

**ANURAG PARLA (002127710)**

## **Program Structures & Algorithms**

### **Fall 2021**

#### **Assignment No. 02**

##### **Task:**

- 1) To implement 3 methods in a class called Timer and verify the implementation by running unit tests in BenchmarkTest and TimerTest.
- 2) Implement InsertionSort by simply looking up the insertion code used by Arrays.sort and verify the implementation by running unit tests in InsertionSortTest.
- 3) Implement a main program to run the following benchmarks: measure the running times of this sort using the following four different initial array ordering situations:
  - i) Random
  - ii) Ordered
  - iii) Reverse Ordered
  - iv) Partially Ordered

Also use the doubling method for choosing 'n' and test for at least five values of 'n'.

### **Relationship Conclusion for Task 3:**

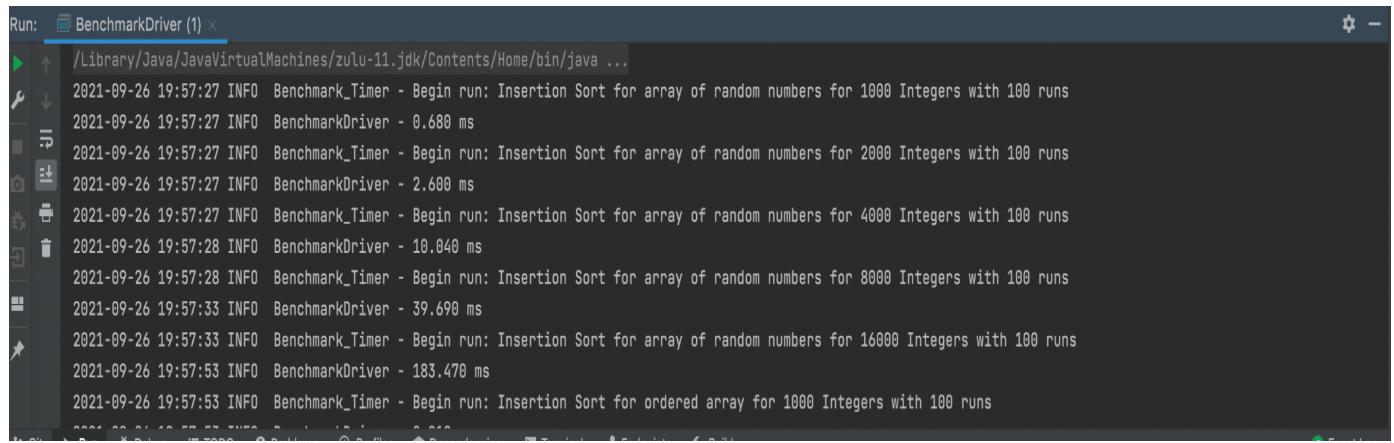
From the observations of the graph i.e. relationship between N and T for different ordered arrays, following conclusions are drawn:

1. Worst case scenario corresponds to the reverse ordered array elements. Hence the time complexity is  $O(N^2)$  which is a quadratic growth graph.
2. Average case scenario corresponds to the partially ordered array elements. Hence the time complexity is  $O(N \log(N))$  which is a linearithmic graph.
3. Best case scenario corresponds to the ordered array elements. Hence the time complexity is  $O(N)$  which is a constant time graph.

