

PSA Assignment 6

002791446

Code:

https://github.com/rohitpanicker06/INFO6205/commits/Assignment_6_002791446

Conclusion:

We have executed two tests with instrumentation on and off, and by comparing both of them we can conclude that the hits, copies, swaps and compares can be the predictors. Also since the graph of **hits** is more linear we can consider hits to be the more accurate predictor.

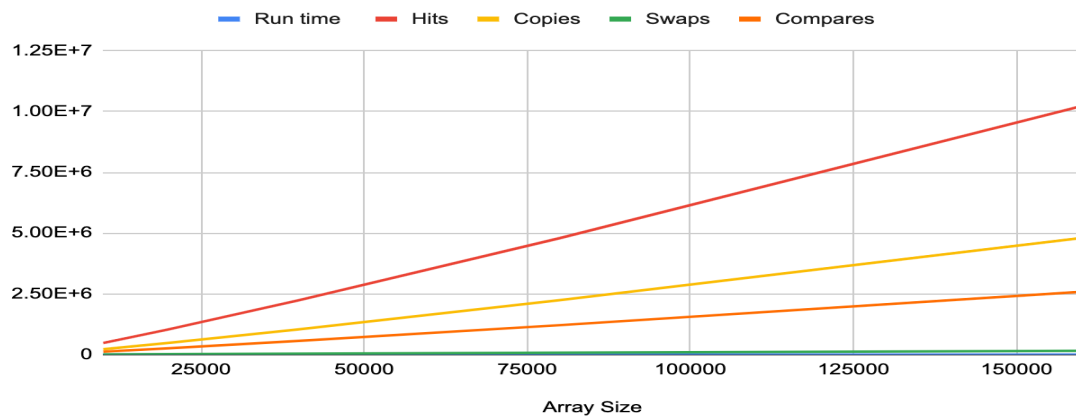
Graphical Representation:

	Merge Sort				
Array Size	Run time	Hits	Copies	Swaps	Compares
10000	2.1	478,943	220,000	9,736	121,524
20000	2.2	1,037,834	480,000	19,458	262,973
40000	5.1	2,236,530	1,040,000	39,133	566,097
80000	10.9	4,792,219	2,240,000	78,055	1,212,135
160000	23.7	10,224,385	4,800,000	156,096	2,584,017

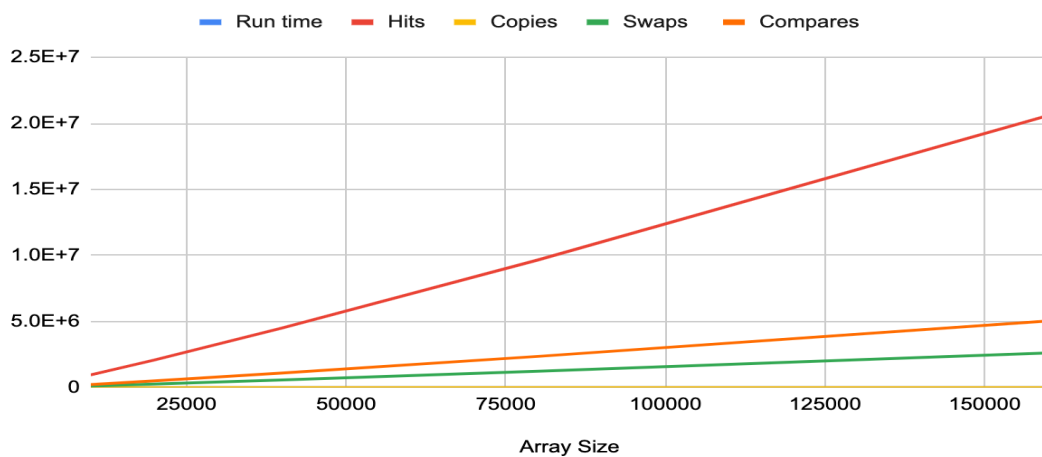
	Heap Sort				
Array Size	Run time	Hits	Copies	Swaps	Compares
10000	1.7	967,466	0	124,187	235,360
20000	3.1	2,095,557	0	268,476	510,826
40000	6.1	4,510,571	0	576,879	1,101,528
80000	12.9	9,660,134	0	1,233,536	2,362,994
160000	27.9	20,600,636	0	2,627,173	5,045,973

