Sutiksh Verma – Sec 6

002122052

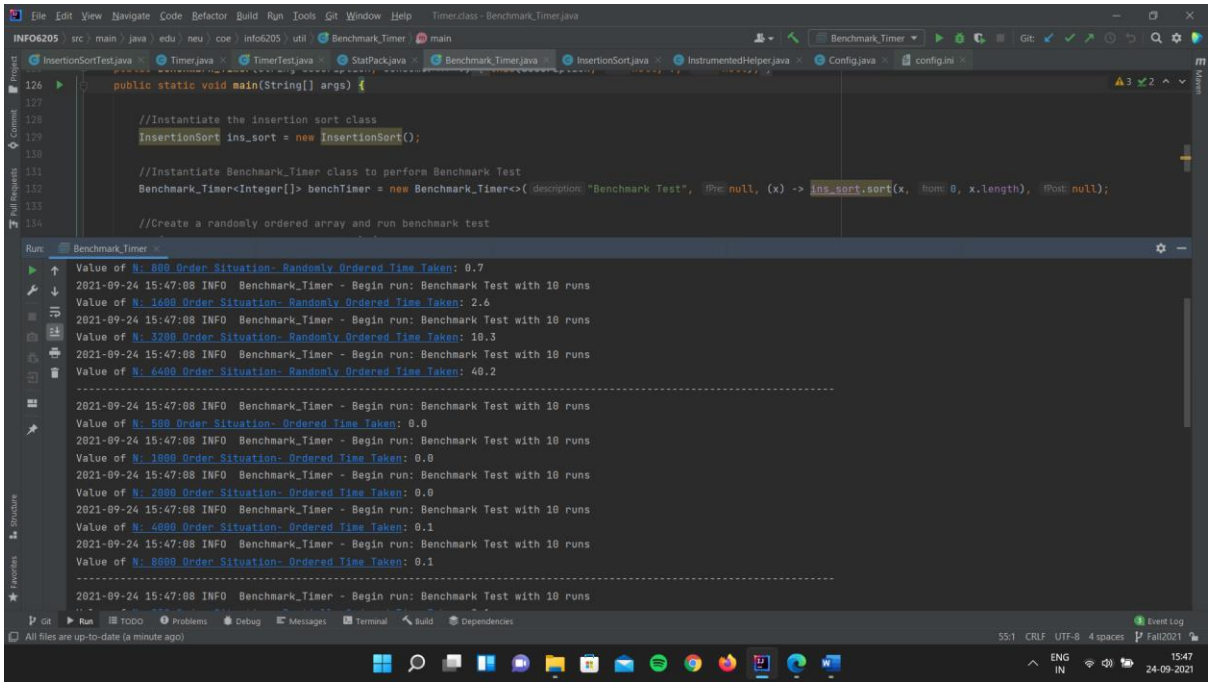
Program Structures and Algorithms

Fall 2021

Assignment No: 2

Task: To implement the class Timer.java and InsertionSort.java and also measure the running times of the sort using four different initial array ordering situations i.e. Random, Ordered, Partially-Ordered and Reverse ordered array.

Output:



Console Output:

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 200 Order Situation- Randomly Ordered Time Taken: 0.3

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 400 Order Situation- Randomly Ordered Time Taken: 0.6

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 800 Order Situation- Randomly Ordered Time Taken: 0.9

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 1600 Order Situation- Randomly Ordered Time Taken: 2.4

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 3200 Order Situation- Randomly Ordered Time Taken: 11.5

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 6400 Order Situation- Randomly Ordered Time Taken: 40.0

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 200 Order Situation- Ordered Time Taken: 0.0

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 400 Order Situation- Ordered Time Taken: 0.0

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 800 Order Situation- Ordered Time Taken: 0.0

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 1600 Order Situation- Ordered Time Taken: 0.0

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 3200 Order Situation- Ordered Time Taken: 0.0