## Evidence to support the conclusion:

### Screenshot of the code output in the terminal:

#### i) Random:



```
Run: BenchmarkDriver (1) ×
/Library/Java/JavaVirtualMachines/zulu-11.jdk/Contents/Home/bin/java ...
2021-09-26 19:57:27 INFO Benchmark_Timer - Begin run: Insertion Sort for array of random numbers for 1000 Integers with 100 runs
2021-09-26 19:57:27 INFO BenchmarkDriver - 0.680 ms
2021-09-26 19:57:27 INFO Benchmark_Timer - Begin run: Insertion Sort for array of random numbers for 2000 Integers with 100 runs
2021-09-26 19:57:27 INFO BenchmarkDriver - 2.600 ms
2021-09-26 19:57:27 INFO Benchmark_Timer - Begin run: Insertion Sort for array of random numbers for 4000 Integers with 100 runs
2021-09-26 19:57:28 INFO BenchmarkDriver - 10.040 ms
2021-09-26 19:57:28 INFO Benchmark_Timer - Begin run: Insertion Sort for array of random numbers for 8000 Integers with 100 runs
2021-09-26 19:57:33 INFO BenchmarkDriver - 39.690 ms
2021-09-26 19:57:33 INFO Benchmark_Timer - Begin run: Insertion Sort for array of random numbers for 16000 Integers with 100 runs
2021-09-26 19:57:53 INFO BenchmarkDriver - 183.470 ms
2021-09-26 19:57:53 INFO Benchmark_Timer - Begin run: Insertion Sort for ordered array for 1000 Integers with 100 runs
2021-09-26 19:57:53 INFO BenchmarkDriver - 0.010 ms
```

#### ii) Ordered:



```
2021-09-26 19:57:53 INFO Benchmark_Timer - Begin run: Insertion Sort for ordered array for 1000 Integers with 100 runs
2021-09-26 19:57:53 INFO BenchmarkDriver - 0.010 ms
2021-09-26 19:57:53 INFO Benchmark_Timer - Begin run: Insertion Sort for ordered array for 2000 Integers with 100 runs
2021-09-26 19:57:53 INFO BenchmarkDriver - 0.010 ms
2021-09-26 19:57:53 INFO Benchmark_Timer - Begin run: Insertion Sort for ordered array for 4000 Integers with 100 runs
2021-09-26 19:57:53 INFO BenchmarkDriver - 0.010 ms
2021-09-26 19:57:53 INFO Benchmark_Timer - Begin run: Insertion Sort for ordered array for 8000 Integers with 100 runs
2021-09-26 19:57:53 INFO BenchmarkDriver - 0.030 ms
2021-09-26 19:57:53 INFO Benchmark_Timer - Begin run: Insertion Sort for ordered array for 16000 Integers with 100 runs
2021-09-26 19:57:53 INFO BenchmarkDriver - 0.070 ms
```

### **iii) Reverse Ordered:**

```
2021-09-26 19:57:53 INFO Benchmark_Timer - Begin run: Insertion Sort for reverse ordered array for 1000 Integers with 100 runs
2021-09-26 19:57:53 INFO BenchmarkDriver - 1.220 ms
2021-09-26 19:57:53 INFO Benchmark_Timer - Begin run: Insertion Sort for reverse ordered array for 2000 Integers with 100 runs
2021-09-26 19:57:53 INFO BenchmarkDriver - 4.940 ms
2021-09-26 19:57:53 INFO Benchmark_Timer - Begin run: Insertion Sort for reverse ordered array for 4000 Integers with 100 runs
2021-09-26 19:57:56 INFO BenchmarkDriver - 19.370 ms
2021-09-26 19:57:56 INFO Benchmark_Timer - Begin run: Insertion Sort for reverse ordered array for 8000 Integers with 100 runs
2021-09-26 19:58:05 INFO BenchmarkDriver - 80.970 ms
2021-09-26 19:58:05 INFO Benchmark_Timer - Begin run: Insertion Sort for reverse ordered array for 16000 Integers with 100 runs
2021-09-26 19:58:40 INFO BenchmarkDriver - 321.520 ms
```

#### iv) Partially Ordered:



The screenshot shows a terminal window titled "BenchmarkDriver (1)". The output of the program is displayed, showing the time taken for insertion sort on partially ordered arrays of size N=1000, 2000, 4000, 8000, and 16000. The times are 0.180 ms, 0.650 ms, 2.520 ms, 10.440 ms, and 44.720 ms respectively. The terminal interface includes standard navigation keys like up, down, left, right, and delete.

