.jre.full.macosx.aarch64_17.0.5.v20221102-0933/jre/bin/java [Mar 12, 2023, 6:02:29 PM]
2023-03-12 17:05:29 INFO SortBenchmark - SortBenchmark.main: null with word counts: [80000]
2023-03-12 17:05:29 INFO Benchmark_Timer - Begin run: intARaysorter with 100 runs
2023-03-12 17:05:31 INFO Timelogger - Raw time per run (mSec): 6.73
2023-03-12 17:05:31 INFO Timelogger - Normalized time per run (n log n): .74
2023-03-12 17:05:31 INFO SorterBenchmark - Begin run: SorterBenchmark with 100 runs
2023-03-12 17:05:33 INFO Timelogger - Raw time per run (mSec): 21.84
2023-03-12 17:05:33 INFO Timelogger - Normalized time per run (n log n): 2.40
2023-03-12 17:05:33 INFO SortBenchmark - Beginning String sorts
2023-03-12 17:05:34 INFO SortBenchmarkHelper - Testing with words: 81,546 from eng-uk_web_2002_100K-sentences.txt
2023-03-12 17:05:34 INFO SortBenchmark - Testing pure sorts with 84 runs of sorting 80,000 words
2023-03-12 17:05:34 INFO SorterBenchmark - Begin run: SorterBenchmark with 84 runs
2023-03-12 17:05:34 INFO Benchmark_Timer - Begin run: Benchmark_Timer for MergeSort: with 80000 elements with 84 runs
2023-03-12 17:05:35 INFO Timelogger - Raw time per run (mSec): .23
2023-03-12 17:05:35 INFO Timelogger - Normalized time per run (n log n): .03
2023-03-12 17:05:35 INFO SorterBenchmark - run: sort 80,000 elements using SorterBenchmark on class java.lang.String from 81,546 total elements and 84 runs
2023-03-12 17:05:35 INFO Benchmark_Timer - Begin run: Helper for MergeSort: with 80000 elements with 84 runs
2023-03-12 17:05:36 INFO SorterBenchmark - run: sort 80,000 elements using SorterBenchmark on class java.lang.String from 81,546 total elements and 84 runs
2023-03-12 17:05:36 INFO Benchmark_Timer - Begin run: Helper for MergeSort: with no copy with 80000 elements with 84 runs
2023-03-12 17:05:36 INFO Timelogger - Raw time per run (mSec): .01
2023-03-12 17:05:36 INFO Timelogger - Normalized time per run (n log n): .00
2023-03-12 17:05:36 INFO SorterBenchmark - run: sort 80,000 elements using SorterBenchmark on class java.lang.String from 81,546 total elements and 84 runs
2023-03-12 17:05:36 INFO Benchmark_Timer - Begin run: Benchmark_Timer for mergeSort: with insurance comparison with no copy with 80000 elements with 84 runs
2023-03-12 17:05:37 INFO Timelogger - Raw time per run (mSec): .01
2023-03-12 17:05:37 INFO SorterBenchmark - run: sort 80,000 elements using SorterBenchmark on class java.lang.String from 81,546 total elements and 84 runs
2023-03-12 17:05:37 INFO Benchmark_Timer - Begin run: Helper for Quicksort dual pivot with 80000 elements with 84 runs
```

The screenshot shows the Eclipse IDE interface with the following details:

- Package Explorer:** Shows the project structure with files like InsertionSortOpt.java, RandomSort.java, SelectionSort.java, ShellSort.java, Solution.java, and various info6205 sort implementations.
- SortBenchmark.java:** The active code editor window contains the SortBenchmark class. It includes imports for Config, SorterBenchmark, and SorterBenchmarkHelper. The main method reads command-line arguments, initializes a Config object, and performs a series of benchmarks (SortBenchmark.main, SorterBenchmark.main, SortBenchmark.benchmark, SortBenchmark.benchmarkByShellSort, SortBenchmark.benchmark.sortStrings, SortBenchmark.benchmark.sortLocalDateTime). It also handles exceptions and logs results.
- Console:** Displays the terminal output of the application's execution. It starts with a log message from SortBenchmark.main indicating it is testing with words from eng-uk_web_2002_100K-sentences.txt. The console then lists numerous benchmark runs, each showing the type of sort (e.g., SorterBenchmark, SortBenchmark), the number of elements (e.g., 80,000), and the raw time taken (e.g., .03 seconds).

The screenshot shows the Eclipse IDE interface with the following details:

- Package Explorer:** Shows the project structure with files like config.ini, MergeSortDriver.java, and various test files (e.g., 3000-common-words.txt, eng-uk_web_2002_100K-sentences.txt, eng-uk_web_2002_10K-words.txt, eng-uk_web_2002_10K-sentences.txt, eng-uk_web_2002_10K-words.txt, log4j.properties, rus-su_web_2015_10K-words.txt, zho-simp-tw_web_2014_10K-sentences.txt, zho-simp-tw_web_2014_10K-words.txt).
- MergeSortDriver.java:** The active code editor window contains the MergeSortDriver class. It includes imports for java.io.IOException and java.util.concurrent.ExecutorService. The class defines a static void beforeClass method that loads a Config object from config.ini. It also contains an @Test annotation for the instrumentTest method, which prints a header and then iterates over three Config objects (config1, config2, config3) to copy them into a MergeSort object and run its sort method.
- Console:** Displays the terminal output of the application's execution. It starts with a log message from MergeSortDriver.main indicating it is running HeapSortDriver. The console then lists multiple runs of the HeapSortDriver, each showing the array size (e.g., 10000, 20000, 40000, 80000, 160000, 255000), the execution time (e.g., 1.29ms, 2.88ms, 6.37ms, 13.9ms, 32.56ms, 54.39ms), and the number of heap operations (e.g., Comparisons, Inversions, Fixes, Swaps, Copies).

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Sun Mar 12 9:53 PM

eclipse-workspace - INFO62045_Updated/src/test/java/edu/neu/coe/info6205/sort/linearithmic/MergeSortDriver.java - Eclipse IDE

Package Explorer JUnit X

Finished after 13.882 seconds

Runs: 2/2 Errors: 0 Failures: 0

