Program structures and Algorithm’s

Fall 23

Name: Amar Nagargoje
NUID: 002273113

TASK:

To determine the best predictor of total execution time by running benchmarks for merge sort, quick sort (dual-pivot), and heap sort. Following parameters are checked for determining the best predictor

a. swaps
b. compares c. copies
d. hits

RELATIONSHIP CONCLUSION:

The frequency of array access, denoted as "hits," emerges as the foremost indicator of the overall execution time across comparison-based sorting algorithms like quick sort, merge sort, and heap sort. Array access significantly influences sorting algorithm performance due to its potential time and resource costs. Moreover, the manner in which array elements are accessed can impact sorting algorithm efficiency. For instance, algorithms requiring random element access may experience frequent cache misses, necessitating data retrieval from main memory, which is slower than accessing cached data.

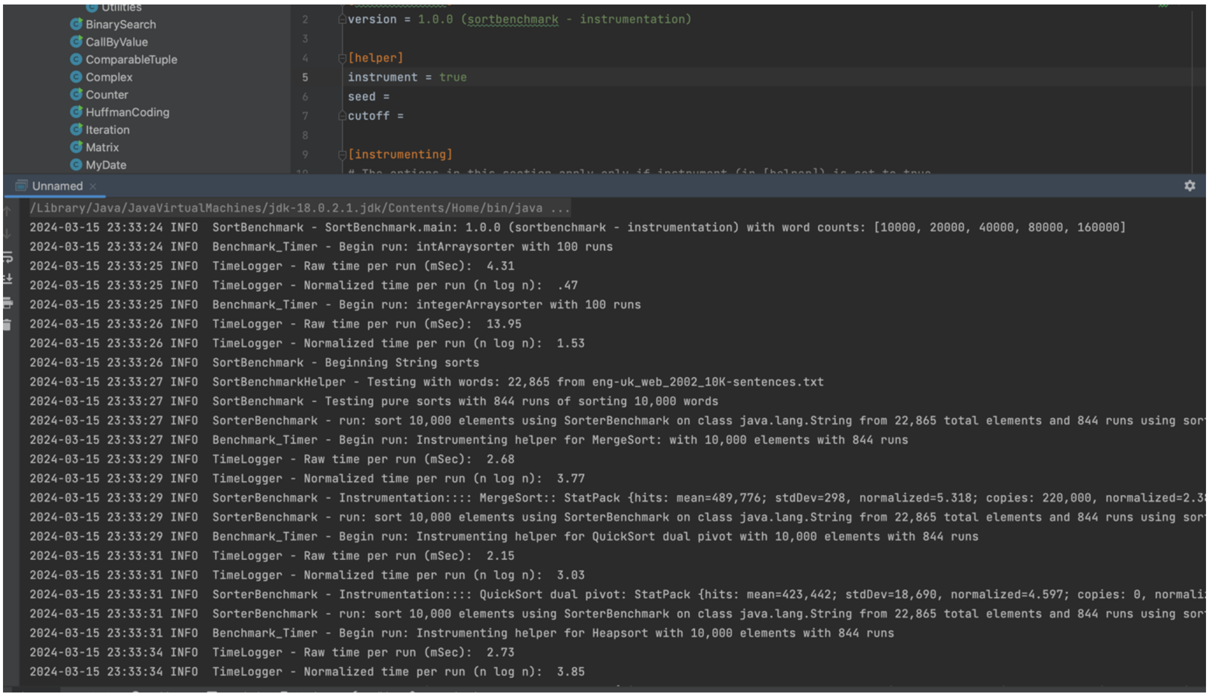
In merge sort, additional memory is necessitated for copying values, with no swapping involved. During value copying from auxiliary memory, comparisons facilitate pointer movement in sub-arrays. Consequently, the descending order of impact on total execution time is hits, copies, comparisons, and swaps (which are nearly negligible).

Conversely, heap sort and quicksort do not require additional memory, eliminating time spent on copy operations. Instead, they establish proper partitions within the array by continuously comparing and swapping elements. Consequently, the descending order of impact on total execution time is hits, comparisons, and swaps, with copying playing no role.

Furthermore, when plotting normalized metrics, the hits graph closely mirrors the time graph across all sorting algorithms, suggesting it serves as the most reliable predictor of total execution time.