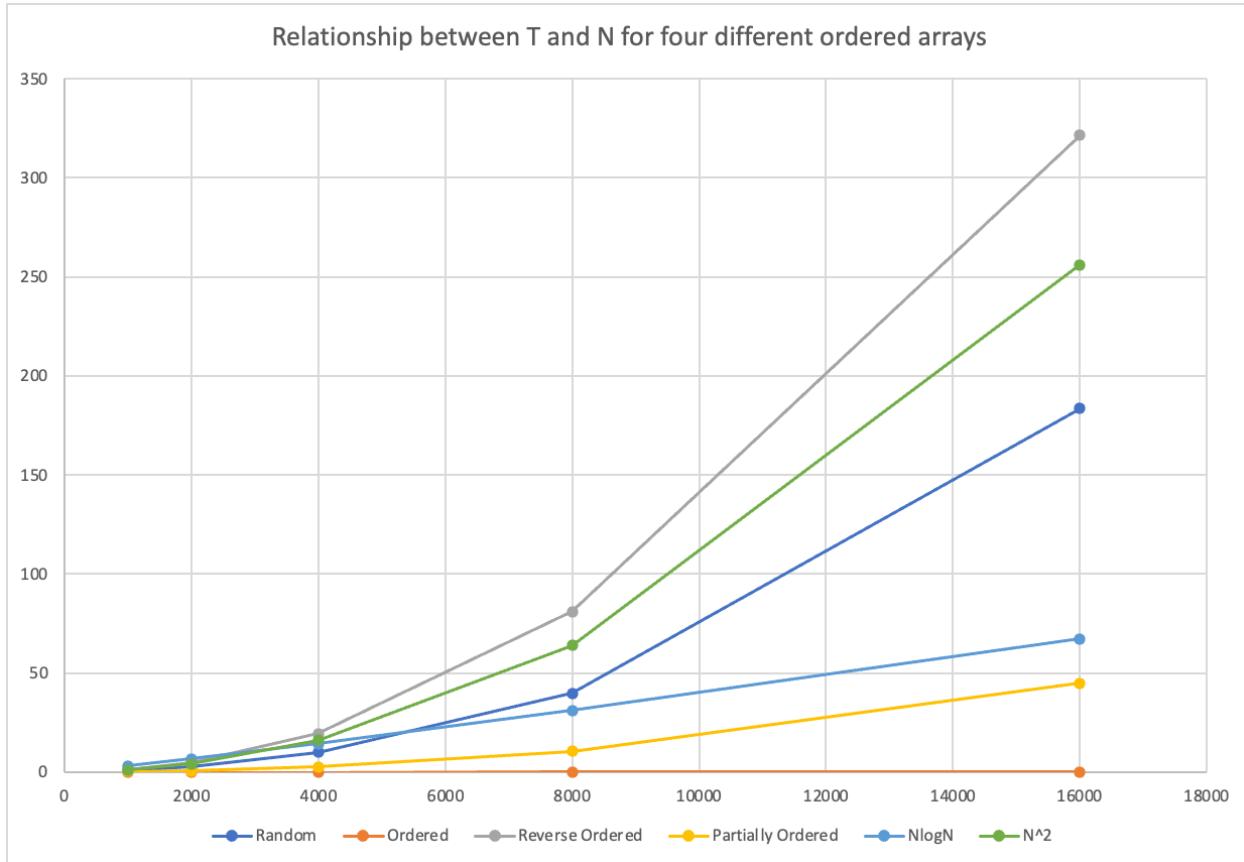
```
Run: BenchmarkDriver (1) ×
▶ ⌂ 2021-09-26 19:58:40 INFO Benchmark_Timer - Begin run: Insertion Sort for partially ordered array for 1000 Integers with 100 runs
▶ ⌂ 2021-09-26 19:58:40 INFO BenchmarkDriver - 0.180 ms
▶ ⌂ 2021-09-26 19:58:40 INFO Benchmark_Timer - Begin run: Insertion Sort for partially ordered array for 2000 Integers with 100 runs
▶ ⌂ 2021-09-26 19:58:40 INFO BenchmarkDriver - 0.650 ms
▶ ⌂ 2021-09-26 19:58:40 INFO Benchmark_Timer - Begin run: Insertion Sort for partially ordered array for 4000 Integers with 100 runs
▶ ⌂ 2021-09-26 19:58:40 INFO BenchmarkDriver - 2.520 ms
▶ ⌂ 2021-09-26 19:58:40 INFO Benchmark_Timer - Begin run: Insertion Sort for partially ordered array for 8000 Integers with 100 runs
▶ ⌂ 2021-09-26 19:58:41 INFO BenchmarkDriver - 10.440 ms
▶ ⌂ 2021-09-26 19:58:41 INFO Benchmark_Timer - Begin run: Insertion Sort for partially ordered array for 16000 Integers with 100 runs
▶ ⌂ 2021-09-26 19:58:46 INFO BenchmarkDriver - 44.720 ms

Process finished with exit code 0
```

