

Assignment 6

Main Code Changes:

```
package edu.neu.coe.info6205.util;

import edu.neu.coe.info6205.sort.BaseHelper;
import edu.neu.coe.info6205.sort.InstrumentedHelper;
import edu.neu.coe.info6205.sort.elementary.HeapSort;
import edu.neu.coe.info6205.sort.linearithmic.MergeSortBasic;
import edu.neu.coe.info6205.sort.linearithmic.QuickSort;
import edu.neu.coe.info6205.sort.linearithmic.QuickSort_DualPivot;
import edu.neu.coe.info6205.util.Benchmark_Timer;
import edu.neu.coe.info6205.util.Config;
import edu.neu.coe.info6205.util.PrivateMethodTester;
import edu.neu.coe.info6205.util.StatPack;

import java.util.Random;
import java.util.function.Consumer;
import java.util.function.Supplier;

public class NewSortBenchmark {

    public static void main(String[] args) {

        int n = 10000;

        final Config config = Config.setupConfig("true", "0", "1", "1", "");

        // Merge Sort
        BaseHelper<Integer> h1 = new InstrumentedHelper<>("test", config);

        MergeSortBasic<Integer> merge = new MergeSortBasic<>(h1);

        Consumer<Integer[]> rF1 = rA1 -> merge.sort(rA1);
        Benchmark_Timer<Integer[]> rT1 = new Benchmark_Timer<>("Sorts array of " +
+ n + " elements", rF1);
        Supplier<Integer[]> r1 = () -> {
            Random randI = new Random();
            Integer[] rA1 = new Integer[n];
            for(int i=0; i<n; i++) {
                int randInt = randI.nextInt(n);
                rA1[i] = randInt+1;
            }
            return rA1;
        };
        rF1.accept(r1.get());
        double randTime1 = rT1.run(r1.get(), 1);
```

```

        System.out.println("Time taken for " + n + " elements of array using
Merge Sort is : " + randTime1);

        h1.postProcess(merge.sort(r1.get()));

        PrivateMethodTester privateMethodTester1 = new PrivateMethodTester(h1);
        StatPack statPack1 = (StatPack)
privateMethodTester1.invokePrivate("getStatPack");

        int compares_merge = (int)
statPack1.getStatistics(InstrumentedHelper.COMPARES).mean();
        int hits_merge = (int)
statPack1.getStatistics(InstrumentedHelper.HITS).mean();
        int swaps_merge = (int)
statPack1.getStatistics(InstrumentedHelper.SWAPS).mean();

        System.out.println("No of compares taken using Merge Sort : " +
compares_merge);
        System.out.println("No of hits taken using Merge Sort : " + hits_merge);
        System.out.println("No of swaps taken using Merge Sort : " +
swaps_merge);

        // Quick Sort
        BaseHelper<Integer> h2 = new InstrumentedHelper<>("test", config);

        QuickSort<Integer> quick = new QuickSort_DualPivot<>(h2);

        Consumer<Integer[]> rF2 = rA2 -> quick.sort(rA2);
        Benchmark_Timer<Integer[]> rT2 = new Benchmark_Timer<>("Sorts array of " +
+ n + " elements", rF2);
        Supplier<Integer[]> r2 = () -> {
            Random randI = new Random();
            Integer[] rA2 = new Integer[n];
            for(int i=0; i<n; i++) {
                int randInt = randI.nextInt(n);
                rA2[i] = randInt+1;
            }
            return rA2;
        };
        rF2.accept(r2.get());
        double randTime2 = rT2.run(r2.get(), 1);
        System.out.println("Time taken for " + n + " elements of array using
Quick Sort is : " + randTime2);

        h2.postProcess(quick.sort(r2.get()));

        PrivateMethodTester privateMethodTester = new PrivateMethodTester(h2);
        StatPack statPack2 = (StatPack)
privateMethodTester.invokePrivate("getStatPack");

```

```

        int compares_quick = (int)
statPack2.getStatistics(InstrumentedHelper.COMPARES).mean();
        int hits_quick = (int)
statPack2.getStatistics(InstrumentedHelper.HITS).mean();
        int swaps_quick = (int)
statPack2.getStatistics(InstrumentedHelper.SWAPS).mean();

        System.out.println("No of compares taken using Quick Sort : " +
compares_quick);
        System.out.println("No of hits taken using Quick Sort : " + hits_quick);
        System.out.println("No of swaps taken using Quick Sort : " +
swaps_quick);

        // Heap Sort
BaseHelper<Integer> h3 = new InstrumentedHelper<>("test", config);

        HeapSort<Integer> heap = new HeapSort<>(h3);

        Consumer<Integer[]> rF3 = rA3 -> merge.sort(rA3);
        Benchmark_Timer<Integer[]> rT3 = new Benchmark_Timer<>("Sorts array of " +
+ n + " elements", rF3);
        Supplier<Integer[]> r3 = () -> {
            Random randI = new Random();
            Integer[] rA3 = new Integer[n];
            for(int i=0; i<n; i++) {
                int randInt = randI.nextInt(n);
                rA3[i] = randInt+1;
            }
            return rA3;
        };
        rF3.accept(r3.get());
        double randTime3 = rT3.run(r3.get(), 1);
        System.out.println("Time taken for " + n + " elements of array using
Heap Sort is : " + randTime3);

        h3.postProcess(heap.sort(r3.get()));

        PrivateMethodTester privateMethodTester2 = new PrivateMethodTester(h3);
        StatPack statPack3 = (StatPack)
privateMethodTester2.invokePrivate("getStatPack");

        int compares_heap = (int)
statPack3.getStatistics(InstrumentedHelper.COMPARES).mean();
        int hits_heap = (int)
statPack3.getStatistics(InstrumentedHelper.HITS).mean();
        int swaps_heap = (int)
statPack3.getStatistics(InstrumentedHelper.SWAPS).mean();