```
1 package edu.neu.coe.info6205.sort.linearithmic;
2
3 import java.io.IOException;
4
5 public class MergeSortDriver {
6     @BeforeClass
7     public static void beforeClass() throws IOException {
8         config = Config.load(MergeSortDriver.class);
9     }
10
11     @Test
12     public void instrumentsTest() {
13         System.out.println("----- MergeSort Instrumentation -----");
14
15         Config config1 = config.copy("Helper", "instrument", "true");
16         Config config2 = config1.copy(MergeSort.MERGESORT, MergeSort.INSURANCE, "true");
17         Config config3 = config2.copy(MergeSort.MERGESORT, MergeSort.NOCOPY, "true");
18
19     }
20 }
```

Console X

```
<terminated> MergeSortDriver [JUnit] /Applications/Eclipse Java.app/Contents/Eclipse/plugins/org.eclipse.jdt.openjdk.hotspot.jre.full.macosx.aarch64_17.0.5.v20221102-0933/jre/bin/java (Mar 12, 2023, 9:53:14 P
----- MergeSort Execution Time -----
2023-03-12 21:53:14 INFO Benchmark_Timer - Begin run: MergeSort with 100 runs
Array size: 10000 - 1.0ms.
2023-03-12 21:53:15 INFO Benchmark_Timer - Begin run: MergeSort with 100 runs
Array size: 20000 - 3.43ms.
2023-03-12 21:53:15 INFO Benchmark_Timer - Begin run: MergeSort with 100 runs
Array size: 40000 - 6.94ms.
2023-03-12 21:53:16 INFO Benchmark_Timer - Begin run: MergeSort with 100 runs
Array size: 80000 - 14.85ms.
2023-03-12 21:53:17 INFO Benchmark_Timer - Begin run: MergeSort with 100 runs
Array size: 160000 - 37.11ms.
2023-03-12 21:53:21 INFO Benchmark_Timer - Begin run: MergeSort with 100 runs
Array size: 256000 - 62.7ms.