EVIDENCE TO SUPPORT THAT CONCLUSION

Instrumentation:

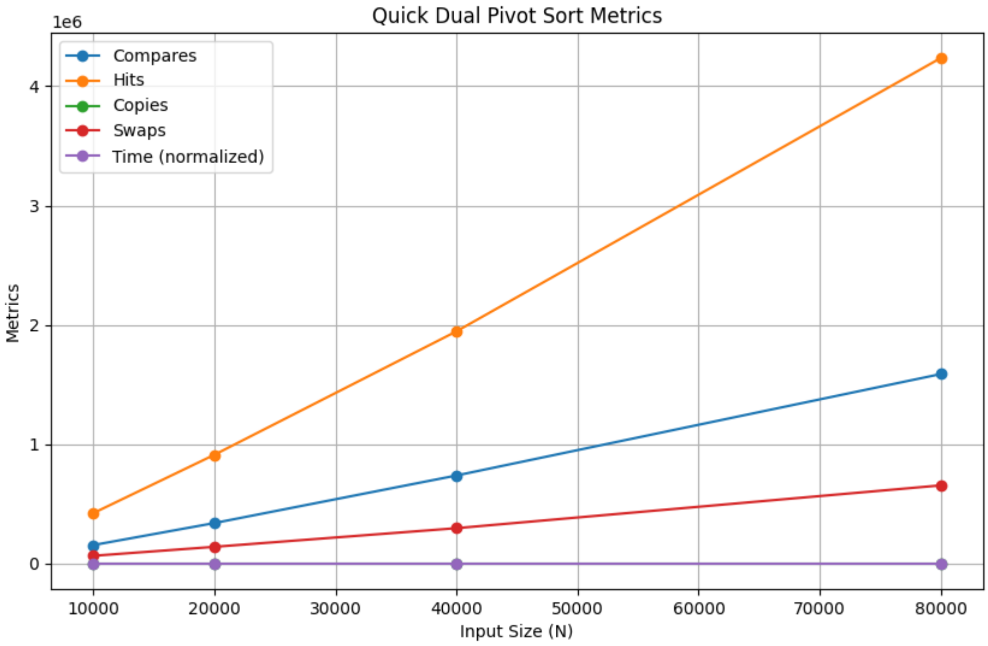


```
version = 1.0.0 (sortbenchmark - instrumentation)

[helper]
instrument = true
seed =
cutoff =

[instrumenting]
# The options in this section apply only if instrument (in [helper]) is set to true
```

```
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
2024-03-15 23:33:24 INFO  SortBenchmark - SortBenchmark.main: 1.0.0 (sortbenchmark - instrumentation) with word counts: [10000, 20000, 40000, 80000, 160000]
2024-03-15 23:33:24 INFO  Benchmark_Timer - Begin run: intArraysorter with 100 runs
2024-03-15 23:33:25 INFO  TimeLogger - Raw time per run (mSec):  4.31
2024-03-15 23:33:25 INFO  TimeLogger - Normalized time per run (n log n):  .47
2024-03-15 23:33:25 INFO  Benchmark_Timer - Begin run: integerArraysorter with 100 runs
2024-03-15 23:33:26 INFO  TimeLogger - Raw time per run (mSec):  13.95
2024-03-15 23:33:26 INFO  TimeLogger - Normalized time per run (n log n):  1.53
2024-03-15 23:33:26 INFO  SortBenchmark - Beginning String sorts
2024-03-15 23:33:27 INFO  SortBenchmarkHelper - Testing with words: 22,865 from eng-uk_web_2002_10K-sentences.txt
2024-03-15 23:33:27 INFO  SortBenchmark - Testing pure sorts with 844 runs of sorting 10,000 words
2024-03-15 23:33:27 INFO  SorterBenchmark - run: sort 10,000 elements using SorterBenchmark on class java.lang.String from 22,865 total elements and 844 runs using sor
2024-03-15 23:33:27 INFO  Benchmark_Timer - Begin run: Instrumenting helper for MergeSort: with 10,000 elements with 844 runs
2024-03-15 23:33:29 INFO  TimeLogger - Raw time per run (mSec):  2.68
2024-03-15 23:33:29 INFO  TimeLogger - Normalized time per run (n log n):  3.77
2024-03-15 23:33:29 INFO  SorterBenchmark - Instrumentation:::: MergeSort:: StatPack {hits: mean=489,776; stdDev=298, normalized=5.318; copies: 220,000, normalized=2.3
2024-03-15 23:33:29 INFO  SorterBenchmark - run: sort 10,000 elements using SorterBenchmark on class java.lang.String from 22,865 total elements and 844 runs using sor
2024-03-15 23:33:29 INFO  Benchmark_Timer - Begin run: Instrumenting helper for QuickSort dual pivot with 10,000 elements with 844 runs
2024-03-15 23:33:31 INFO  TimeLogger - Raw time per run (mSec):  2.15
2024-03-15 23:33:31 INFO  TimeLogger - Normalized time per run (n log n):  3.03
2024-03-15 23:33:31 INFO  SorterBenchmark - Instrumentation:::: QuickSort dual pivot: StatPack {hits: mean=423,442; stdDev=18,690, normalized=4.597; copies: 0, normali
2024-03-15 23:33:31 INFO  SorterBenchmark - run: sort 10,000 elements using SorterBenchmark on class java.lang.String from 22,865 total elements and 844 runs using sor
2024-03-15 23:33:31 INFO  Benchmark_Timer - Begin run: Instrumenting helper for Heapsort with 10,000 elements with 844 runs
2024-03-15 23:33:34 INFO  TimeLogger - Raw time per run (mSec):  2.73
2024-03-15 23:33:34 INFO  TimeLogger - Normalized time per run (n log n):  3.85
```