```

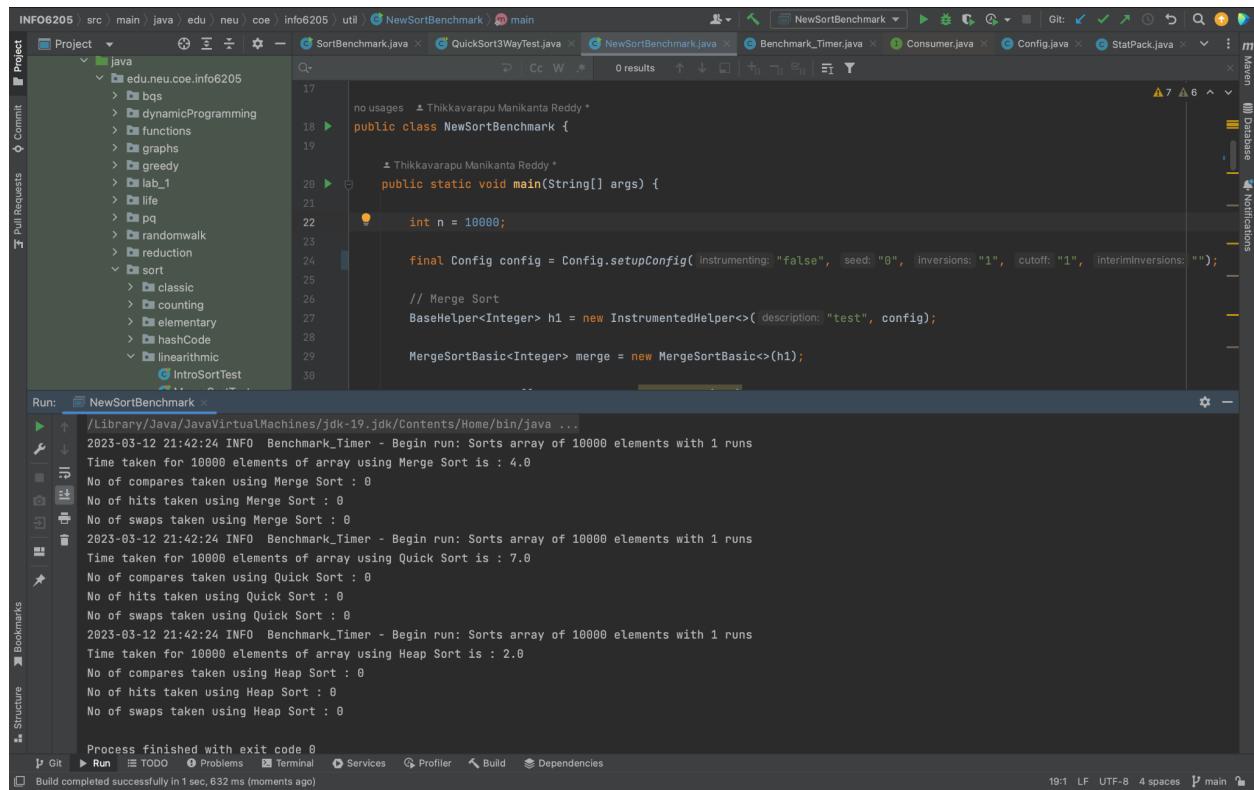
```

        System.out.println("No of compares taken using Heap Sort : " +
compares_heap);
        System.out.println("No of hits taken using Heap Sort : " + hits_heap);
        System.out.println("No of swaps taken using Heap Sort : " + swaps_heap);
    }

}

```

Outputs:



The screenshot shows an IDE interface with the following details:

- Project Structure:** The project is named "INFO6205" and contains a "java" package with various sub-directories like "bfs", "dynamicProgramming", "functions", "graphs", "greedy", "lab_1", "life", "pq", "randomwalk", "reduction", "sort", and "linearithmic".
- Code Editor:** The main file being edited is "NewSortBenchmark.java". It includes imports for "Config", "BaseHelper", and "MergeSortBasic". The code defines a main method that initializes a "Config" object with specific parameters and creates instances of "BaseHelper<Integer>" and "MergeSortBasic<Integer>".
- Run Tab:** The "Run" tab is active, showing three runs:
 - Run 1: Sorts array of 10000 elements with 1 runs. Time taken: 4.0. Metrics: 0 compares, 0 hits, 0 swaps.
 - Run 2: Sorts array of 10000 elements with 1 runs. Time taken: 7.0. Metrics: 0 compares, 0 hits, 0 swaps.
 - Run 3: Sorts array of 10000 elements with 1 runs. Time taken: 2.0. Metrics: 0 compares, 0 hits, 0 swaps.
- Output:** The terminal output shows the results of each run, including the time taken and the number of compares, hits, and swaps for each sorting algorithm (Merge Sort, Quick Sort, and Heap Sort).
- Bottom Status Bar:** The status bar indicates the process finished with exit code 0, the build completed successfully in 1 sec, 632 ms (moments ago), and the current file is "main.java".

INFO6205 src main java edu neu coe info6205 util NewSortBenchmark main

Project Pull Requests Commit Bookmarks Structure

java edu.neu.coe.info6205 bqs dynamicProgramming functions graphs greedy lab_1 life pq randomwalk reduction sort classic counting elementary hashCode linearithmic IntroSortTest

NewSortBenchmark.java QuickSort3WayTest.java NewSortBenchmark.java Benchmark_Timer.java Consumer.java Config.java StatPack.java

17 no usages ▲ Thikkavarapu Manikanta Reddy *

18 ▶ public class NewSortBenchmark { ↗ Thikkavarapu Manikanta Reddy *

19

20 ▶ public static void main(String[] args) { ↗ Thikkavarapu Manikanta Reddy *

21

22

23

24 int n = 10000;

25

26 final Config config = Config.setupConfig(instrumenting: "false", seed: "0", inversions: "1", cutoff: "1", interimInversions: "");