**Table consisting of N and T values for the four different types of ordered arrays:**

N	Random (ms)	Ordered( ms)	Reverse Ordered( ms)	Partially Ordered (ms)	Nlog(N)	$N^2$
1000	0.68	0.01	1.22	0.18	3	1
2000	2.6	0.01	4.94	0.65	6.602	4
4000	10.04	0.01	19.37	2.52	14.408	16
8000	39.69	0.03	80.97	10.44	31.225	64
16000	183.47	0.07	321.52	44.72	67.266	256

## Graphical relationship between the time T and size of the array N for different types of ordered arrays:



## Unit tests result(Snapshot of successful unit test run):

### i) Timer test:

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project:** INFO6205 > src > test > java > edu > neu > coe > info6205 > util > TimerTest
- Code Editor:** The file TimerTest.java is open, showing test cases for a Timer class. A code completion tooltip is visible over the word 'Millisecs'.
- Run Tab:** Shows the results of a run named "BenchmarkDriver". It indicates "Tests passed: 10 of 10 tests ~ 2 sec 247 ms".
- Test Results:** A detailed list of test results is shown, including:
  - testPauseAndLapResume: 207 ms
  - testPauseAndLapResume': 312 ms
  - testLap: 210 ms
  - testPause: 210 ms
  - testStop: 105 ms
  - testMillisecs: 106 ms
  - testRepeat1: 123 ms
  - testRepeat2: 247 ms
  - testRepeat3: 622 ms
  - testPauseAndLap: 105 ms
- Bottom Status:** "Tests passed: 10 (a minute ago)"

## ii) Benchmark test:

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project:** INFO6205
- File:** BenchmarkTest.java
- Code:**

```
INFO6205 - BenchmarkTest.java
INFO6205 > src > test > java > edu > neu > coe > info6205 > util > BenchmarkTest > testWaitPeriods
1 package edu.neu.coe.info6205.util;
2
3 import ...
4
5 @SuppressWarnings("ALL")
6 public class BenchmarkTest {
7
8     int pre = 0;
9     int run = 0;
10    int post = 0;
11
12    @Test // Slow
13    public void testWaitPeriods() throws Exception {
14        int nRuns = 2;
15
16        int sum = 0;
17        for (int i = 0; i < nRuns; i++) {
18            sum += run;
19        }
20    }
21}
```
- Run:** BenchmarkDriver
- Output:**

```
Tests passed: 2 of 2 tests - 1sec 429 ms
BenchmarkTest (edu.neu.coe.info6205.util.BenchmarkTest)
  ✓ testWaitPeriods 1sec 429 ms
  ✓ getWarmupRuns 0ms
Process finished with exit code 0
```
- Status Bar:** Tests passed: 2 (moments ago)
- System Tray:** Shows various Mac OS X application icons.