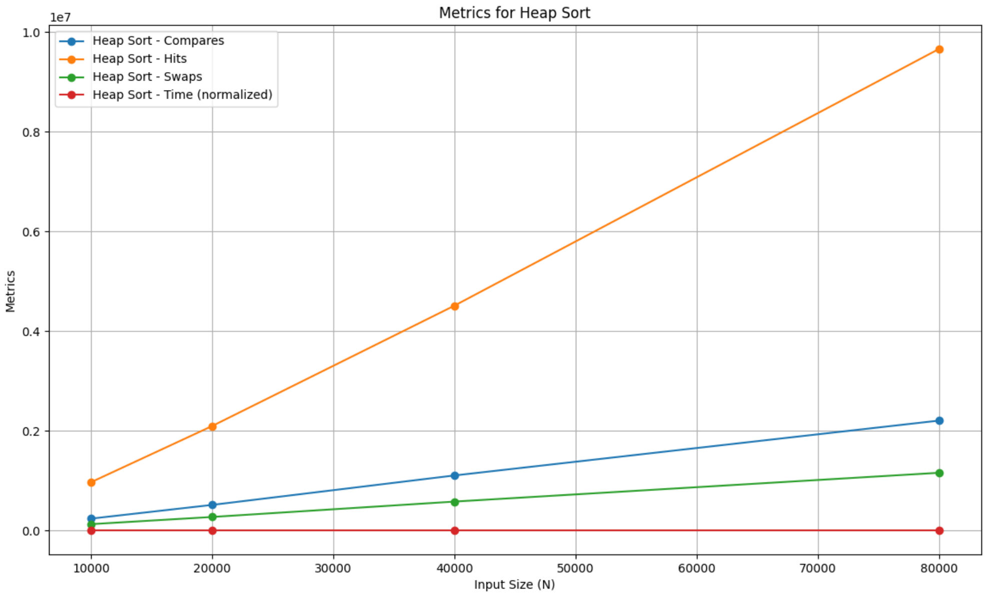
Quick Dual Pivot Sort

N	Compares (mean)	Hits (mean)	Copies (mean)	Swaps (mean)	Time (normalized)
10000	156,086	423,442	0	66,395	1.53
20000	340,671	912,283	0	141,627	2.92
40000	739,229	1,946,617	0	297,881	4.60
80000	1,588,867	4,236,166	0	656,359	4.69



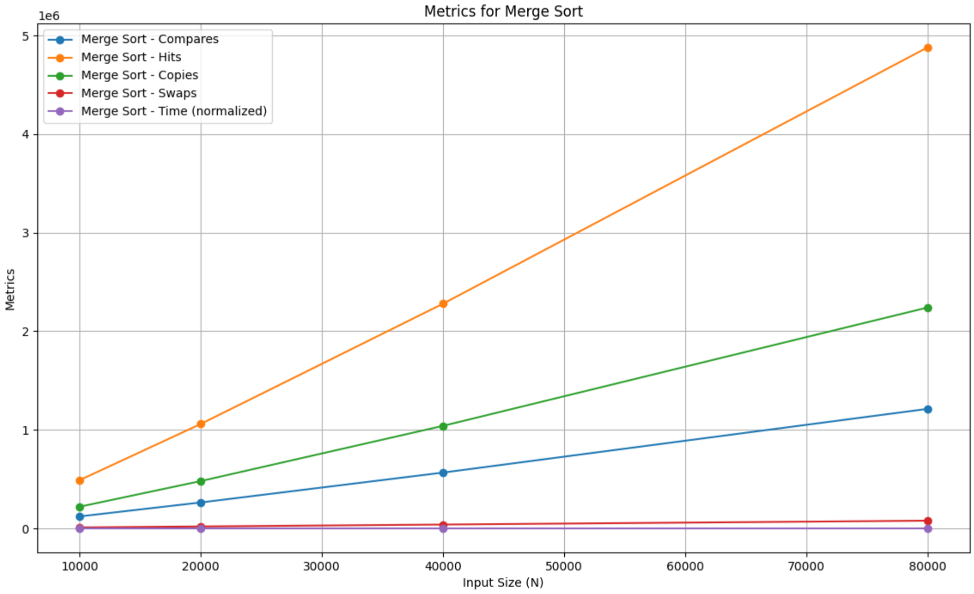
Heap Sort

N	Compares (mean)	Hits (mean)	Copies (mean)	Swaps (mean)	Time (normalized)
10000	235,370	967,555	0	124,204	3.85
20000	510,750	2,095,111	0	268,403	4.01
40000	1,101,504	4,510,197	0	576,797	4.67
80000	2,202,016	9,660,173	0	1,154,766	10.64