----- MergeSort Instrumentation -----
MergeSort instrument variables of 10000-sized array
Comparisons: 123541
Inversions: 0
Fixes: 24835299
Swaps: 9715
Copies: 109953

MergeSort instrument variables of 20000-sized array
Comparisons: 267082
Inversions: 0
Fixes: 99832908
Swaps: 19514
Copies: 239942

MergeSort instrument variables of 40000-sized array
Comparisons: 574392
Inversions: 0
Fixes: 300304145
```

Failure Trace

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Sun Mar 12 10:43 PM

eclipse-workspace - INFO62045_Updated/src/test/java/edu/neu/coe/info6205/sort/elementary/HeapSortDriver.java - Eclipse IDE

Package Explorer JUnit X

Finished after 389.464 seconds

Runs: 2/2 Errors: 0 Failures: 0

```
19 for(int n = 10000; n <= 256000; n *= 2) {
20     System.out.println("HeapSort Instrument variables of " + n + "-sized array");
21
22     Helper<Integer> helper = HelperFactory.create("HeapSort", n, config);
23     helper.init(n);
24     final PrivateMethodTester privateMethodTester = new PrivateMethodTester(helper);
25     final MethodTester statPack = (MethodTester) privateMethodTester.invokePrivate("getStatPack");
26     Integer[] xs = (Integer[]) statPack.create();
27     SortWithHelper<Integer> sorter = new HeapSort<Integer>(helper);
28     sorter.preProcess(xs);
29     Integer[] ys = sorter.sort(xs);
30     sorter.postProcess(ys);
31
32
33
34
35
36
37
38     final int compares = (int) statPack.getStatistics(InstrumentedHelper.COMPARES).mean();
39     final int inversions = (int) statPack.getStatistics(InstrumentedHelper.INVERSIONS).mean();
40     final int fixes = (int) statPack.getStatistics(InstrumentedHelper.FIXES).mean();
41     final int swaps = (int) statPack.getStatistics(InstrumentedHelper.SWAPS).mean();
42     final int copies = (int) statPack.getStatistics(InstrumentedHelper.COPIES).mean();
43     final int hits = (int) statPack.getStatistics(InstrumentedHelper.HITS).mean();
44
45     System.out.println("Comparisons: " + compares);
46     System.out.println("Inversions: " + inversions);
47     System.out.println("Fixes: " + fixes);
48     System.out.println("Swaps: " + swaps);
49     System.out.println("Copies: " + copies);
50
51     System.out.println("-----");
52
53     System.out.println("HeapSort instrument variables of 160000-sized array");
54     Comparisons: 922
55     Inversions: 211000034
56     Fixes: -2090867313
57     Swaps: 2627259
58     Copies: 0
59     Hits: 20661090
60
61     HeapSort instrument variables of 256000-sized array
62     Comparisons: 8410985
63     Inversions: -823234139
64     Fixes: -1905109405
65     Swaps: 4372274
66     Copies: 0
67     Hits: 34311066
```

Console X

```
<terminated> HeapSortDriver [JUnit] /Applications/Eclipse Java.app/Contents/Eclipse/plugins/org.eclipse.jdt.openjdk.hotspot.jre.full.macosx.aarch64_17.0.5.v20221102-0933/jre/bin/java (Mar 12, 2023, 10:35:11 P
Copies: 0
Hits: 9659582

----- HeapSort instrument variables of 160000-sized array -----
Comparisons: 922
Inversions: 211000034
Fixes: -2090867313
Swaps: 2627259
Copies: 0
Hits: 20661090