	Dual Pivot Quick Sort				
Array Size	Run time	Hits	Copies	Swaps	Compares
10000	2	413,682	0	65,788	154,895
20000	9.3	884,560	0	138,008	341,237
40000	4.5	1,938,229	0	306,929	727,857
80000	9.4	4,145,994	0	652,089	1,572,553
160000	23.3	9,019,253	0	1,411,525	3,442,560

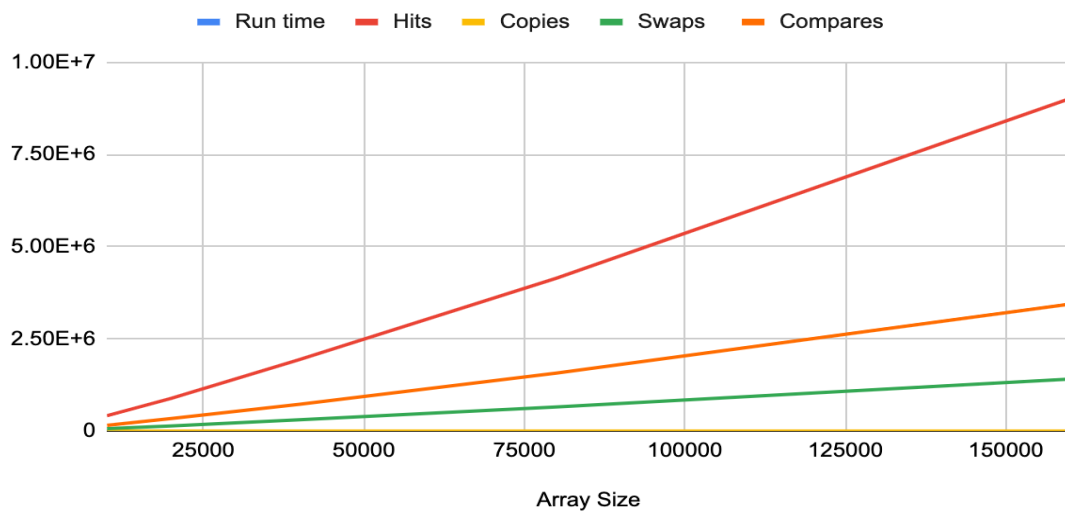
Merge Sort



Heap Sort



Dual Pivot Quick Sort



TEST OUTPUT:

```
/opt/homebrew/Cellar/openjdk/19/libexec/openjdk.jdk/Contents/Home/bin/java
-javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=49366:/Applications/IntelliJ
IDEA CE.app/Contents/bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8
-Dsun.stderr.encoding=UTF-8 -classpath /Users/rohitpanicker/Desktop/NEU/SEM
2/PSA/INFO6205/target/classes:/Users/rohitpanicker/.m2/repository/com/phasmidsoftware/args_2
.13/1.0.3/args_2.13-1.0.3.jar:/Users/rohitpanicker/.m2/repository/org/scala-lang/scala-library/2.13.
7/scala-library-2.13.7.jar:/Users/rohitpanicker/.m2/repository/org/scala-lang/modules/scala-parser-
combinators_2.13/1.1.2/scala-parser-combinators_2.13-1.1.2.jar:/Users/rohitpanicker/.m2/reposi
tory/org/apache/logging/log4j/log4j-api/2.19.0/log4j-api-2.19.0.jar:/Users/rohitpanicker/.m2/reposi
tory/log4j/log4j/1.2.17/log4j-1.2.17.jar:/Users/rohitpanicker/.m2/repository/com/google/guava/guava/
31.1-jre/guava-31.1-jre.jar:/Users/rohitpanicker/.m2/repository/com/google/guava/failureaccess/1.
0.1/failureaccess-1.0.1.jar:/Users/rohitpanicker/.m2/repository/com/google/guava/listenablefuture/
9999.0-empty-to-avoid-conflict-with-guava/listenablefuture-9999.0-empty-to-avoid-conflict-with-gu
ava.jar:/Users/rohitpanicker/.m2/repository/com/google/code/findbugs/jsr305/3.0.2/jsr305-3.0.2.jar
:/Users/rohitpanicker/.m2/repository/org/checkerframework/checker-qual/3.12.0/checker-qual-3.12
.0.jar:/Users/rohitpanicker/.m2/repository/com/google/errorprone/error_prone_annotations/2.11.0/
error_prone_annotations-2.11.0.jar:/Users/rohitpanicker/.m2/repository/com/google/j2objc/j2objc-a
nnotations/1.3/j2objc-annotations-1.3.jar:/Users/rohitpanicker/.m2/repository/org/ini4j/ini4j/0.5.4/ini
4j-0.5.4.jar edu.neu.coe.info6205.util.SortBenchmark
2023-03-12 23:34:30 INFO SortBenchmark - SortBenchmark.main: 1.0.0 (sortbenchmark) with
min: 10000 max: 256000 strategy: doubling
2023-03-12 23:34:30 INFO SorterBenchmark - run: sort 10,000 elements using SorterBenchmark
on class java.lang.Integer from 10,000 total elements and 10 runs using sorter: MergeSortBasic
2023-03-12 23:34:30 INFO Benchmark_Timer - Begin run: Instrumenting helper for