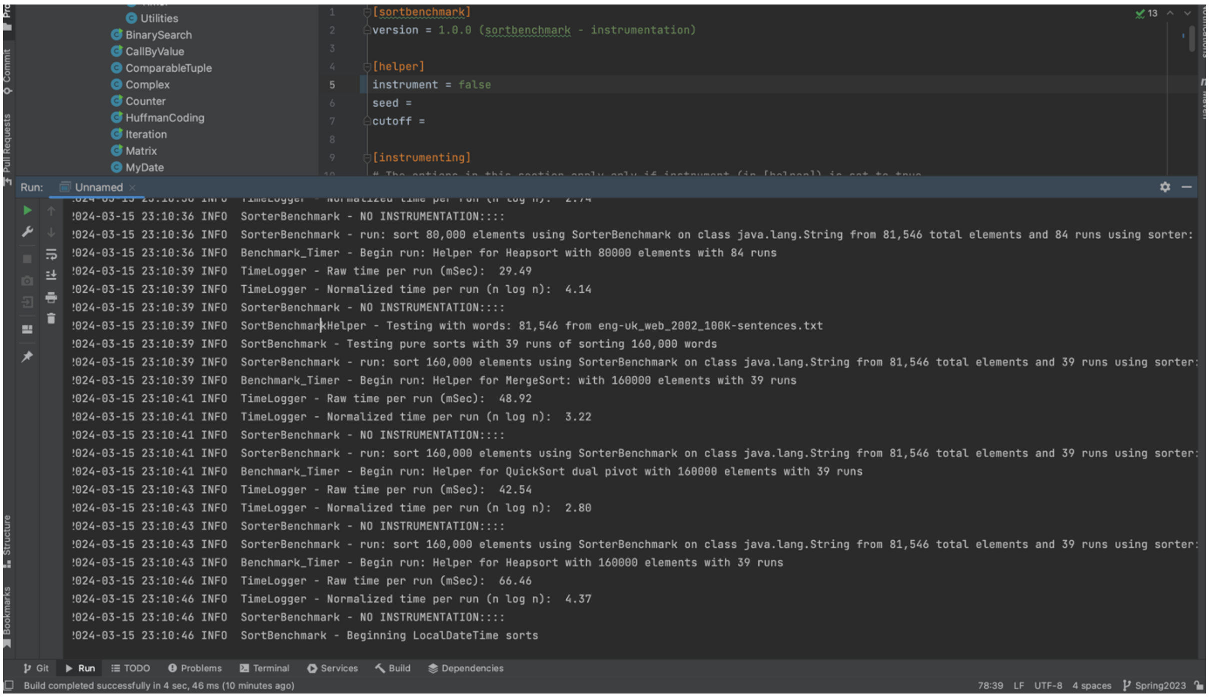


Merge Sort

N	Compares (mean)	Hits (mean)	Copies (mean)	Swaps (mean)	Time (normalized)
10000	121,501	489,776	220,000	9,758	3.77
20000	263,010	1,059,567	480,000	19,518	3.35
40000	566,002	2,279,038	1,040,000	39,010	3.62
80000	1,212,033	4,878,269	2,240,000	78,082	3.62



Non Instrumentation:



```
[sortbenchmark]
version = 1.0.0 (sortbenchmark - instrumentation)

[helper]
instrument = false
seed =
cutoff =

[instrumenting]
```