27 // Merge Sort

28 BaseHelper<Integer> h1 = new InstrumentedHelper<>(description: "test", config);

29 MergeSortBasic<Integer> merge = new MergeSortBasic<>(h1);

30

Run: NewSortBenchmark

/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
2023-03-12 21:44:49 INFO Benchmark_Timer - Begin run: Sorts array of 160000 elements with 1 runs
Time taken for 160000 elements of array using Merge Sort is : 61.0
No of compares taken using Merge Sort : 0
No of hits taken using Merge Sort : 0
No of swaps taken using Merge Sort : 0
2023-03-12 21:44:50 INFO Benchmark_Timer - Begin run: Sorts array of 160000 elements with 1 runs
Time taken for 160000 elements of array using Quick Sort is : 48.0
No of compares taken using Quick Sort : 0
No of hits taken using Quick Sort : 0
No of swaps taken using Quick Sort : 0
2023-03-12 21:44:50 INFO Benchmark_Timer - Begin run: Sorts array of 160000 elements with 1 runs
Time taken for 160000 elements of array using Heap Sort is : 41.0
No of compares taken using Heap Sort : 0
No of hits taken using Heap Sort : 0
No of swaps taken using Heap Sort : 0
Process finished with exit code 0

Git Run TODO Problems Terminal Services Profiler Build Dependencies

Build completed successfully in 1 sec, 451 ms (17 minutes ago)

27:75 LF UTF-8 4 spaces main

INFO6205 src main java edu neu coe info6205 util NewSortBenchmark main

Project Pull Requests Commit Bookmarks Structure

java edu.neu.coe.info6205 bqs dynamicProgramming functions graphs greedy lab_1 life pq randomwalk reduction sort classic counting elementary hashCode linearithmic IntroSortTest

NewSortBenchmark.java QuickSort3WayTest.java NewSortBenchmark.java Benchmark_Timer.java Consumer.java Config.java StatPack.java

17 no usages ▲ Thikkavarapu Manikanta Reddy *

18 ▶ public class NewSortBenchmark { ↗ Thikkavarapu Manikanta Reddy *

19

20 ▶ public static void main(String[] args) { ↗ Thikkavarapu Manikanta Reddy *

21

22

23

24 int n = 10000;

25

26 final Config config = Config.setupConfig(instrumenting: "true", seed: "0", inversions: "1", cutoff: "1", interimInversions: "");

27 // Merge Sort

28 BaseHelper<Integer> h1 = new InstrumentedHelper<>(description: "test", config);

29 MergeSortBasic<Integer> merge = new MergeSortBasic<>(h1);

30

Run: NewSortBenchmark

2023-03-12 22:02:16 INFO Benchmark_Timer - Begin run: Sorts array of 10000 elements with 1 runs
Time taken for 10000 elements of array using Merge Sort is : 5.0
No of compares taken using Merge Sort : 120567
No of hits taken using Merge Sort : 534464
No of swaps taken using Merge Sort : 0
2023-03-12 22:02:17 INFO Benchmark_Timer - Begin run: Sorts array of 10000 elements with 1 runs
Time taken for 10000 elements of array using Quick Sort is : 190.0
No of compares taken using Quick Sort : 15910
No of hits taken using Quick Sort : 478127
No of swaps taken using Quick Sort : 76429
2023-03-12 22:02:18 INFO Benchmark_Timer - Begin run: Sorts array of 10000 elements with 1 runs
Time taken for 10000 elements of array using Heap Sort is : 1.0
No of compares taken using Heap Sort : 235668
No of hits taken using Heap Sort : 968932
No of swaps taken using Heap Sort : 124399
Process finished with exit code 0

Git Run TODO Problems Terminal Services Profiler Build Dependencies

Build completed successfully in 1 sec, 576 ms (a minute ago)

26:22 LF UTF-8 4 spaces main

INFO6205 src main java edu neu coe info6205 util NewSortBenchmark main

Project Pull Requests Commit Bookmarks Structure

java edu.neu.coe.info6205 bqs dynamicProgramming functions graphs greedy lab_1 life pq randomwalk reduction sort classic counting elementary hashCode linearithmic IntroSortTest

SortBenchmark.java QuickSort3WayTest.java NewSortBenchmark.java Benchmark_Timer.java Consumer.java Config.java StatPack.java

NewSortBenchmark.java

```
17 no usages ▲ Thikkavarapu Manikanta Reddy *
18 ▶ public class NewSortBenchmark {
19   ▲ Thikkavarapu Manikanta Reddy *
20   ▶   public static void main(String[] args) {
21     ▶     int n = 20000;
22     ▶     BaseHelper<Integer> h1 = new InstrumentedHelper<>( description: "test", config);
23     ▶     MergeSortBasic<Integer> merge = new MergeSortBasic<>(h1);
24     ▶     ...
25   }
26   ▶   // Merge Sort
27   ▶   final Config config = Config.setupConfig( instrumenting: "true", seed: "0", inversions: "1", cutoff: "1", interInversions: "" );
28   ▶   ...
29 }
30 }
```

Run: NewSortBenchmark

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
2023-03-12 22:04:44 INFO Benchmark_Timer - Begin run: Sorts array of 20000 elements with 1 runs
Time taken for 20000 elements of array using Merge Sort is : 7.0
No of compares taken using Merge Sort : 260798
No of hits taken using Merge Sort : 1148928
No of swaps taken using Merge Sort : 0
2023-03-12 22:04:46 INFO Benchmark_Timer - Begin run: Sorts array of 20000 elements with 1 runs
Time taken for 20000 elements of array using Quick Sort is : 656.0
No of compares taken using Quick Sort : 317482
No of hits taken using Quick Sort : 969130
No of swaps taken using Quick Sort : 156511
2023-03-12 22:04:49 INFO Benchmark_Timer - Begin run: Sorts array of 20000 elements with 1 runs
Time taken for 20000 elements of array using Heap Sort is : 4.0
No of compares taken using Heap Sort : 510631
No of hits taken using Heap Sort : 2094398
No of swaps taken using Heap Sort : 268284

Process finished with exit code 0
```

Git Run TODO Problems Terminal Services Profiler Build Dependencies

Build completed successfully in 1 sec, 510 ms (moments ago)

4:47 LF UTF-8 4 spaces main

INFO6205 src main java edu neu coe info6205 util NewSortBenchmark main

Project Pull Requests Commit Bookmarks Structure

java edu.neu.coe.info6205 bqs dynamicProgramming functions graphs greedy lab_1 life pq randomwalk reduction sort classic counting elementary hashCode linearithmic IntroSortTest

SortBenchmark.java QuickSort3WayTest.java NewSortBenchmark.java Benchmark_Timer.java Consumer.java Config.java StatPack.java

NewSortBenchmark.java