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 6400 Order Situation- Ordered Time Taken: 0.1

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 200 Order Situation- Partially Ordered Time Taken: 0.0

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 400 Order Situation- Partially Ordered Time Taken: 0.1

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 800 Order Situation- Partially Ordered Time Taken: 0.4

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 1600 Order Situation- Partially Ordered Time Taken: 2.0

2021-09-24 16:03:52 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 3200 Order Situation- Partially Ordered Time Taken: 8.6

2021-09-24 16:03:53 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 6400 Order Situation- Partially Ordered Time Taken: 18.0

2021-09-24 16:03:53 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 200 Order Situation- Reverse Ordered Time Taken: 0.1

2021-09-24 16:03:53 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 400 Order Situation- Reverse Ordered Time Taken: 0.3

2021-09-24 16:03:53 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 800 Order Situation- Reverse Ordered Time Taken: 1.2

2021-09-24 16:03:53 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 1600 Order Situation- Reverse Ordered Time Taken: 5.0

2021-09-24 16:03:53 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 3200 Order Situation- Reverse Ordered Time Taken: 19.0

2021-09-24 16:03:53 INFO Benchmark_Timer - Begin run: Benchmark Test with 10 runs

Value of N: 6400 Order Situation- Reverse Ordered Time Taken: 72.3

Process finished with exit code 0

Relationship Conclusion: It can be observed from the results of the benchmark test:

In best-case scenario, which happens when the array is sorted, the insertion sort runs in O(n) time as it simply comparesthe elements with no swapping required

In the average-case, when the array is sorted in randomly-ordered or partially ordered manner the insertion sort runs in O(n^2) time

Similarly, in the worst-case scenario, when the array is sorted in the reverse order manner the insertion sort takes O(n^2) time because for inserting the last element, the algorithm will need n-1 comparisons and perform n-1 swaps and so on.

Therefore, we get: 2(1+2+3+…...+n-2+n-1)

= n(n-1) [Using Sum of n formula]