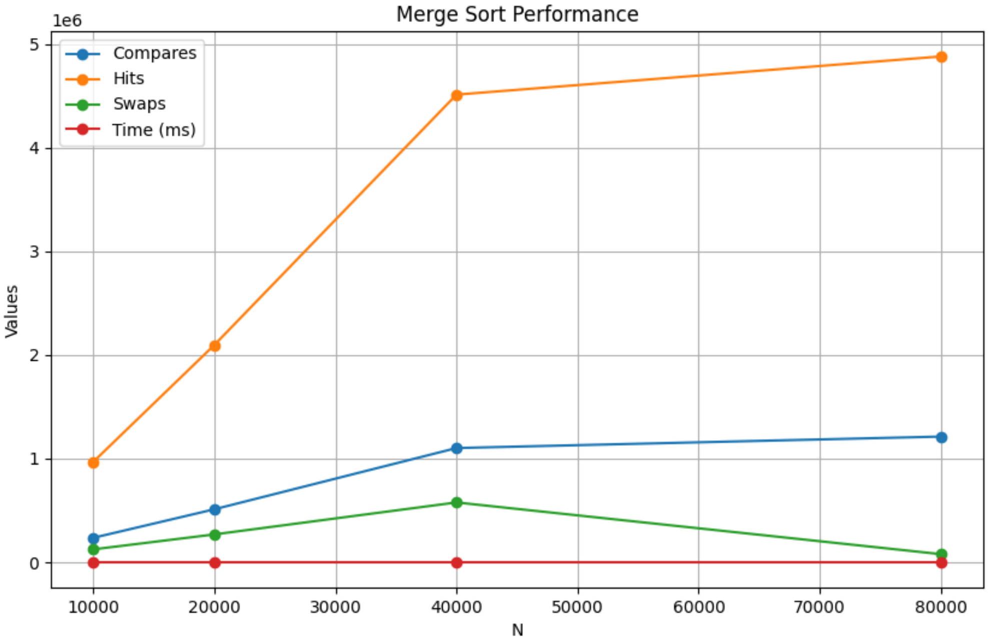
```
!024-03-15 23:10:36 INFO  SorterBenchmark - NO INSTRUMENTATION::::
!024-03-15 23:10:36 INFO  SorterBenchmark - run: sort 80,000 elements using SorterBenchmark on class java.lang.String from 81,546 total elements and 84 runs using sorter:
!024-03-15 23:10:36 INFO  Benchmark_Timer - Begin run: Helper for Heapsort with 80000 elements with 84 runs
!024-03-15 23:10:39 INFO  TimeLogger - Raw time per run (mSec):  29.49
!024-03-15 23:10:39 INFO  TimeLogger - Normalized time per run (n log n):  4.14
!024-03-15 23:10:39 INFO  SorterBenchmark - NO INSTRUMENTATION::::
!024-03-15 23:10:39 INFO  SortBenchmarkHelper - Testing with words: 81,546 from eng-uk_web_2002_100K-sentences.txt
!024-03-15 23:10:39 INFO  SortBenchmark - Testing pure sorts with 39 runs of sorting 160,000 words
!024-03-15 23:10:39 INFO  SorterBenchmark - run: sort 160,000 elements using SorterBenchmark on class java.lang.String from 81,546 total elements and 39 runs using sorter:
!024-03-15 23:10:39 INFO  Benchmark_Timer - Begin run: Helper for MergeSort: with 160000 elements with 39 runs
!024-03-15 23:10:41 INFO  TimeLogger - Raw time per run (mSec):  48.92
!024-03-15 23:10:41 INFO  TimeLogger - Normalized time per run (n log n):  3.22
!024-03-15 23:10:41 INFO  SorterBenchmark - NO INSTRUMENTATION::::
!024-03-15 23:10:41 INFO  SorterBenchmark - run: sort 160,000 elements using SorterBenchmark on class java.lang.String from 81,546 total elements and 39 runs using sorter:
!024-03-15 23:10:41 INFO  Benchmark_Timer - Begin run: Helper for QuickSort dual pivot with 160000 elements with 39 runs
!024-03-15 23:10:43 INFO  TimeLogger - Raw time per run (mSec):  42.54
!024-03-15 23:10:43 INFO  TimeLogger - Normalized time per run (n log n):  2.80
!024-03-15 23:10:43 INFO  SorterBenchmark - NO INSTRUMENTATION::::
!024-03-15 23:10:43 INFO  SorterBenchmark - run: sort 160,000 elements using SorterBenchmark on class java.lang.String from 81,546 total elements and 39 runs using sorter:
!024-03-15 23:10:43 INFO  Benchmark_Timer - Begin run: Helper for Heapsort with 160000 elements with 39 runs
!024-03-15 23:10:46 INFO  TimeLogger - Raw time per run (mSec):  66.46
!024-03-15 23:10:46 INFO  TimeLogger - Normalized time per run (n log n):  4.37
!024-03-15 23:10:46 INFO  SorterBenchmark - NO INSTRUMENTATION::::
!024-03-15 23:10:46 INFO  SortBenchmark - Beginning LocalDateTime sorts
```

Build completed successfully in 4 sec, 46 ms (10 minutes ago)

Merge Sort

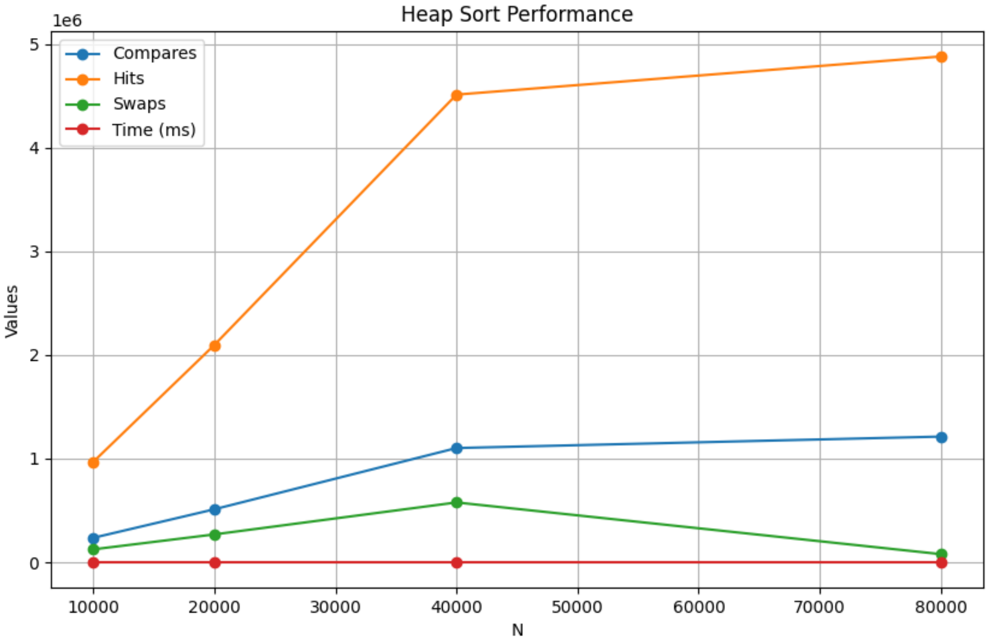
Merge Sort Table:

N	Compares	Swaps	Hits	Time (ms)
10000	235369	124200	967539	2.58
20000	510746	268401	2095097	5.75
40000	1101501	576802	4510210	12.81
80000	1212028	78082	4878279	29.49



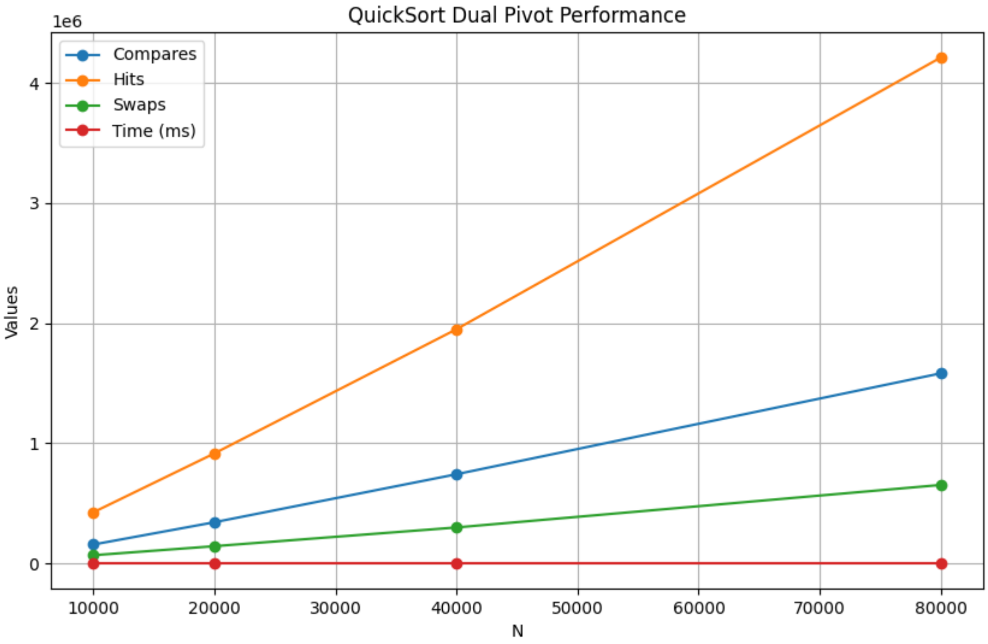
Heap Sort

N	Compares	Hits	Swaps	Time (ms)
10,000	235,369	967,539	124,200	2.58
20,000	510,746	2,095,097	268,401	5.75
40,000	1,101,501	4,510,210	576,802	12.81
80,000	1,212,028	4,878,279	78,082	29.49