```
17 no usages ▲ Thikkavarapu Manikanta Reddy *
18 ▶ public class NewSortBenchmark {
19   ▲ Thikkavarapu Manikanta Reddy *
20   ▶   public static void main(String[] args) {
21     ▶     int n = 40000;
22     ▶     BaseHelper<Integer> h1 = new InstrumentedHelper<>( description: "test", config);
23     ▶     MergeSortBasic<Integer> merge = new MergeSortBasic<>(h1);
24     ▶     ...
25   }
26   ▶   // Merge Sort
27   ▶   final Config config = Config.setupConfig( instrumenting: "true", seed: "0", inversions: "1", cutoff: "1", interInversions: "" );
28   ▶   ...
29 }
30 }
```

Run: NewSortBenchmark

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
2023-03-12 22:08:49 INFO Benchmark_Timer - Begin run: Sorts array of 40000 elements with 1 runs
Time taken for 40000 elements of array using Merge Sort is : 11.0
No of compares taken using Merge Sort : 561836
No of hits taken using Merge Sort : 2457856
No of swaps taken using Merge Sort : 0
2023-03-12 22:08:52 INFO Benchmark_Timer - Begin run: Sorts array of 40000 elements with 1 runs
Time taken for 40000 elements of array using Quick Sort is : 2884.0
No of compares taken using Quick Sort : 725857
No of hits taken using Quick Sort : 2080922
No of swaps taken using Quick Sort : 3260087
2023-03-12 22:09:03 INFO Benchmark_Timer - Begin run: Sorts array of 40000 elements with 1 runs
Time taken for 40000 elements of array using Heap Sort is : 9.0
No of compares taken using Heap Sort : 1161996
No of hits taken using Heap Sort : 4511776
No of swaps taken using Heap Sort : 576946

Process finished with exit code 0
```

Git Run TODO Problems Terminal Services Profiler Build Dependencies

Build completed successfully in 1 sec, 544 ms (moments ago)

22:23 LF UTF-8 4 spaces main

INFO6205 src main java edu neu coe info6205 util NewSortBenchmark main

Project Pull Requests Commit Bookmarks Structure

java edu.neu.coe.info6205 bqs dynamicProgramming functions graphs greedy lab_1 life pq randomwalk reduction sort classic counting elementary hashCode linearithmic IntroSortTest

SortBenchmark.java QuickSort3WayTest.java NewSortBenchmark.java Benchmark_Timer.java Consumer.java Config.java StatPack.java

NewSortBenchmark.java

```
17 no usages ▲ Thikkavarapu Manikanta Reddy *
18 ▶ public class NewSortBenchmark {
19   ▲ Thikkavarapu Manikanta Reddy *
20   ▶   public static void main(String[] args) {
21     ▶     int n = 40000;
22     ▶     BaseHelper<Integer> h1 = new InstrumentedHelper<>( description: "test", config);
23     ▶     MergeSortBasic<Integer> merge = new MergeSortBasic<>(h1);
24     ▶     ...
25   }
26   ▶   // Merge Sort
27   ▶   final Config config = Config.setupConfig( instrumenting: "true", seed: "0", inversions: "1", cutoff: "1", interInversions: "" );
28   ▶   ...
29   ▶   MergeSortBasic<Integer> merge = new MergeSortBasic<>(h1);
30 }
```

Run: NewSortBenchmark

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
2023-03-12 22:08:49 INFO Benchmark_Timer - Begin run: Sorts array of 40000 elements with 1 runs
Time taken for 40000 elements of array using Merge Sort is : 11.0
No of compares taken using Merge Sort : 561836
No of hits taken using Merge Sort : 2457856
No of swaps taken using Merge Sort : 0
2023-03-12 22:08:52 INFO Benchmark_Timer - Begin run: Sorts array of 40000 elements with 1 runs
Time taken for 40000 elements of array using Quick Sort is : 2884.0
No of compares taken using Quick Sort : 725857
No of hits taken using Quick Sort : 2088922
No of swaps taken using Quick Sort : 326087
2023-03-12 22:09:03 INFO Benchmark_Timer - Begin run: Sorts array of 40000 elements with 1 runs
Time taken for 40000 elements of array using Heap Sort is : 9.0
No of compares taken using Heap Sort : 1101996
No of hits taken using Heap Sort : 4511776
No of swaps taken using Heap Sort : 576946

Process finished with exit code 0
```

Git Run TODO Problems Terminal Services Profiler Build Dependencies

Build completed successfully in 1 sec, 544 ms (2 minutes ago)

4:47 LF UTF-8 4 spaces main

INFO6205 src main java edu neu coe info6205 util NewSortBenchmark main

Project Pull Requests Commit Bookmarks Structure

java edu.neu.coe.info6205 bqs dynamicProgramming functions graphs greedy lab_1 life pq randomwalk reduction sort classic counting elementary hashCode linearithmic IntroSortTest

SortBenchmark.java QuickSort3WayTest.java NewSortBenchmark.java Benchmark_Timer.java Consumer.java Config.java StatPack.java

NewSortBenchmark.java

```
17 no usages ▲ Thikkavarapu Manikanta Reddy *
18 ▶ public class NewSortBenchmark {
19   ▲ Thikkavarapu Manikanta Reddy *
20   ▶   public static void main(String[] args) {
21     ▶     int n = 80000;
22     ▶     BaseHelper<Integer> h1 = new InstrumentedHelper<>( description: "test", config);
23     ▶     MergeSortBasic<Integer> merge = new MergeSortBasic<>(h1);
24     ▶     ...
25   }
26   ▶   // Merge Sort
27   ▶   final Config config = Config.setupConfig( instrumenting: "true", seed: "0", inversions: "1", cutoff: "1", interInversions: "" );
28   ▶   ...
29   ▶   MergeSortBasic<Integer> merge = new MergeSortBasic<>(h1);
30 }
```

Run: NewSortBenchmark

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
2023-03-12 22:11:25 INFO Benchmark_Timer - Begin run: Sorts array of 80000 elements with 1 runs
Time taken for 80000 elements of array using Merge Sort is : 25.0
No of compares taken using Merge Sort : 1203498
No of hits taken using Merge Sort : 5235732
No of swaps taken using Merge Sort : 0
2023-03-12 22:11:49 INFO Benchmark_Timer - Begin run: Sorts array of 80000 elements with 1 runs
Time taken for 80000 elements of array using Quick Sort is : 11129.0
No of compares taken using Quick Sort : 1711807
No of hits taken using Quick Sort : 4769825
No of swaps taken using Quick Sort : 738929
2023-03-12 22:12:45 INFO Benchmark_Timer - Begin run: Sorts array of 80000 elements with 1 runs
Time taken for 80000 elements of array using Heap Sort is : 28.0
No of compares taken using Heap Sort : 2363076
No of hits taken using Heap Sort : 9661028
No of swaps taken using Heap Sort : 1233722

Process finished with exit code 0
```

