

INFO 6205

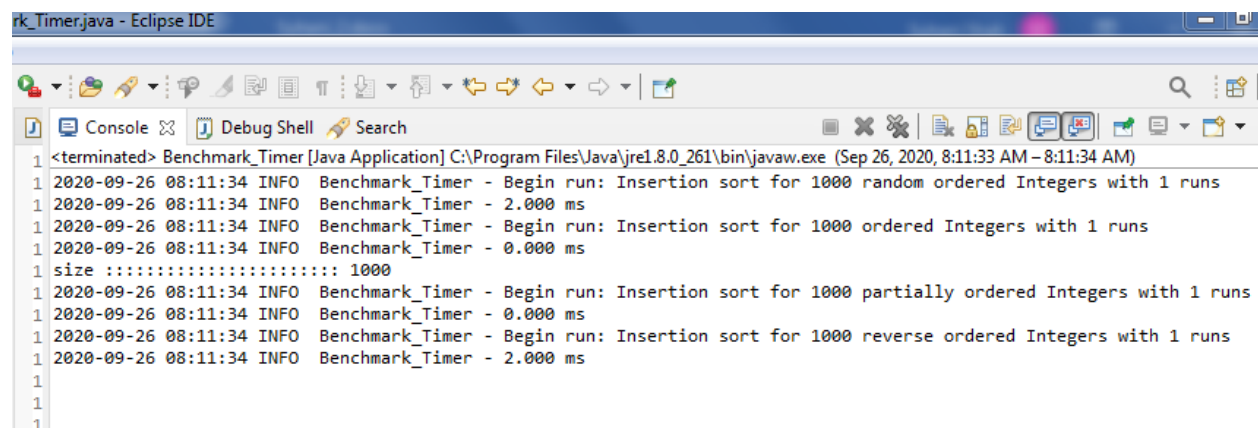
Program Structures & Algorithms

Fall 2020

Assignment No.2

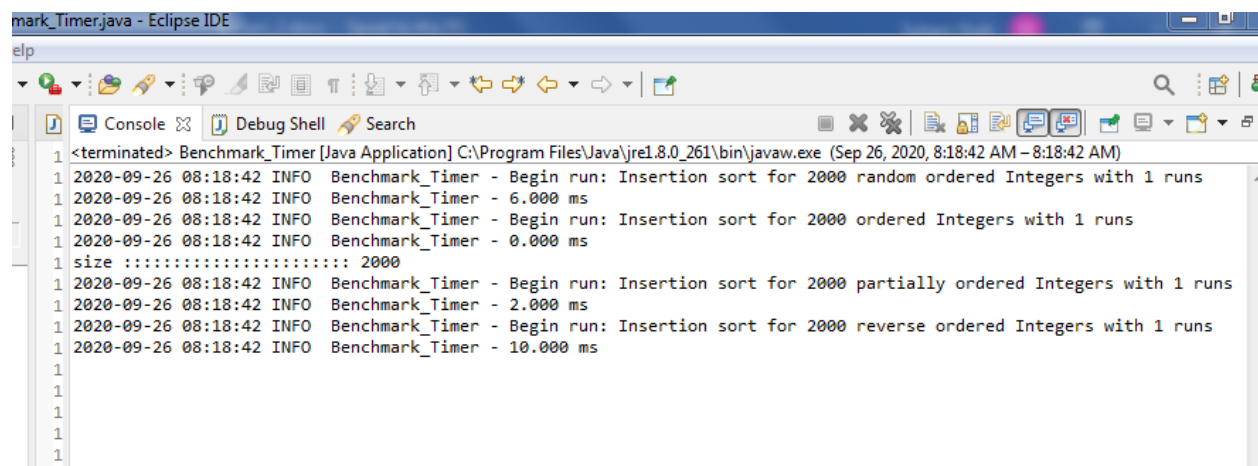
Insertion sort (PART2 and PART3)