### **iii) Insertion sort test:**

The screenshot shows the IntelliJ IDEA interface with the following details:

- Title Bar:** INFO6205 – InsertionSortTest.java
- Project Bar:** INFO6205, src, test, java, edu, neu, coe, info6205, sort, elementary, InsertionSortTest, sort0.
- Code Editor:** The code for `InsertionSortTest.java` is displayed. It contains a single test method `sort0()` that creates a list of integers [1, 2, 3, 4] and then converts it to an array. It uses a helper factory to create a `Helper` object and a `PrivateMethodTester` to invoke the private `statPack` method.
- Run Tab:** The `BenchmarkDriver` configuration is selected, and the output shows the test results:
  - Tests passed:** 6 of 6 tests – 117 ms
  - Test Details:** `InsertionSortTest (edu.neu.co` took 117 ms.
    - `testMutatingInsertionSort` took 106 ms.
    - `sort0` took 5 ms.
    - `sort1` took 0 ms.
    - `sort2` took 3 ms.
    - `sort3` took 2 ms.
    - `testStaticInsertionSort` took 1 ms.
  - Logs:** Detailed logs for each test case are shown, including DEBUG messages about config settings like seed, inversions, cutoff, and compares.
- Bottom Bar:** Shows various Mac OS X application icons (Calculator, Mail, Finder, etc.) and the IntelliJ IDEA status bar indicating the current time (Sep 26, 2021, 3:24 PM), file encoding (UTF-8), and code style (4 spaces).

#### iv) Insertion sort MSDT test:

The screenshot shows an IDE interface with the following details:

- Project:** INFO6205
- File:** InsertionSortMSDTest.java
- Code:** The code defines two test methods: `sort0()` and `sort1()`.
  - `sort0()` takes a string "she sells seashells by the seashore the shells she sells are surely seashells" and splits it into ["she", "sells", "seashells", "by", "the", "seashore", "shells", "she", "sells", "are", "surely", "seashells"].
  - `sort1()` takes the same string and splits it into ["seashells", "seashells", "seashore", "sells", "sells", "she", "she", "shells", "surely"].
- Run Output:** The run output shows 4 tests passed in 3ms.
  - sort1: 3ms
  - sort0: 0ms
  - sort1: 0ms
  - sort2: 0msProcess finished with exit code 0.
- Status Bar:** Tests passed: 4 (moments ago)
- Bottom Right:** Event Log icon

## v) Insertion sort opt test:

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Bar:** Shows the project structure with files like InsertionSort.java, InsertionSortTest.java, BenchmarkDriver.java, InsertionSortMSDTest.java, InsertionSortOptTest.java, GenericSort.java, BaseHelper.java, Benchmarks.java, and Helper.java.
- Code Editor:** Displays the Java code for `InsertionSortOptTest`. The code includes a test method `sort0()` that creates a list of integers [1, 2, 3, 4] and sorts it using the insertion sort algorithm. It also includes setup and helper configurations.
- Run Tab:** Shows the output of the test run. It indicates "Tests passed: 4 of 4 tests - 267 ms". Below this, a detailed log of the test execution is shown:

  - testMutatingInsertionSort:** Took 100ms.
    - sort0:** Took 6ms.
    - sort1:** Took 0ms.
    - sort2:** Took 161ms.
  - Log output:

```
2021-09-26 22:29:46 DEBUG Config - Config.get(helper, instrument) = true
2021-09-26 22:29:46 DEBUG Config - Config.get(helper, seed) = 0
2021-09-26 22:29:46 DEBUG Config - Config.get(instrumenting, copies) = true
2021-09-26 22:29:46 DEBUG Config - Config.get(instrumenting, swaps) = true
2021-09-26 22:29:46 DEBUG Config - Config.get(instrumenting, compares) = true
2021-09-26 22:29:46 DEBUG Config - Config.get(instrumenting, inversions) = 1
2021-09-26 22:29:46 DEBUG Config - Config.get(instrumenting, fixes) = true
2021-09-26 22:29:46 DEBUG Config - Config.get(instrumenting, hits) = true
2021-09-26 22:29:46 DEBUG Config - Config.get(helper, cutoff) =
Helper for InsertionSortOpt with 4 elements
```

- Bottom Status Bar:** Shows the time as 19:14, file type as LF, encoding as UTF-8, 4 spaces, Fall2021, and an event log icon.