MergeSortBasic with 10,000 elements with 10 runs
2023-03-12 23:34:30 INFO TimeLogger - Raw time per run (mSec): 2.30
2023-03-12 23:34:30 INFO TimeLogger - Normalized time per run (n log n): 3.24
2023-03-12 23:34:30 INFO SortBenchmark - MergeSortBasic: StatPack {hits: mean=479,034;
stdDev=393, normalized=5.201; copies: 220,000, normalized=2.389; inversions: <unset>; swaps:
mean=9,758; stdDev=98, normalized=0.106; fixes: mean=25,041,334; stdDev=215,662,
normalized=271.883; compares: mean=121,471; stdDev=97, normalized=1.319}
2023-03-12 23:34:30 INFO SorterBenchmark - run: sort 10,000 elements using SorterBenchmark
on class java.lang.Integer from 10,000 total elements and 10 runs using sorter: HeapSort
2023-03-12 23:34:30 INFO Benchmark_Timer - Begin run: Instrumenting helper for HeapSort
with 10,000 elements with 10 runs
2023-03-12 23:34:32 INFO TimeLogger - Raw time per run (mSec): 181.60
2023-03-12 23:34:32 INFO TimeLogger - Normalized time per run (n log n): 255.51
2023-03-12 23:34:32 INFO SortBenchmark - HeapSort: StatPack {hits: mean=967,466;
stdDev=601, normalized=10.504; copies: 0, normalized=0.000; inversions: <unset>; swaps:
mean=124,187; stdDev=101, normalized=1.348; fixes: mean=75,427,451; stdDev=270,868,
normalized=818.943; compares: mean=235,360; stdDev=103, normalized=2.555}
```

2023-03-12 23:34:32 INFO SorterBenchmark - run: sort 10,000 elements using SorterBenchmark on class java.lang.Integer from 10,000 total elements and 10 runs using sorter: QuickSort_DualPivot

2023-03-12 23:34:32 INFO Benchmark_Timer - Begin run: Instrumenting helper for QuickSort_DualPivot with 10,000 elements with 10 runs

2023-03-12 23:34:34 INFO TimeLogger - Raw time per run (mSec): 160.80

2023-03-12 23:34:34 INFO TimeLogger - Normalized time per run (n log n): 226.24

2023-03-12 23:34:34 INFO SortBenchmark - QuickSort_DualPivot: StatPack {hits: mean=413,682; stdDev=20,177, normalized=4.491; copies: 0, normalized=0.000; inversions: <unset>; swaps: mean=65,788; stdDev=3,545, normalized=0.714; fixes: mean=29,140,003; stdDev=4,022,480, normalized=316.384; compares: mean=154,895; stdDev=7,492, normalized=1.682}

2023-03-12 23:34:34 INFO SorterBenchmark - run: sort 20,000 elements using SorterBenchmark on class java.lang.Integer from 20,000 total elements and 10 runs using sorter: MergeSortBasic

2023-03-12 23:34:34 INFO Benchmark_Timer - Begin run: Instrumenting helper for MergeSortBasic with 20,000 elements with 10 runs