1. Console output for 1000 elements



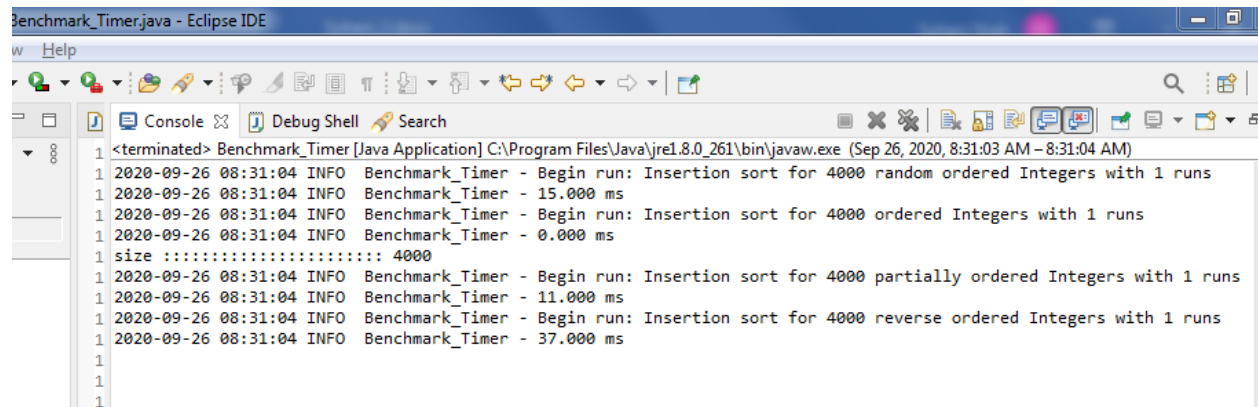
```
rk_Timer.java - Eclipse IDE
1 <terminated> Benchmark_Timer [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Sep 26, 2020, 8:11:33 AM - 8:11:34 AM)
1 2020-09-26 08:11:34 INFO Benchmark_Timer - Begin run: Insertion sort for 1000 random ordered Integers with 1 runs
1 2020-09-26 08:11:34 INFO Benchmark_Timer - 2.000 ms
1 2020-09-26 08:11:34 INFO Benchmark_Timer - Begin run: Insertion sort for 1000 ordered Integers with 1 runs
1 2020-09-26 08:11:34 INFO Benchmark_Timer - 0.000 ms
1 size :::::::::::::::::::: 1000
1 2020-09-26 08:11:34 INFO Benchmark_Timer - Begin run: Insertion sort for 1000 partially ordered Integers with 1 runs
1 2020-09-26 08:11:34 INFO Benchmark_Timer - 0.000 ms
1 2020-09-26 08:11:34 INFO Benchmark_Timer - Begin run: Insertion sort for 1000 reverse ordered Integers with 1 runs
1 2020-09-26 08:11:34 INFO Benchmark_Timer - 2.000 ms
1
1
1
1
```

2. Console output for 2000 elements



```
mark_Timer.java - Eclipse IDE
elp
1 <terminated> Benchmark_Timer [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Sep 26, 2020, 8:18:42 AM - 8:18:42 AM)
1 2020-09-26 08:18:42 INFO Benchmark_Timer - Begin run: Insertion sort for 2000 random ordered Integers with 1 runs
1 2020-09-26 08:18:42 INFO Benchmark_Timer - 6.000 ms
1 2020-09-26 08:18:42 INFO Benchmark_Timer - Begin run: Insertion sort for 2000 ordered Integers with 1 runs
1 2020-09-26 08:18:42 INFO Benchmark_Timer - 0.000 ms
1 size :::::::::::::::::::: 2000
1 2020-09-26 08:18:42 INFO Benchmark_Timer - Begin run: Insertion sort for 2000 partially ordered Integers with 1 runs
1 2020-09-26 08:18:42 INFO Benchmark_Timer - 2.000 ms
1 2020-09-26 08:18:42 INFO Benchmark_Timer - Begin run: Insertion sort for 2000 reverse ordered Integers with 1 runs
1 2020-09-26 08:18:42 INFO Benchmark_Timer - 10.000 ms
1
1
1
1
1
1
1
1
```