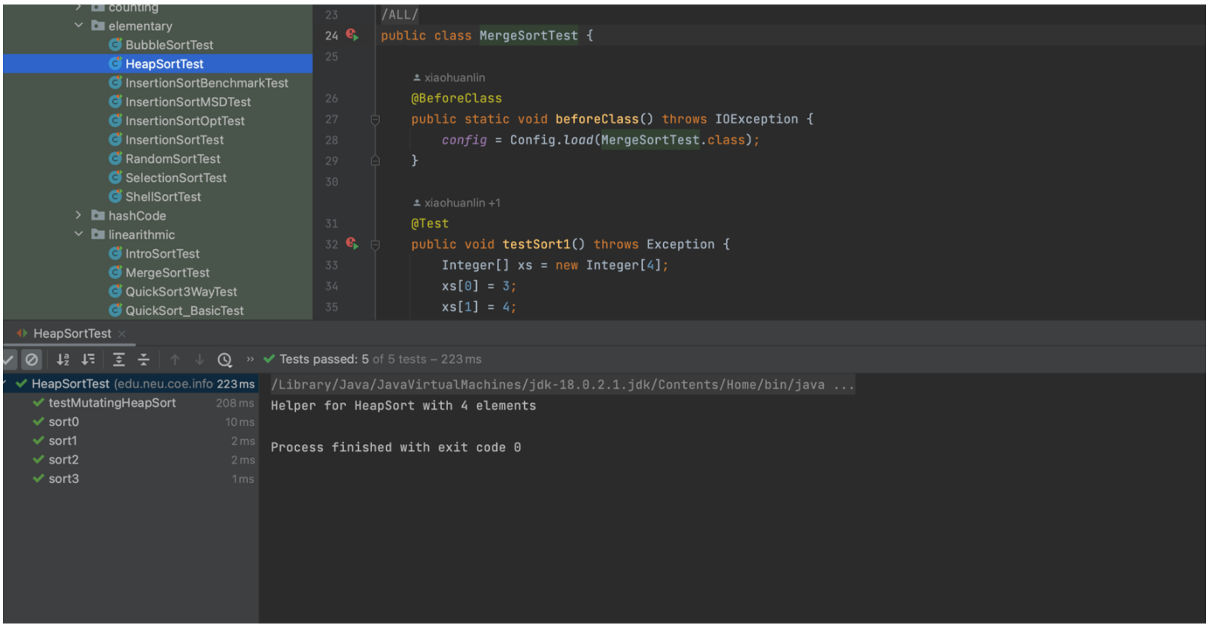


QuickSort dual pivot

N	Compares	Hits	Swaps	Time (ms)
10,000	155,615	423,690	66,575	2.02
20,000	341,262	914,287	141,985	4.29
40,000	741,161	1,948,759	297,931	9.34
80,000	1,581,480	4,211,701	652,095	19.55



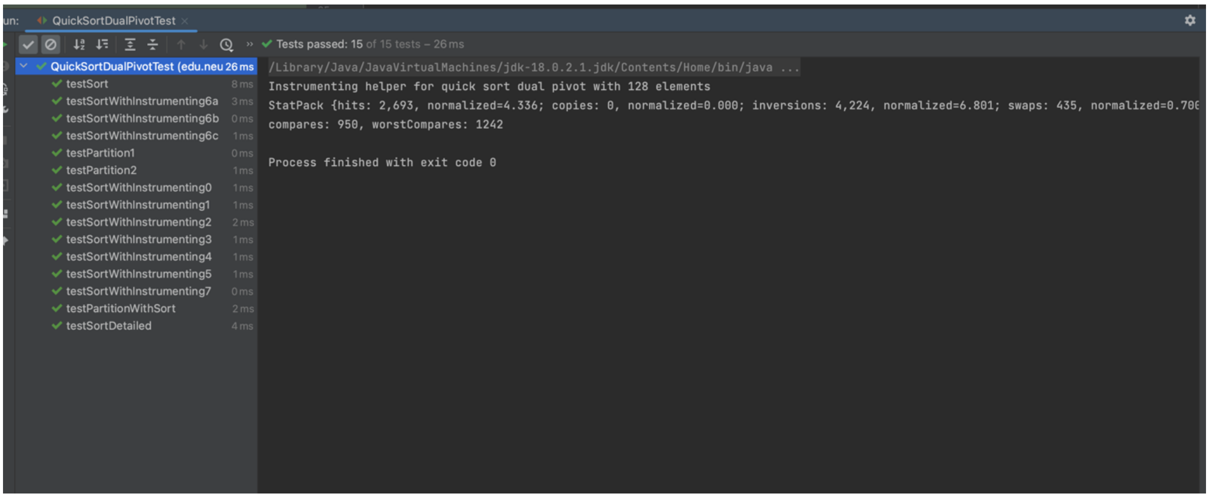
Test case SS:



```java
/ALL/
public class MergeSortTest {

    @BeforeClass
    public static void beforeClass() throws IOException {
        config = Config.load(MergeSortTest.class);
    }

    @Test
    public void testSort1() throws Exception {
        Integer[] xs = new Integer[4];
        xs[0] = 3;
        xs[1] = 4;
```