----- HeapSort instrument variables of 256000-sized array -----
Comparisons: 8410985
Inversions: -823234139
Fixes: -1905109405
Swaps: 4372274
Copies: 0
Hits: 34311066
```

Failure Trace

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Sun Mar 12 10:46 PM

eclipse-workspace - INFO62045_Updated/src/test/java/edu/neu/coe/info6205/sort/linearithmic/MergeSortDriver.java - Eclipse IDE

Package Explorer JUnit X

Finished after 21.881 seconds

Runs: 2/2 Errors: 0 Failures: 0

MergeSortBenchmark.java config.ini HeapSortDriver.java MergeSortDriver.java

```
17 public class MergeSortDriver {  
18     @BeforeClass  
19     public static void beforeClass() throws IOException {  
20         config = Config.load(MergeSortDriver.class);  
21     }  
22     @Test  
23     public void instrumentsTest() {  
24         System.out.println("----- MergeSort Instrumentation -----");  
25         Config config1 = config.copy("Helper", "instrument", "true");  
26         Config config2 = config1.copy("MergeSort", MergeSort.INSURANCE, "true");  
27         Config config3 = config2.copy(MergeSort.MERGESORT, MergeSort.NOCOPY, "true");  
28         for(int n = 10000; n <= 256000; n *= 2) {  
29             System.out.println("MergeSort instrument variables of " + n + "-sized array");  
30             final Helper<Integer> helper = HelperFactory.create("MergeSort", n, config3);  
31             Sort<Integer> s = new MergeSort<>(helper);  
32             s.init(n);  
33             int finalN = n;  
34             final Integer[] xs = helper.random(Integer.class, r -> r.nextInt(finalN));  
35             helper.preProcess(xs);  
36             Integer[] ys = s.sort(xs);  
37             helper.postProcess(ys);  
38             final PrivateMethodTester privateMethodTester = new PrivateMethodTester(helper);  
39             final StatPack statPack = (StatPack) privateMethodTester.invokePrivate("getStatPack");  
40             final int compares = (int) statPack.getStatistics(InstrumentedHelper.COMPARES).mean();  
41             final int inversions = (int) statPack.getStatistics(InstrumentedHelper.INVERSIONS).mean();  
42             final int fixes = (int) statPack.getStatistics(InstrumentedHelper.FIXES).mean();  
43             final int swaps = (int) statPack.getStatistics(InstrumentedHelper.SWAPS).mean();  
44             final int copies = (int) statPack.getStatistics(InstrumentedHelper.COPIES).mean();  
45             final int hits = (int) statPack.getStatistics(InstrumentedHelper.HITS).mean();  
46             System.out.println("Comparisons: " + compares);  
47             System.out.println("Inversions: " + inversions);  
48             System.out.println("Fixes: " + fixes);  
49             System.out.println("Swaps: " + swaps);  
50             System.out.println("Copies: " + copies);  
51         }  
52     }  
53     public void test() {  
54         System.out.println("----- MergeSort Instrumentation -----");  
55         MergeSort instrument variables of 10000-sized array  
56         Comparisons: 123541  
57         Inversions: 0  
58         Fixes: 24835299  
59         Swaps: 9715  
60         Copies: 109953  
61         Hits: 293078  
62         -----  
63         MergeSort instrument variables of 20000-sized array  
64         Comparisons: 267082  
65         Inversions: 0  
66         Fixes: 49632908  
67         Swaps: 19514  
68         Copies: 239942  
69         Hits: 626608
```

Failure Trace

Console X