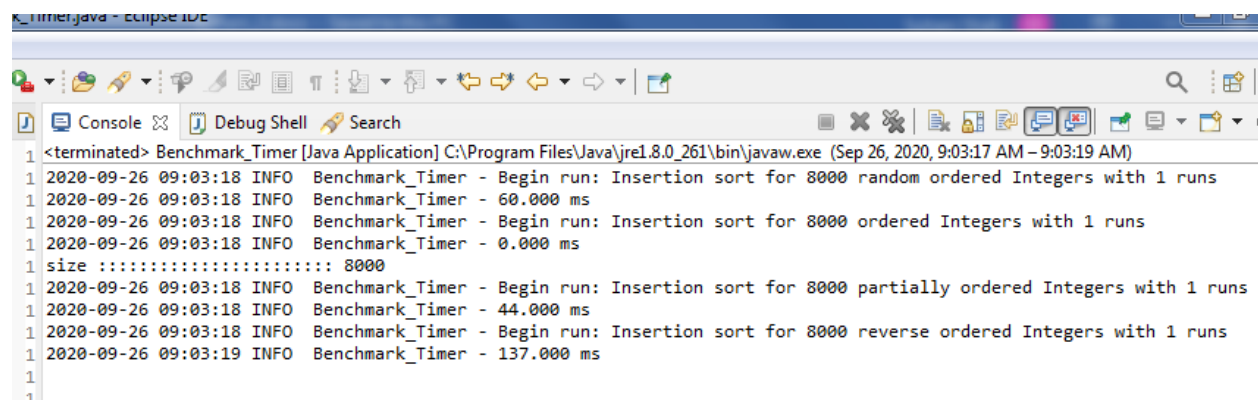
3. Console output for 4000 elements



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output displays the execution of the 'Benchmark_Timer' Java application. It shows four runs of insertion sort for 4000 elements: random, ordered, partially ordered, and reverse ordered. The times are 15.000 ms, 0.000 ms, 11.000 ms, and 37.000 ms respectively. The size of the array is confirmed as 4000.

```
<terminated> Benchmark_Timer [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Sep 26, 2020, 8:31:03 AM - 8:31:04 AM)
1 2020-09-26 08:31:04 INFO Benchmark_Timer - Begin run: Insertion sort for 4000 random ordered Integers with 1 runs
1 2020-09-26 08:31:04 INFO Benchmark_Timer - 15.000 ms
1 2020-09-26 08:31:04 INFO Benchmark_Timer - Begin run: Insertion sort for 4000 ordered Integers with 1 runs
1 2020-09-26 08:31:04 INFO Benchmark_Timer - 0.000 ms
1 size :::::::::::::::::::: 4000
1 2020-09-26 08:31:04 INFO Benchmark_Timer - Begin run: Insertion sort for 4000 partially ordered Integers with 1 runs
1 2020-09-26 08:31:04 INFO Benchmark_Timer - 11.000 ms
1 2020-09-26 08:31:04 INFO Benchmark_Timer - Begin run: Insertion sort for 4000 reverse ordered Integers with 1 runs
1 2020-09-26 08:31:04 INFO Benchmark_Timer - 37.000 ms
1
1
1
```

4. Console output for 8000 elements



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output displays the execution of the 'Benchmark_Timer' Java application for 8000 elements. It shows four runs of insertion sort: random, ordered, partially ordered, and reverse ordered. The times are 60.000 ms, 0.000 ms, 44.000 ms, and 137.000 ms respectively. The size of the array is confirmed as 8000.

```
<terminated> Benchmark_Timer [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (Sep 26, 2020, 9:03:17 AM - 9:03:19 AM)
1 2020-09-26 09:03:18 INFO Benchmark_Timer - Begin run: Insertion sort for 8000 random ordered Integers with 1 runs
1 2020-09-26 09:03:18 INFO Benchmark_Timer - 60.000 ms
1 2020-09-26 09:03:18 INFO Benchmark_Timer - Begin run: Insertion sort for 8000 ordered Integers with 1 runs
1 2020-09-26 09:03:18 INFO Benchmark_Timer - 0.000 ms
1 size :::::::::::::::::::: 8000
1 2020-09-26 09:03:18 INFO Benchmark_Timer - Begin run: Insertion sort for 8000 partially ordered Integers with 1 runs
1 2020-09-26 09:03:18 INFO Benchmark_Timer - 44.000 ms
1 2020-09-26 09:03:18 INFO Benchmark_Timer - Begin run: Insertion sort for 8000 reverse ordered Integers with 1 runs
1 2020-09-26 09:03:19 INFO Benchmark_Timer - 137.000 ms
1
1
1
```