= n^2 – n

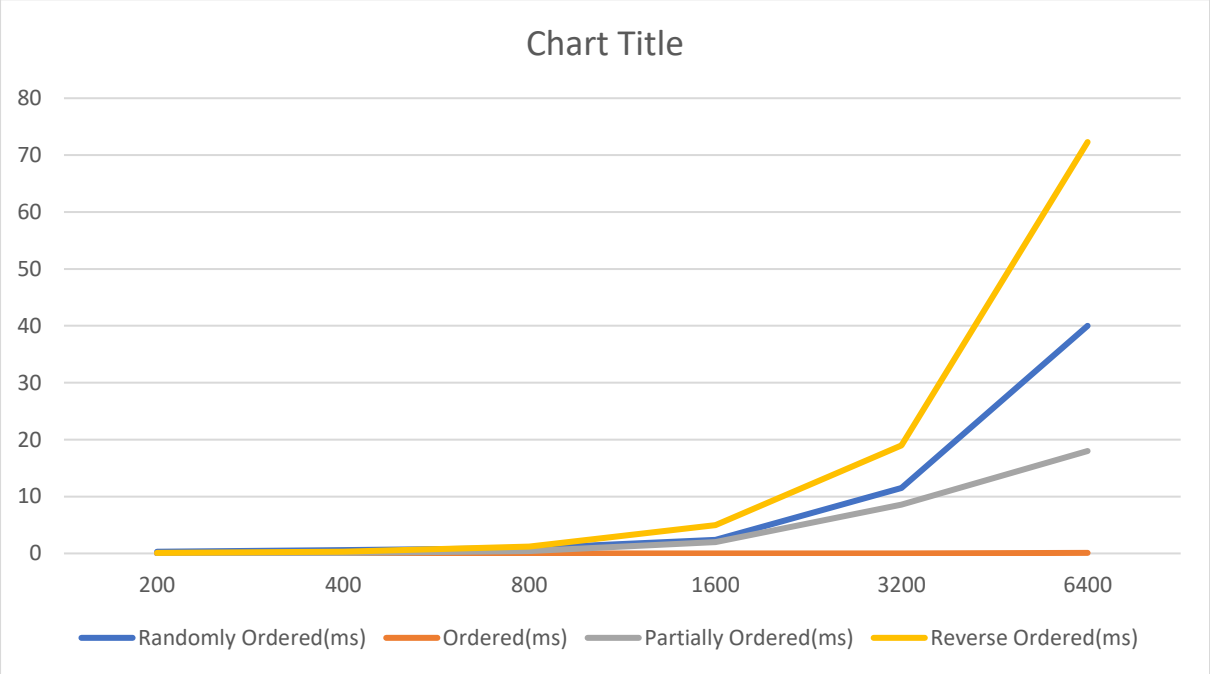
~ n^2

Thus, it can be derived that the following pattern is followed by the run time of insertion sort for different order situation in arrays

Ordered < Partially Ordered < Randomly Ordered < Reverse Ordered

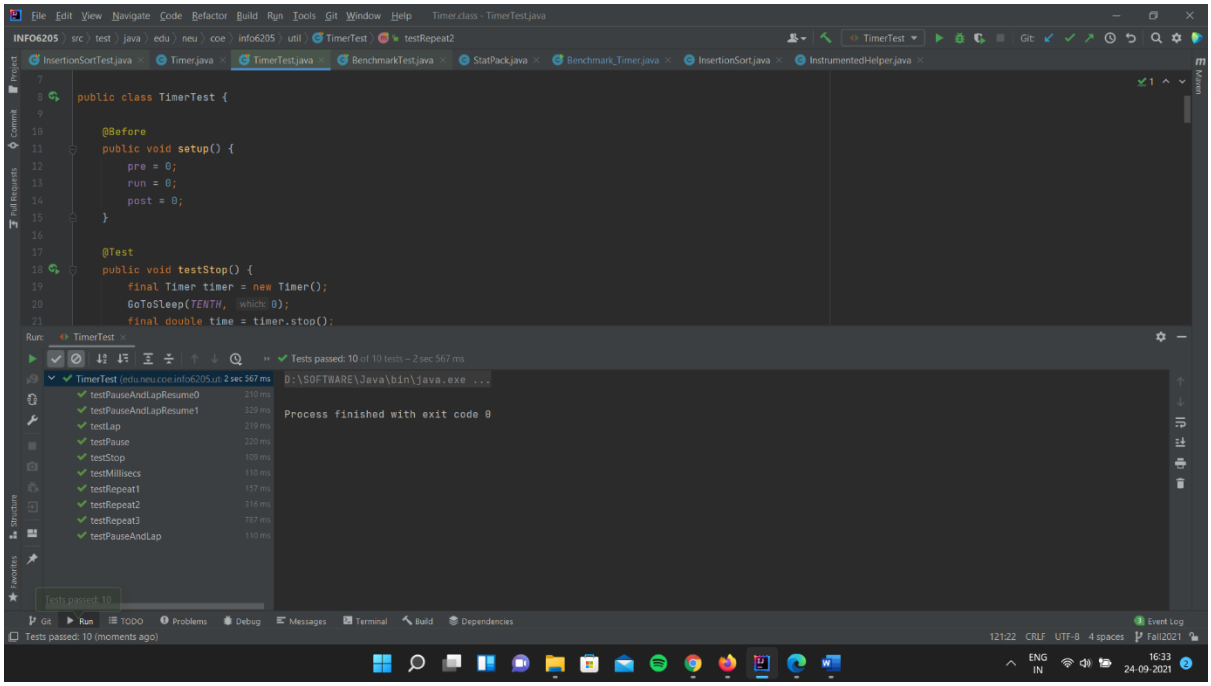
Evidence: I have attached a chart and a table to show the data of the different outputs observed conducting different experiments with different values of n. As a result, We can see the proportionate increase of the time with respect to the increased value of n using different sorted situations-