2023-03-12 23:34:34 INFO TimeLogger - Raw time per run (mSec): 2.40

2023-03-12 23:34:34 INFO TimeLogger - Normalized time per run (n log n): 1.56

2023-03-12 23:34:34 INFO SortBenchmark - MergeSortBasic: StatPack {hits: mean=1,038,064; stdDev=619, normalized=5.241; copies: 480,000, normalized=2.423; inversions: <unset>; swaps: mean=19,516; stdDev=155, normalized=0.099; fixes: mean=99,847,938; stdDev=267,111, normalized=504.105; compares: mean=263,041; stdDev=112, normalized=1.328}

2023-03-12 23:34:34 INFO SorterBenchmark - run: sort 20,000 elements using SorterBenchmark on class java.lang.Integer from 20,000 total elements and 10 runs using sorter: HeapSort

2023-03-12 23:34:34 INFO Benchmark_Timer - Begin run: Instrumenting helper for HeapSort with 20,000 elements with 10 runs

2023-03-12 23:34:43 INFO TimeLogger - Raw time per run (mSec): 738.90

2023-03-12 23:34:43 INFO TimeLogger - Normalized time per run (n log n): 479.35

2023-03-12 23:34:43 INFO SortBenchmark - HeapSort: StatPack {hits: mean=2,095,557; stdDev=599, normalized=10.580; copies: 0, normalized=0.000; inversions: <unset>; swaps: mean=268,476; stdDev=96, normalized=1.355; fixes: mean=302,756,322; stdDev=446,633, normalized=1528.534; compares: mean=510,826; stdDev=121, normalized=2.579}

2023-03-12 23:34:43 INFO SorterBenchmark - run: sort 20,000 elements using SorterBenchmark on class java.lang.Integer from 20,000 total elements and 10 runs using sorter: QuickSort_DualPivot

2023-03-12 23:34:43 INFO Benchmark_Timer - Begin run: Instrumenting helper for QuickSort_DualPivot with 20,000 elements with 10 runs

2023-03-12 23:34:49 INFO TimeLogger - Raw time per run (mSec): 490.10

2023-03-12 23:34:49 INFO TimeLogger - Normalized time per run (n log n): 317.95

2023-03-12 23:34:49 INFO SortBenchmark - QuickSort_DualPivot: StatPack {hits: mean=884,560; stdDev=44,783, normalized=4.466; copies: 0, normalized=0.000; inversions: <unset>; swaps: mean=138,008; stdDev=7,561, normalized=0.697; fixes: mean=113,064,623;

stdDev=18,720,895, normalized=570.832; compares: mean=341,237; stdDev=20,237, normalized=1.723}

2023-03-12 23:34:49 INFO SorterBenchmark - run: sort 40,000 elements using SorterBenchmark on class java.lang.Integer from 40,000 total elements and 10 runs using sorter: MergeSortBasic

2023-03-12 23:34:49 INFO Benchmark_Timer - Begin run: Instrumenting helper for MergeSortBasic with 40,000 elements with 10 runs

2023-03-12 23:34:49 INFO TimeLogger - Raw time per run (mSec): 5.10

2023-03-12 23:34:49 INFO TimeLogger - Normalized time per run (n log n): 1.53

2023-03-12 23:34:49 INFO SortBenchmark - MergeSortBasic: StatPack {hits: mean=2,235,978; stdDev=722, normalized=5.275; copies: 1,040,000, normalized=2.454; inversions: <unset>; swaps: mean=38,995; stdDev=181, normalized=0.092; fixes: mean=399,315,560; stdDev=1,372,031, normalized=942.081; compares: mean=565,986; stdDev=122, normalized=1.335}

2023-03-12 23:34:49 INFO SorterBenchmark - run: sort 40,000 elements using SorterBenchmark on class java.lang.Integer from 40,000 total elements and 10 runs using sorter: HeapSort

2023-03-12 23:34:49 INFO Benchmark_Timer - Begin run: Instrumenting helper for HeapSort with 40,000 elements with 10 runs