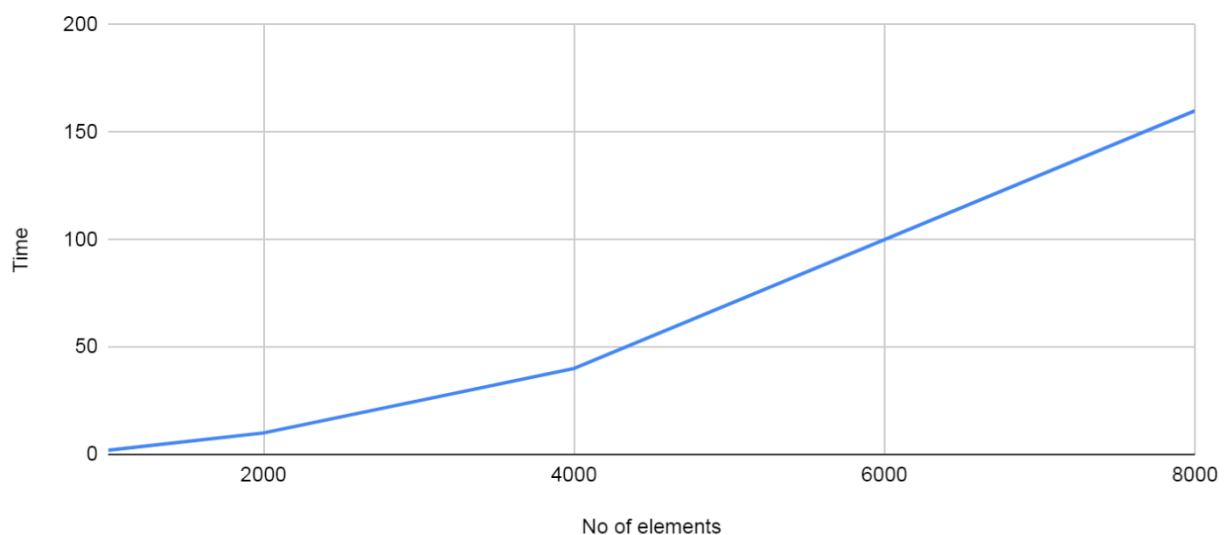
- Insertion sort takes liner time for ordered array i.e. $O(n)$
- For 1000 elements, it takes ~ 0.01 ms
- For 2000 elements, it takes ~ 0.02 ms
- For 4000 elements, it takes ~ 0.04 ms and so on.

Therefore, Order of growth is liner for ordered array in insertion sort.