Git Run TODO Problems Terminal Services Profiler Build Dependencies

Build completed successfully in 2 sec, 344 ms (2 minutes ago)

22:18 LF UTF-8 4 spaces main

The screenshot shows an IDE interface with the following details:

- Project:** INFO6205
- File:** NewSortBenchmark.java
- Code Snippet:**

```
no usages ▲ Thikkavarapu Manikanta Reddy *
public class NewSortBenchmark {
    public static void main(String[] args) {
        int n = 160000;
        final Config config = Config.setupConfig( instrumenting: "true", seed: "0", inversions: "1", cutoff: "1", interInversions: "" );
        // Merge Sort
        BaseHelper<Integer> h1 = new InstrumentedHelper<>( description: "test", config );
        MergeSortBasic<Integer> merge = new MergeSortBasic<>(h1);
    }
}
```

- Run Output:**

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
2023-03-12 22:22:39 INFO Benchmark_Timer - Begin run: Sorts array of 160000 elements with 1 runs
Time taken for 160000 elements of array using Merge Sort is : 57.0
No of compares taken using Merge Sort : 2566815
No of hits taken using Merge Sort : 11111424
No of swaps taken using Merge Sort : 0
2023-03-12 22:23:31 INFO Benchmark_Timer - Begin run: Sorts array of 160000 elements with 1 runs
Time taken for 160000 elements of array using Quick Sort is : 74018.0
No of compares taken using Quick Sort : 3497600
No of hits taken using Quick Sort : 9667996
No of swaps taken using Quick Sort : 1491426
2023-03-12 22:28:15 INFO Benchmark_Timer - Begin run: Sorts array of 160000 elements with 1 runs
Time taken for 160000 elements of array using Heap Sort is : 43.0
No of compares taken using Heap Sort : 5045838
No of hits taken using Heap Sort : 20600116
No of swaps taken using Heap Sort : 2627110
Process finished with exit code 0
```

- Bottom Status:** Build completed successfully in 1 sec, 472 ms (8 minutes ago)

The screenshot shows an IDE interface with the following details:

- Project:** INFO6205
- File:** NewSortBenchmark.java
- Code Snippet:**

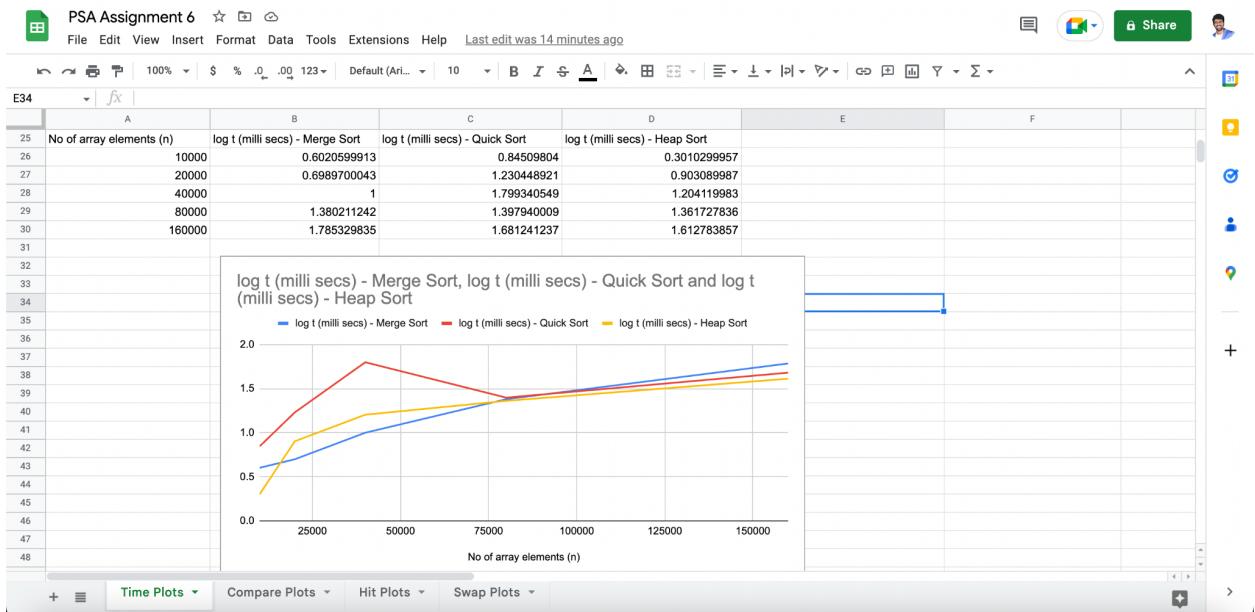