2023-03-12 23:35:27 INFO TimeLogger - Raw time per run (mSec): 3167.40

2023-03-12 23:35:27 INFO TimeLogger - Normalized time per run (n log n): 953.22

2023-03-12 23:35:27 INFO SortBenchmark - HeapSort: StatPack {hits: mean=4,510,571; stdDev=543, normalized=10.642; copies: 0, normalized=0.000; inversions: <unset>; swaps: mean=576,879; stdDev=96, normalized=1.361; fixes: mean=1,211,286,241; stdDev=1,042,700, normalized=2857.714; compares: mean=1,101,528; stdDev=103, normalized=2.599}

2023-03-12 23:35:27 INFO SorterBenchmark - run: sort 40,000 elements using SorterBenchmark on class java.lang.Integer from 40,000 total elements and 10 runs using sorter:

QuickSort_DualPivot

2023-03-12 23:35:27 INFO Benchmark_Timer - Begin run: Instrumenting helper for QuickSort_DualPivot with 40,000 elements with 10 runs

2023-03-12 23:35:57 INFO TimeLogger - Raw time per run (mSec): 2549.90

2023-03-12 23:35:57 INFO TimeLogger - Normalized time per run (n log n): 767.38

2023-03-12 23:35:57 INFO SortBenchmark - QuickSort_DualPivot: StatPack {hits: mean=1,938,229; stdDev=83,179, normalized=4.573; copies: 0, normalized=0.000; inversions: <unset>; swaps: mean=306,929; stdDev=16,012, normalized=0.724; fixes: mean=462,538,961; stdDev=63,269,190, normalized=1091.240; compares: mean=727,857; stdDev=27,142, normalized=1.717}

2023-03-12 23:35:57 INFO SorterBenchmark - run: sort 80,000 elements using SorterBenchmark on class java.lang.Integer from 80,000 total elements and 10 runs using sorter: MergeSortBasic

2023-03-12 23:35:57 INFO Benchmark_Timer - Begin run: Instrumenting helper for MergeSortBasic with 80,000 elements with 10 runs

2023-03-12 23:35:57 INFO TimeLogger - Raw time per run (mSec): 11.20

2023-03-12 23:35:57 INFO TimeLogger - Normalized time per run (n log n): 1.57

2023-03-12 23:35:57 INFO SortBenchmark - MergeSortBasic: StatPack {hits: mean=4,792,538; stdDev=1,002, normalized=5.306; copies: 2,240,000, normalized=2.480; inversions: <unset>;

swaps: mean=78,135; stdDev=250, normalized=0.087; fixes: mean=1,601,533,912; stdDev=4,892,570, normalized=1773.212; compares: mean=1,212,009; stdDev=213, normalized=1.342}

2023-03-12 23:35:57 INFO SorterBenchmark - run: sort 80,000 elements using SorterBenchmark on class java.lang.Integer from 80,000 total elements and 10 runs using sorter: HeapSort

2023-03-12 23:35:57 INFO Benchmark_Timer - Begin run: Instrumenting helper for HeapSort with 80,000 elements with 10 runs

2023-03-12 23:38:32 INFO TimeLogger - Raw time per run (mSec): 12758.30

2023-03-12 23:38:32 INFO TimeLogger - Normalized time per run (n log n): 1790.47

2023-03-12 23:38:32 INFO SortBenchmark - HeapSort: StatPack {hits: mean=9,660,134; stdDev=797, normalized=10.696; copies: 0, normalized=0.000; inversions: <unset>; swaps: mean=1,233,536; stdDev=122, normalized=1.366; fixes: mean=546,214,362; stdDev=4,205,507, normalized=604.766; compares: mean=2,362,994; stdDev=186, normalized=2.616}

2023-03-12 23:38:32 INFO SorterBenchmark - run: sort 80,000 elements using SorterBenchmark on class java.lang.Integer from 80,000 total elements and 10 runs using sorter:

QuickSort_DualPivot

2023-03-12 23:38:32 INFO Benchmark_Timer - Begin run: Instrumenting helper for QuickSort_DualPivot with 80,000 elements with 10 runs

2023-03-12 23:40:07 INFO TimeLogger - Raw time per run (mSec): 8423.70

2023-03-12 23:40:07 INFO TimeLogger - Normalized time per run (n log n): 1182.16

2023-03-12 23:40:07 INFO SortBenchmark - QuickSort_DualPivot: StatPack {hits: mean=4,145,994; stdDev=140,997, normalized=4.590; copies: 0, normalized=0.000; inversions: <unset>; swaps: mean=652,089; stdDev=26,430, normalized=0.722; fixes: mean=1,779,514,334; stdDev=101,948,113, normalized=1970.271; compares: mean=1,572,553; stdDev=43,050, normalized=1.741}