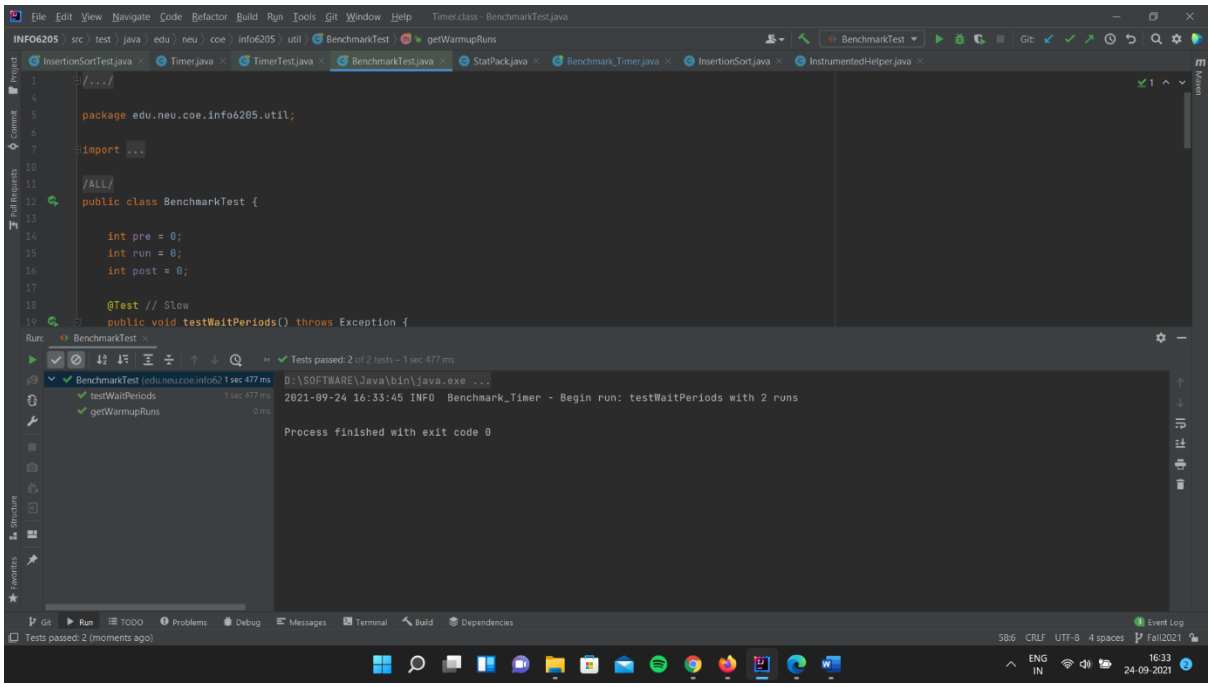
Value of N	Randomly Ordered(ms)	Ordered(ms)	Partially Ordered(ms)	Reverse Ordered(ms)
200	0.3	0.0	0.0	0.1
400	0.6	0.0	0.1	0.3
800	0.9	0.0	0.4	1.2
1600	2.4	0.0	2.0	5.0
3200	11.5	0.0	8.6	19.0
6400	40.0	0.1	18.0	72.3