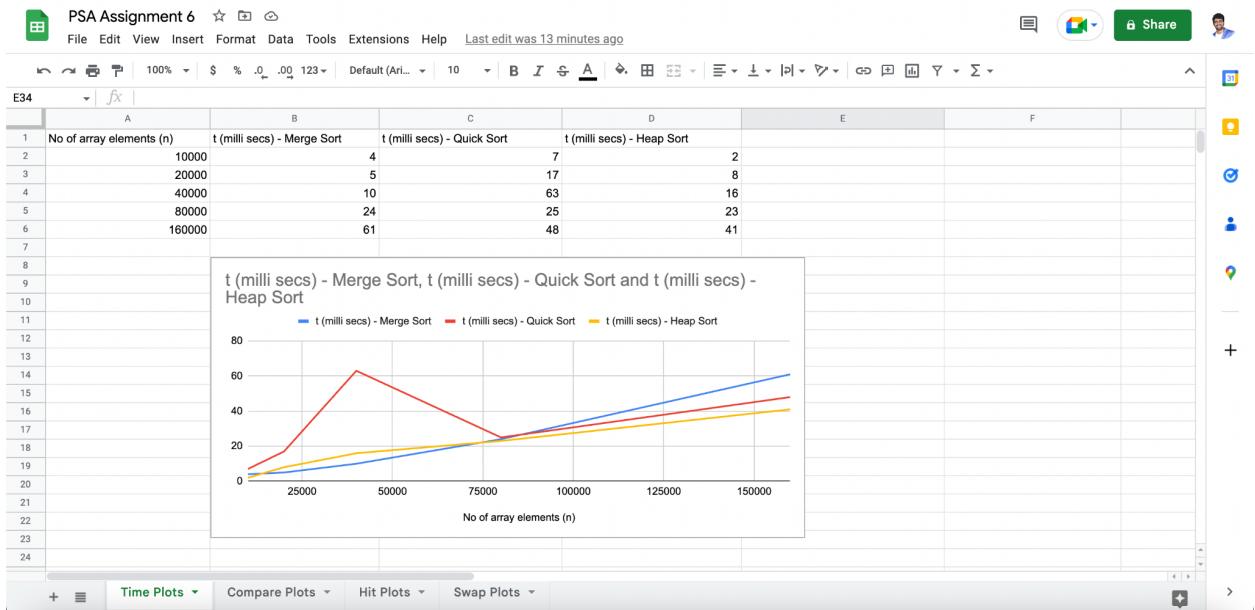
```
no usages ▲ Thikkavarapu Manikanta Reddy *
public class NewSortBenchmark {
    public static void main(String[] args) {
        int n = 160000;
        final Config config = Config.setupConfig( instrumenting: "true", seed: "0", inversions: "1", cutoff: "1", interInversions: "" );
        // Merge Sort
        BaseHelper<Integer> h1 = new InstrumentedHelper<>( description: "test", config );
        MergeSortBasic<Integer> merge = new MergeSortBasic<>(h1);
    }
}
```

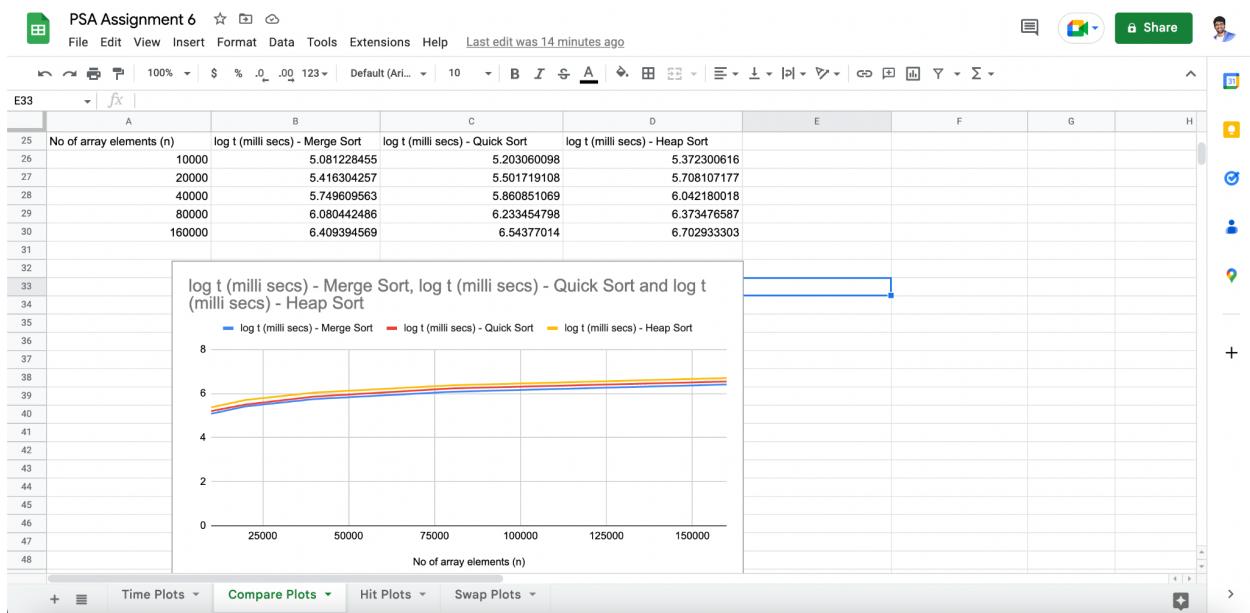
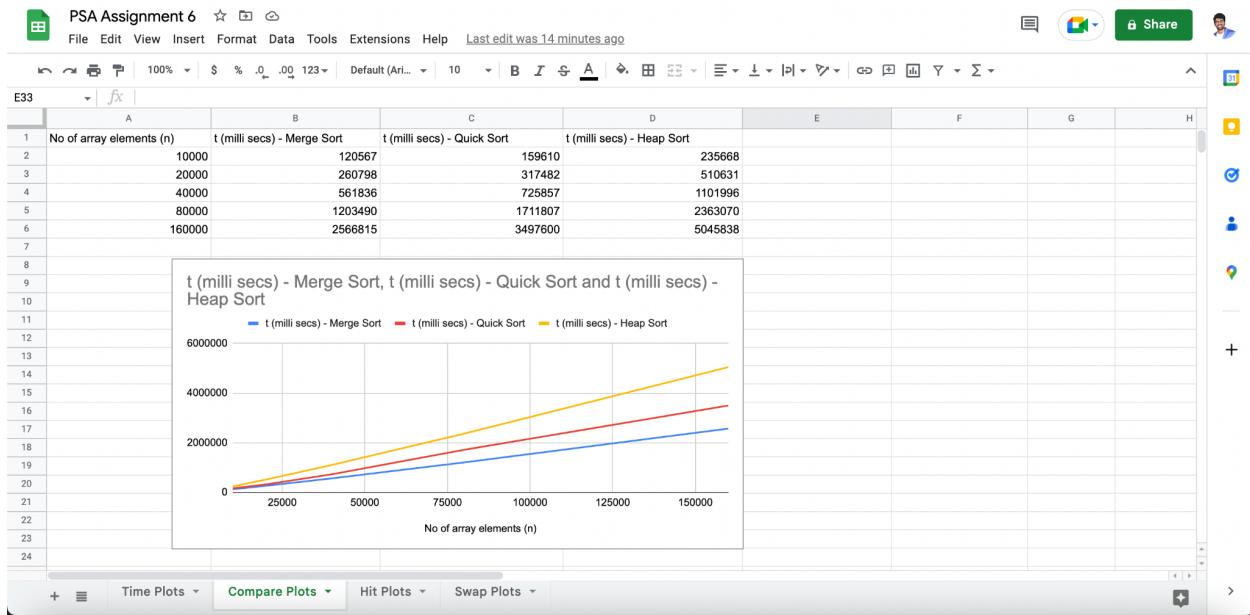
- Run Output:**

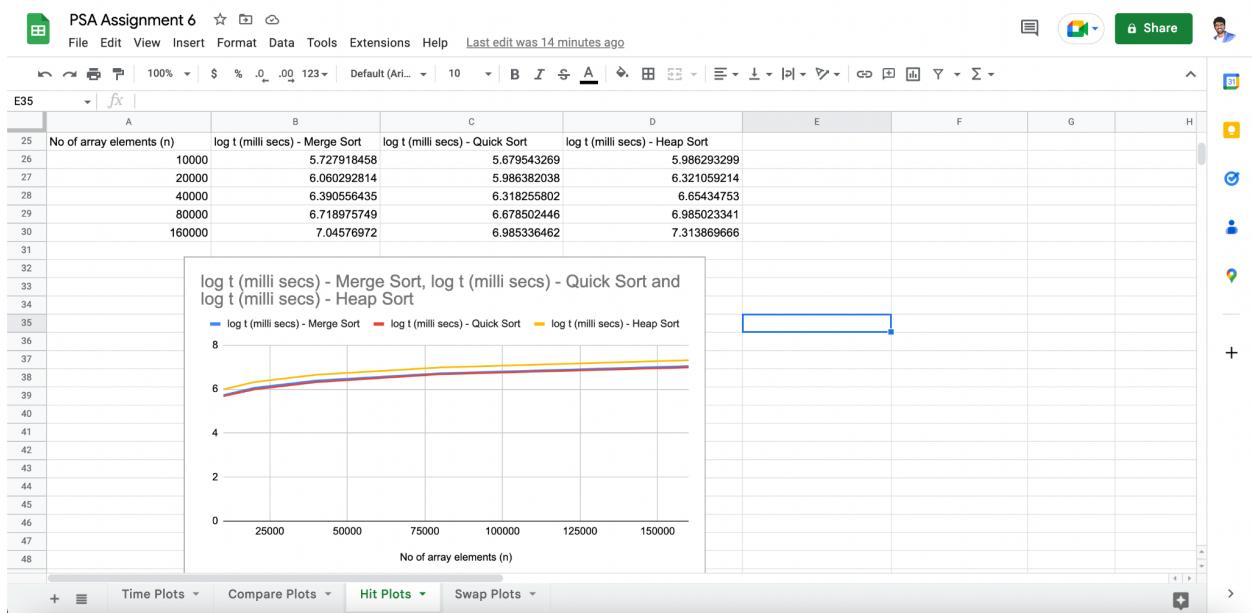
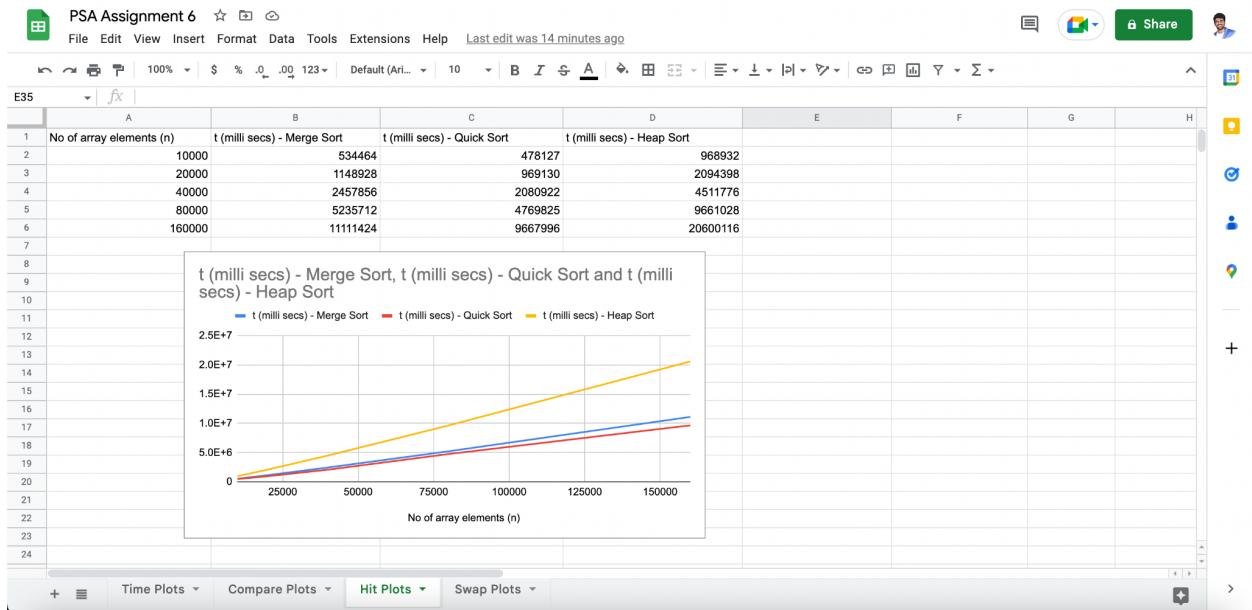
```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/Contents/Home/bin/java ...
2023-03-12 22:22:39 INFO Benchmark_Timer - Begin run: Sorts array of 160000 elements with 1 runs
Time taken for 160000 elements of array using Merge Sort is : 57.0
No of compares taken using Merge Sort : 2566815
No of hits taken using Merge Sort : 11111424