2023-03-12 23:40:07 INFO SorterBenchmark - run: sort 160,000 elements using SorterBenchmark on class java.lang.Integer from 160,000 total elements and 10 runs using sorter: MergeSortBasic

2023-03-12 23:40:07 INFO Benchmark_Timer - Begin run: Instrumenting helper for MergeSortBasic with 160,000 elements with 10 runs

2023-03-12 23:40:08 INFO TimeLogger - Raw time per run (mSec): 24.90

2023-03-12 23:40:08 INFO TimeLogger - Normalized time per run (n log n): 1.64

2023-03-12 23:40:08 INFO SortBenchmark - MergeSortBasic: StatPack {hits: mean=10,224,385; stdDev=1,160, normalized=5.333; copies: 4,800,000, normalized=2.504; inversions: <unset>; swaps: mean=156,096; stdDev=290, normalized=0.081; fixes: mean=2,104,605,835; stdDev=5,280,291, normalized=1097.710; compares: mean=2,584,017; stdDev=326, normalized=1.348}

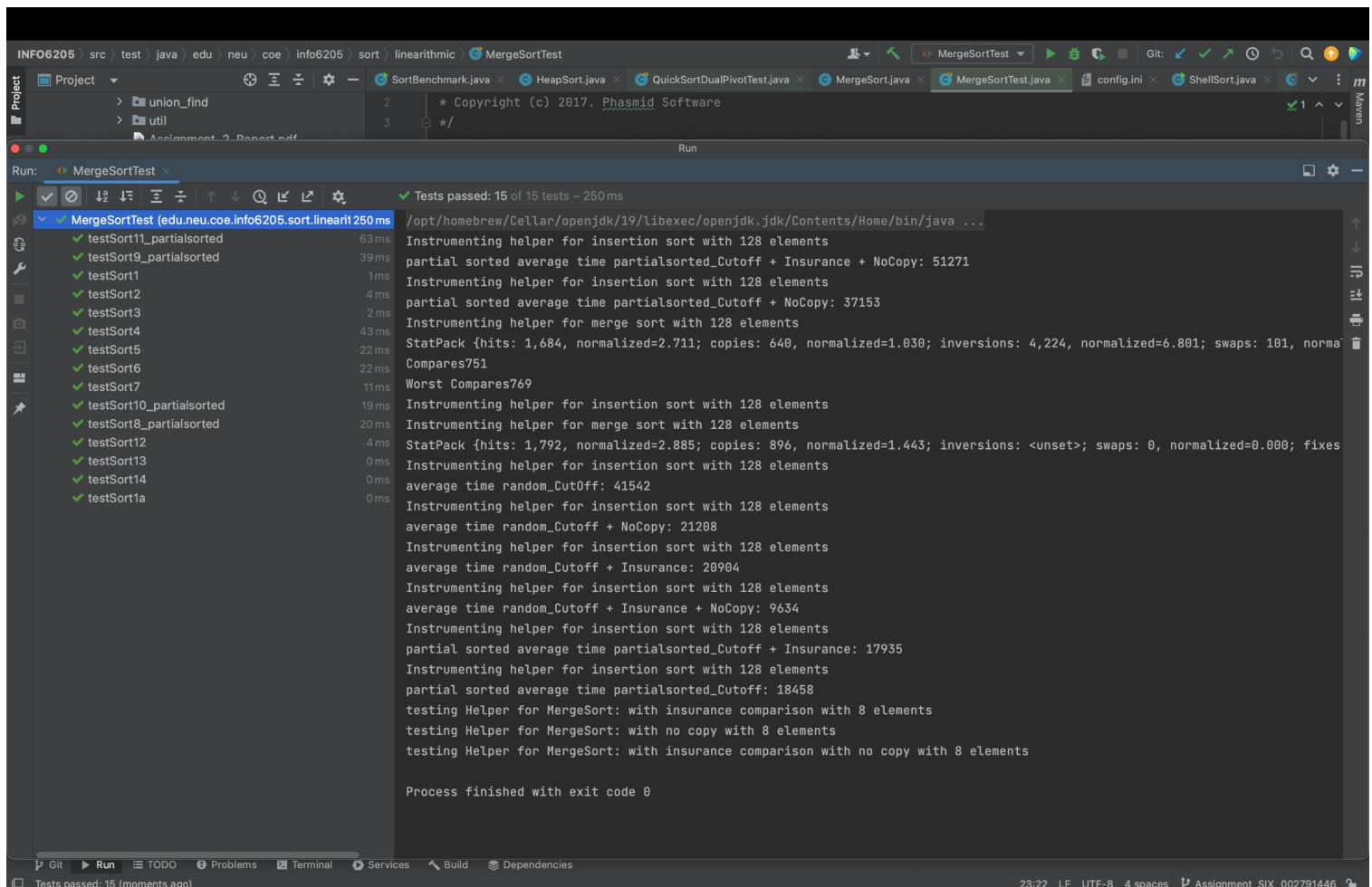
2023-03-12 23:40:08 INFO SorterBenchmark - run: sort 160,000 elements using SorterBenchmark on class java.lang.Integer from 160,000 total elements and 10 runs using sorter: HeapSort