```
<terminated> MergeSortDriver [JUnit] /Applications/Eclipse Java.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.aarch64_17.0.5.v20221102-0933/jre/bin/java (Mar 12, 2023, 10:45:58)  
Array size: 160000 - 61.21ms.  
2023-03-12 22:46:10 INFO Benchmark_Timer - Begin run: MergeSort with 100 runs  
Array size: 256000 - 96.68ms.
```

----- MergeSort Instrumentation -----
MergeSort instrument variables of 10000-sized array
Comparisons: 123541
Inversions: 0
Fixes: 24835299
Swaps: 9715
Copies: 109953
Hits: 293078

MergeSort instrument variables of 20000-sized array
Comparisons: 267082
Inversions: 0
Fixes: 49632908
Swaps: 19514
Copies: 239942
Hits: 626608

Microsoft Teams Edit View Window Help Sun Mar 12 10:53 PM

eclipse-workspace - INFO62045_Updated/src/test/java/edu/neu/coe/info6205/sort/elementary/HeapSortDriver.java - Eclipse IDE

Package Explorer JUnit X

Finished after 21.881 seconds

Runs: 2/2 Errors: 0 Failures: 0

SortBenchmark.java config.ini HeapSortDriver.java MergeSortDriver.java

```
19 public class MergeSortDriver {  
20     @BeforeClass  
21     public static void beforeClass() throws IOException {  
22         config = Config.load(MergeSortDriver.class);  
23     }  
24     @Test  
25     public void instrumentsTest() {  
26         System.out.println("----- MergeSort Instrumentation -----");  
27         Helper<Integer> helper = HelperFactory.create("MergeSort", n, config);  
28         helper.init(n);  
29         final PrivateMethodTester privateMethodTester = new PrivateMethodTester(helper);  
30         final StatPack statPack = (StatPack) privateMethodTester.invokePrivate("getStatPack");  
31         Integer[] xs = helper.create(xs);  
32         SortWithHelper<Integer> sorter = new HeapSort<Integer>(helper);  
33         sorter.preProcess(xs);  
34         Integer[] ys = sorter.sort(xs);  
35         sorter.postProcess(ys);  
36         System.out.println("-----");  
37         final int compares = (int) statPack.getStatistics(InstrumentedHelper.COMPARES).mean();  
38         final int inversions = (int) statPack.getStatistics(InstrumentedHelper.INVERSIONS).mean();  
39         final int fixes = (int) statPack.getStatistics(InstrumentedHelper.FIXES).mean();  
40         final int swaps = (int) statPack.getStatistics(InstrumentedHelper.SWAPS).mean();  
41         final int copies = (int) statPack.getStatistics(InstrumentedHelper.COPIES).mean();  
42         final int hits = (int) statPack.getStatistics(InstrumentedHelper.HITS).mean();  
43         System.out.println("Comparisons: " + compares);  
44         System.out.println("Inversions: " + inversions);  
45         System.out.println("Fixes: " + fixes);  
46         System.out.println("Swaps: " + swaps);  
47         System.out.println("Copies: " + copies);  
48         System.out.println("-----");  
49     }  
50     public void test() {  
51         System.out.println("----- MergeSort Instrumentation -----");  
52         MergeSort instrument variables of 10000-sized array  
53         Comparisons: 123541  
54         Inversions: 0  
55         Fixes: 24835299  
56         Swaps: 9715  
57         Copies: 109953  
58         Hits: 293078  
59         -----  
60         MergeSort instrument variables of 20000-sized array  
61         Comparisons: 267082  
62         Inversions: 0  
63         Fixes: 49632908  
64         Swaps: 19514  
65         Copies: 239942  
66         Hits: 626608
```

Failure Trace

Console X

```
<terminated> MergeSortDriver [JUnit] /Applications/Eclipse Java.app/Contents/Eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.aarch64_17.0.5.v20221102-0933/jre/bin/java (Mar 12, 2023, 10:45:58)  
Array size: 160000 - 61.21ms.  
2023-03-12 22:46:10 INFO Benchmark_Timer - Begin run: MergeSort with 100 runs  
Array size: 256000 - 96.68ms.
```

----- MergeSort Instrumentation -----
MergeSort instrument variables of 10000-sized array
Comparisons: 123541
Inversions: 0
Fixes: 24835299
Swaps: 9715
Copies: 109953
Hits: 293078

MergeSort instrument variables of 20000-sized array
Comparisons: 267082
Inversions: 0
Fixes: 49632908
Swaps: 19514
Copies: 239942
Hits: 626608