## v) Insertion sort opt test:

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Bar:** Shows the project structure with files like InsertionSort.java, InsertionSortTest.java, BenchmarkDriver.java, InsertionSortMSDTest.java, InsertionSortOptTest.java, GenericSort.java, BaseHelper.java, Benchmarks.java, and Helper.java.
- Code Editor:** Displays the Java code for `InsertionSortOptTest`. The code includes a test method `sort0()` that creates a list of integers [1, 2, 3, 4] and sorts it using the insertion sort algorithm. It also includes setup and helper configurations.
- Run Tab:** Shows the output of the test run. It indicates "Tests passed: 4 of 4 tests - 267 ms". Below this, a detailed log of the test execution is shown, including the time taken for each test case and the corresponding log output.
- Bottom Status Bar:** Shows the current time (19:14), file type (LF), encoding (UTF-8), and workspace name (Fall2021).

```
INFO6205 src test java edu neu coe Info6205 sort elementary InsertionSortOptTest
1 .../
4 package edu.neu.coe.info6205.sort.elementary;
6
7 import ...
17
18 /**
19  * public class InsertionSortOptTest {
20
21     @Test
22     public void sort0() throws Exception {
23         final List<Integer> list = new ArrayList<>();
24         list.add(1);
25         list.add(2);
26         list.add(3);
27         list.add(4);
28         Integer[] xs = list.toArray(new Integer[0]);
29         final Config config = ConfigTest.setupConfig(instrumenting: "true", seed: "0", inversions: "1", cutoff: "", interimInversions: "");
30         Helper<Integer> helper = HelperFactory.create(description: "InsertionSortOpt", list.size(), config);
31         helper.init(list.size());
32         final PrivateMethodTester privateMethodTester = new PrivateMethodTester(helper);
33         final StatPack statPack = (StatPack) privateMethodTester.invokePrivate(name: "getStatPack");
34         SortWithHelper<Integer> sorter = new InsertionSortOpt<>(helper);
Run:  InsertionSortOptTest
  Tests passed: 4 of 4 tests - 267 ms
  InsertionSortOptTest(edu.neu 267 ms) /Library/Java/JavaVirtualMachines/zulu-11.jdk/Contents/Home/bin/java ...
    testMutatingInsertionSort 100ms
      2021-09-26 22:29:46 DEBUG Config - Config.get(helper, instrument) = true
      2021-09-26 22:29:46 DEBUG Config - Config.get(helper, seed) = 0
      2021-09-26 22:29:46 DEBUG Config - Config.get(instrumenting, copies) = true
      2021-09-26 22:29:46 DEBUG Config - Config.get(instrumenting, swaps) = true
      2021-09-26 22:29:46 DEBUG Config - Config.get(instrumenting, compares) = true
      2021-09-26 22:29:46 DEBUG Config - Config.get(instrumenting, inversions) = 1
      2021-09-26 22:29:46 DEBUG Config - Config.get(instrumenting, fixes) = true
      2021-09-26 22:29:46 DEBUG Config - Config.get(instrumenting, hits) = true
      2021-09-26 22:29:46 DEBUG Config - Config.get(helper, cutoff) =
      Helper for InsertionSortOpt with 4 elements
    sort0 6ms
    sort1 0ms
    sort2 161ms
  Favorites
  Structure
  Git
  Run
  Debug
  TODO
  Problems
  Profiler
  Dependencies
  Terminal
  Endpoints
  Build
  Tests passed: 4 (moments ago)
  19:14 LF UTF-8 4 spaces Fall2021 Event Log
```