2023-03-12 23:40:08 INFO Benchmark_Timer - Begin run: Instrumenting helper for HeapSort with 160,000 elements with 10 runs

2023-03-12 23:50:55 INFO TimeLogger - Raw time per run (mSec): 53696.60

2023-03-12 23:50:55 INFO TimeLogger - Normalized time per run (n log n): 3530.05
2023-03-12 23:50:55 INFO SortBenchmark - HeapSort: StatPack {hits: mean=20,600,636; stdDev=1,325, normalized=10.745; copies: 0, normalized=0.000; inversions: <unset>; swaps: mean=2,627,173; stdDev=214, normalized=1.370; fixes: mean=-2,099,903,435; stdDev=10,658,115, normalized=-1095.258; compares: mean=5,045,973; stdDev=281, normalized=2.632}
2023-03-12 23:50:55 INFO SorterBenchmark - run: sort 160,000 elements using SorterBenchmark on class java.lang.Integer from 160,000 total elements and 10 runs using sorter: QuickSort_DualPivot
2023-03-12 23:50:55 INFO Benchmark_Timer - Begin run: Instrumenting helper for QuickSort_DualPivot with 160,000 elements with 10 runs
2023-03-12 23:59:30 INFO TimeLogger - Raw time per run (mSec): 44697.20
2023-03-12 23:59:30 INFO TimeLogger - Normalized time per run (n log n): 2938.43
2023-03-12 23:59:30 INFO SortBenchmark - QuickSort_DualPivot: StatPack {hits: mean=9,019,253; stdDev=319,935, normalized=4.704; copies: 0, normalized=0.000; inversions: <unset>; swaps: mean=1,411,525; stdDev=69,092, normalized=0.736; fixes: mean=-1,196,680,823; stdDev=854,876,322, normalized=-624.159; compares: mean=3,442,560; stdDev=109,698, normalized=1.796}

Process finished with exit code 0



The screenshot shows an IDE window with a project named 'INFO6205'. The 'Run' tab is active, displaying test results for 'MergeSortTest (edu.neu.coe.info6205.sort.linearit 250 ms)'. The test results are as follows:

Test Name	Time
testSort11_partialsorted	63ms
testSort9_partialsorted	39ms
testSort1	1ms
testSort2	4ms
testSort3	2ms
testSort4	43ms
testSort5	22ms
testSort6	22ms
testSort7	11ms
testSort10_partialsorted	19ms
testSort8_partialsorted	20ms
testSort12	4ms
testSort13	0ms
testSort14	0ms
testSort1a	0ms

The output text shows the following details:

- Tests passed: 15 of 15 tests - 250 ms
- Instrumenting helper for insertion sort with 128 elements
- partial sorted average time partialsorted_Cutoff + Insurance + NoCopy: 51271
- Instrumenting helper for insertion sort with 128 elements
- partial sorted average time partialsorted_Cutoff + NoCopy: 37153
- Instrumenting helper for merge sort with 128 elements
- StatPack {hits: 1,684, normalized=2.711; copies: 640, normalized=1.030; inversions: 4,224, normalized=6.801; swaps: 101, normalized=0.000; compares: 751, normalized=0.000}
- Worst Compares 769
- Instrumenting helper for insertion sort with 128 elements
- Instrumenting helper for merge sort with 128 elements
- StatPack {hits: 1,792, normalized=2.885; copies: 896, normalized=1.443; inversions: <unset>; swaps: 0, normalized=0.000; fixes: 0, normalized=0.000}
- Instrumenting helper for insertion sort with 128 elements
- average time random_Cutoff: 41542
- Instrumenting helper for insertion sort with 128 elements
- average time random_Cutoff + NoCopy: 21208
- Instrumenting helper for insertion sort with 128 elements
- average time random_Cutoff + Insurance: 20904
- Instrumenting helper for insertion sort with 128 elements
- average time random_Cutoff + Insurance + NoCopy: 9634
- Instrumenting helper for insertion sort with 128 elements
- partial sorted average time partialsorted_Cutoff + Insurance: 17935
- Instrumenting helper for insertion sort with 128 elements
- partial sorted average time partialsorted_Cutoff: 18458
- testing Helper for MergeSort: with insurance comparison with 8 elements
- testing Helper for MergeSort: with no copy with 8 elements
- testing Helper for MergeSort: with insurance comparison with no copy with 8 elements
- Process finished with exit code 0

INFO6205 › src › test › java › edu › neu › coe › info6205 › sort › elementary › HeapSortTest

Project: union_find, util, Assignment_2_Report.pdf

QuickSortDualPivotTest.java, MergeSort.java, MergeSortTest.java, config.ini, ShellSort.java, InsertionSortTest.java, HeapSortTest.java

1 /*
2 * Copyright (c) 2017. Phasmid Software

Run: HeapSortTest

Tests passed: 5 of 5 tests - 110 ms

- ✓ HeapSortTest (edu.neu.coe.info6205.sort.element 110 ms)
/opt/homebrew/Cellar/openjdk/19/libexec/openjdk.jdk/Contents/Home/bin/java ...
- ✓ testMutatingHeapSort 95 ms
- ✓ sort0 7 ms
- ✓ sort1 5 ms
- ✓ sort2 2 ms
- ✓ sort3 1 ms

Helper for HeapSort with 4 elements

Process finished with exit code 0

Git, Run, TODO, Problems, Terminal, Services, Build, Dependencies

Tests passed: 5 (moments ago)

19:36 LF UTF-8 4 spaces Assignment_SIX_002791446

INFO6205 src > test > java > edu > neu > coe > info6205 > sort > linearithmic QuickSortDualPivotTest

Project

- union_find
- util
- Assignment_2_Report.pdf

2 * Copyright (c) 2017. Phasmid Software

3 */

Run

Run: QuickSortDualPivotTest

Tests passed: 15 of 15 tests - 32ms

QuickSortDualPivotTest (edu.neu.coe.info6205.sort) 32 ms

- testSort 10 ms
- testSortWithInstrumenting6a 3 ms
- testSortWithInstrumenting6b 1 ms
- testSortWithInstrumenting6c 0 ms
- testPartition1 1 ms
- testPartition2 1 ms
- testSortWithInstrumenting0 1 ms
- testSortWithInstrumenting1 2 ms
- testSortWithInstrumenting2 2 ms
- testSortWithInstrumenting3 1 ms
- testSortWithInstrumenting4 1 ms
- testSortWithInstrumenting5 0 ms
- testSortWithInstrumenting7 1 ms
- testPartitionWithSort 1 ms
- testSortDetailed 7 ms

/opt/homebrew/Cellar/openjdk/19/libexec/openjdk.jdk/Contents/Home/bin/java ...

Instrumenting helper for quick sort dual pivot with 128 elements

StatPack {hits: 2,619, normalized=4.217; copies: 0, normalized=0.000; inversions: 4,224, normalized=6.881; swaps: 435, normalized=0.000; compares: 950, worstCompares: 1242}

Process finished with exit code 0

Git Run TODO Problems Terminal Services Build Dependencies

Tests passed: 15 (moments ago)

21:1 (19 chars, 1 line break) LF UTF-8 4 spaces Assignment_SIX_002791446