No of swaps taken using Merge Sort : 0
2023-03-12 22:23:31 INFO Benchmark_Timer - Begin run: Sorts array of 160000 elements with 1 runs
Time taken for 160000 elements of array using Quick Sort is : 74018.0
No of compares taken using Quick Sort : 3497600
No of hits taken using Quick Sort : 9667996
No of swaps taken using Quick Sort : 1491426
2023-03-12 22:28:15 INFO Benchmark_Timer - Begin run: Sorts array of 160000 elements with 1 runs
Time taken for 160000 elements of array using Heap Sort is : 43.0
No of compares taken using Heap Sort : 5045838
No of hits taken using Heap Sort : 20600116
No of swaps taken using Heap Sort : 2627110
Process finished with exit code 0
```

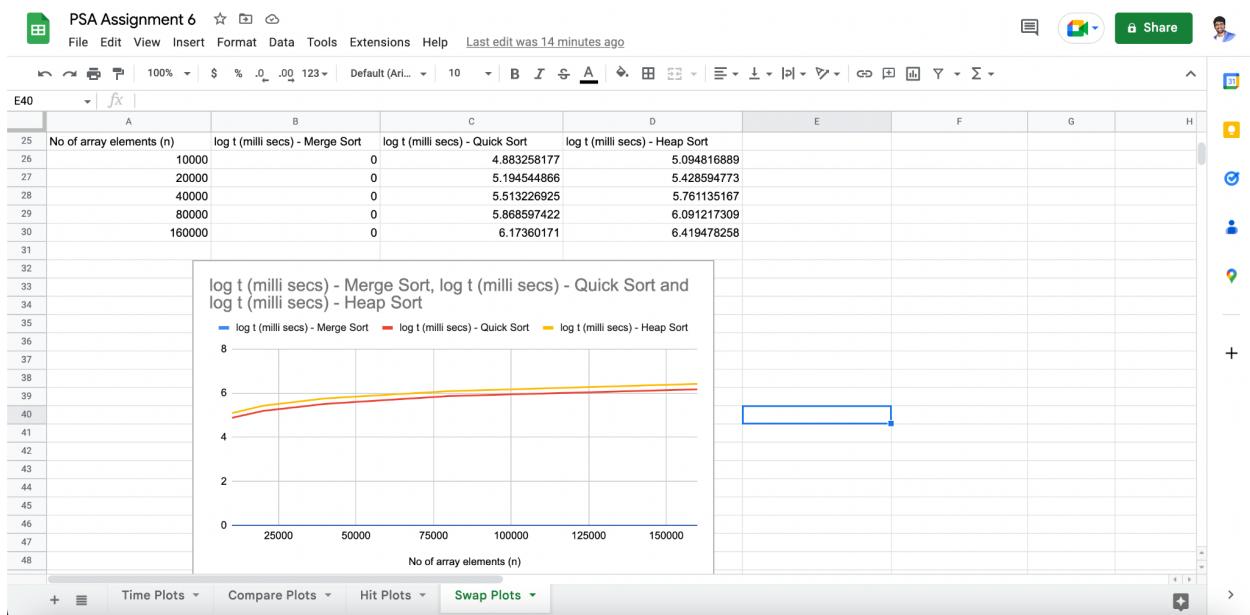
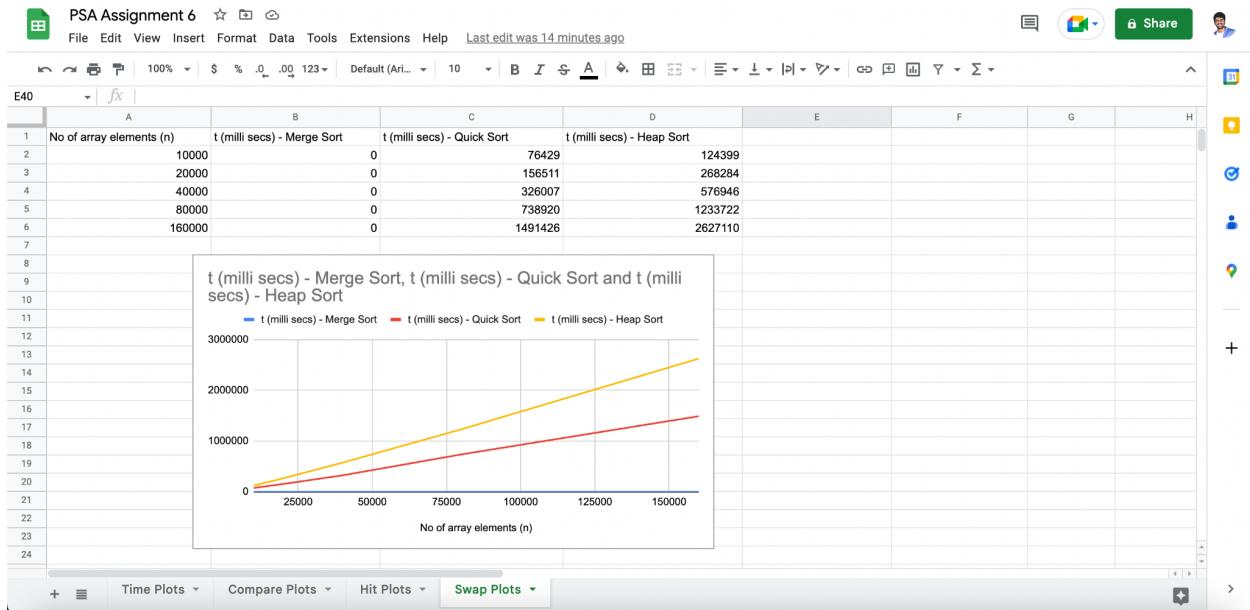
- Bottom Status:** Build completed successfully in 1 sec, 472 ms (20 minutes ago)

Graph plots:









Conclusion:

Log(n) vs Log(t) graphs for time, compare, hit, and swap look similar and variations are horizontally flat, linear, and constant.