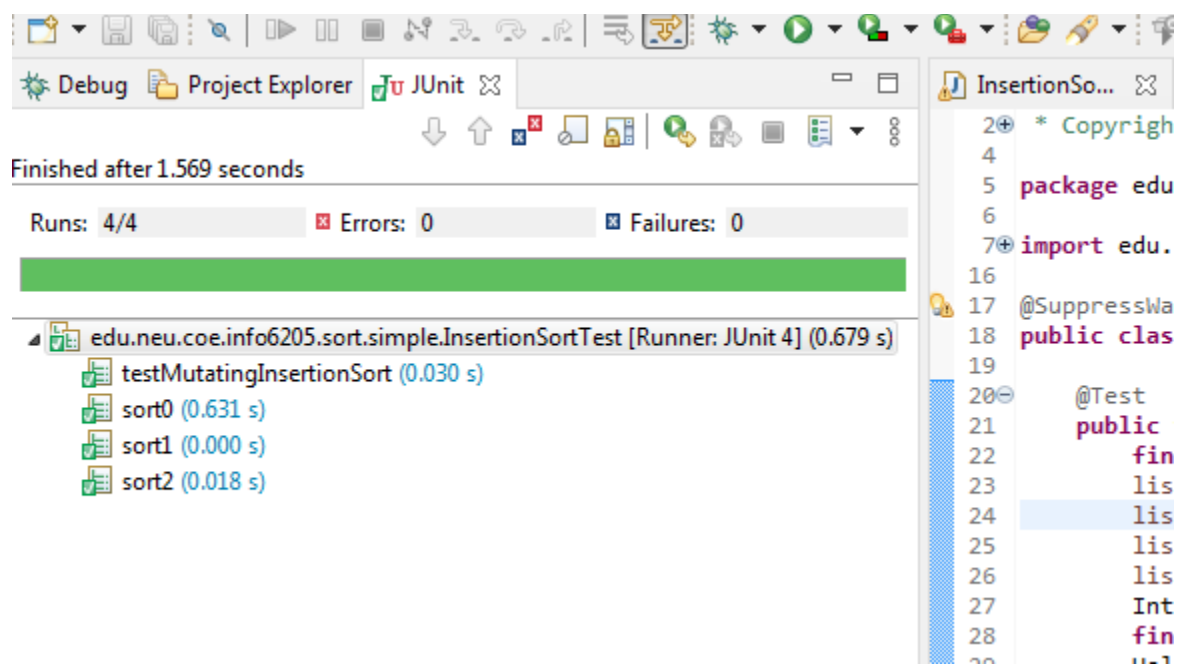
- Insertion sort takes on average $N^2/4$ time for average and worst case.
- When the array of 1000 elements is in reverse order then it takes ~ 2.5 ms ($N^2/4 = (1000*1000)/4$)
- For 4000 elements, insertion sort takes ~ 10 ms ($N^2/4 = (4000*4000)/4$)
- Partially sorted array also takes $N^2/4$ time if half of the array is sorted but the other half of the array is in reverse order. In that case N becomes 1000 for 2000 elements because we need to sort only half elements.

Graph of insertion sort for worst case (Reverse ordered or random ordered array)

Time vs No of elements



Unit test output of InsertionsortTest.java



Finished after 1.569 seconds

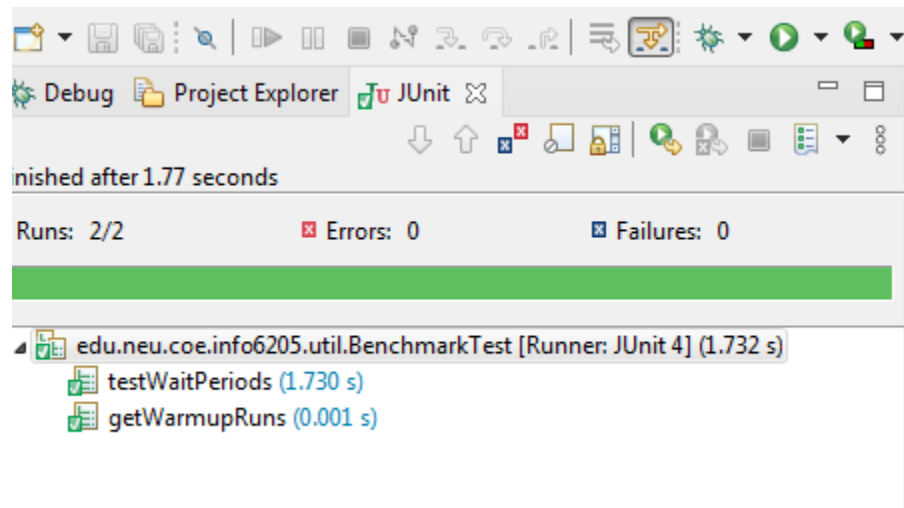
Runs: 4/4 Errors: 0 Failures: 0

edu.neu.coe.info6205.sort.simple.InsertionSortTest [Runner: JUnit 4] (0.679 s)

- testMutatingInsertionSort (0.030 s)
- sort0 (0.631 s)
- sort1 (0.000 s)
- sort2 (0.018 s)

```
2+ * Copyright
4
5 package edu
6
7+ import edu.
16
17 @SuppressWarnings
18 public class
19
20 @Test
21 public
22     fin
23     lis
24     lis
25     lis
26     lis
27     Int
28     fin
29
```

Unit test output of BenchmarkTest.java



Unit test output of TimerTest.java