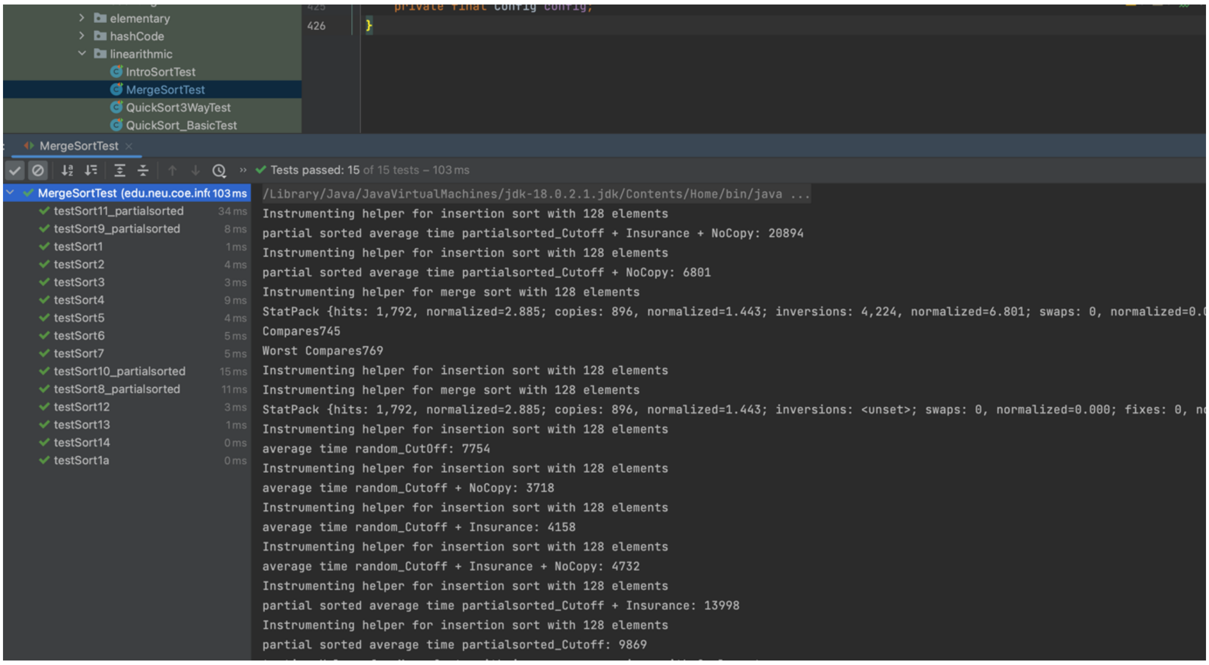
```
Tests passed: 5 of 5 tests – 223 ms

HeapSortTest (edu.neu.coe.info 223 ms
  testMutatingHeapSort   208 ms
  sort0                   10 ms
  sort1                    2 ms
  sort2                    2 ms
  sort3                    1 ms
```

```
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Helper for HeapSort with 4 elements

Process finished with exit code 0
```



```
Tests passed: 15 of 15 tests – 26 ms

QuickSortDualPivotTest (edu.neu 26 ms
  testSort                         8 ms
  testSortWithInstrumenting6a      3 ms
  testSortWithInstrumenting6b      0 ms
  testSortWithInstrumenting6c      1 ms
  testPartition1                   0 ms
  testPartition2                   1 ms
  testSortWithInstrumenting0       1 ms
  testSortWithInstrumenting1       1 ms
  testSortWithInstrumenting2       2 ms
  testSortWithInstrumenting3       1 ms
  testSortWithInstrumenting4       1 ms
  testSortWithInstrumenting5       1 ms
  testSortWithInstrumenting7       0 ms
  testPartitionWithSort            2 ms
  testSortDetailed                 4 ms
```

```
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Instrumenting helper for quick sort dual pivot with 128 elements
StatPack {hits: 2,693, normalized=4.336; copies: 0, normalized=0.000; inversions: 4,224, normalized=6.801; swaps: 435, normalized=0.70
compares: 950, worstCompares: 1242

Process finished with exit code 0
```



```
    private final Config config;
}
```

```
✓ Tests passed: 15 of 15 tests – 103 ms

✓ MergeSortTest (edu.neu.coe.inf  103 ms
    ✓ testSort11_partialsorted      34 ms
    ✓ testSort9_partialsorted        8 ms
    ✓ testSort1                      1 ms
    ✓ testSort2                      4 ms
    ✓ testSort3                      3 ms
    ✓ testSort4                      9 ms
    ✓ testSort5                      4 ms
    ✓ testSort6                      5 ms
    ✓ testSort7                      5 ms
    ✓ testSort10_partialsorted      15 ms
    ✓ testSort8_partialsorted       11 ms
    ✓ testSort12                     3 ms
    ✓ testSort13                     1 ms
    ✓ testSort14                     0 ms
    ✓ testSort1a                     0 ms
```

```
/Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Instrumenting helper for insertion sort with 128 elements
partial sorted average time partialsorted_Cutoff + Insurance + NoCopy: 20894
Instrumenting helper for insertion sort with 128 elements
partial sorted average time partialsorted_Cutoff + NoCopy: 6801
Instrumenting helper for merge sort with 128 elements
StatPack {hits: 1,792, normalized=2.885; copies: 896, normalized=1.443; inversions: 4,224, normalized=6.801; swaps: 0, normalized=0.0
Compares745
Worst Compares769
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, no
Instrumenting helper for insertion sort with 128 elements
average time random_CutOff: 7754
Instrumenting helper for insertion sort with 128 elements
average time random_Cutoff + NoCopy: 3718
Instrumenting helper for insertion sort with 128 elements
average time random_Cutoff + Insurance: 4158
Instrumenting helper for insertion sort with 128 elements
average time random_Cutoff + Insurance + NoCopy: 4732
Instrumenting helper for insertion sort with 128 elements
partial sorted average time partialsorted_Cutoff + Insurance: 13998
Instrumenting helper for insertion sort with 128 elements
partial sorted average time partialsorted_Cutoff: 9869
```