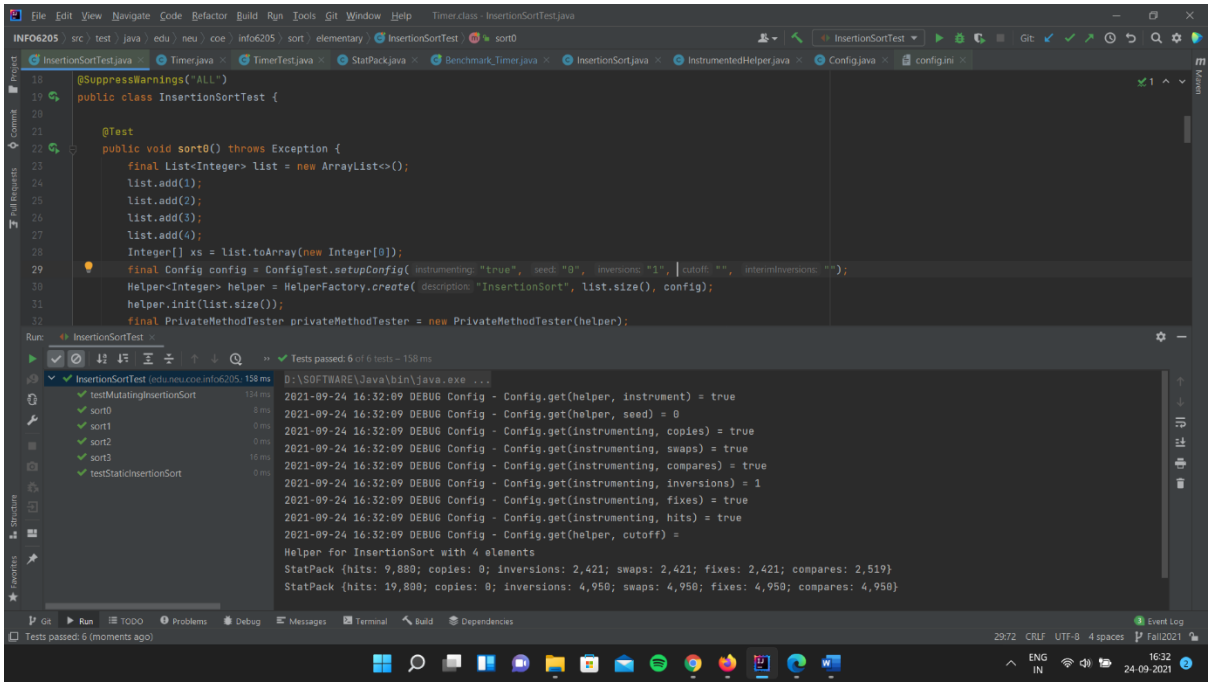


The horizontal axis is N: size of the input array.

Passed Unit Tests: Following are the images of passes unit tests which are TimerTest.java, BenchmarkTimer.java, InsertionSortTest.java.


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Timer.class - TimerTest.java
INFO6205 src test java edu neu coe info6205 util TimerTest testRepeat2
InsertionSortTest.java Timer.java TimerTest.java BenchmarkTest.java StatPack.java Benchmark_Timer.java InsertionSort.java InstrumentedHelper.java
public class TimerTest {
@Before
public void setup() {
pre = 0;
run = 0;
post = 0;
}
@Test
public void testStop() {
final Timer timer = new Timer();
GoToSleep(TENTH, which: 0);
final double time = timer.stop();
Run: TimerTest
Tests passed: 10 of 10 tests – 2 sec 567 ms
TimerTest (edu.neu.coe.info6205.uti 2 sec 567 ms
testPauseAndLapResume0 210 ms
testPauseAndLapResume1 329 ms
testLap 219 ms
testPause 220 ms
testStop 109 ms
testMillisecs 110 ms
testRepeat1 157 ms
testRepeat2 316 ms
testRepeat3 787 ms
testPauseAndLap 110 ms
D:\SOFTWARE\Java\bin\java.exe ...
Process finished with exit code 0
Git Run TODO Problems Debug Messages Terminal Build Dependencies
Event Log
Tests passed: 10 (moments ago)
121:22 CRLF UTF-8 4 spaces Fall2021
ENG IN 16:33 24-09-2021


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Timer.class - BenchmarkTest.java
INFO6205 src test java edu neu coe info6205 util BenchmarkTest getWarmupRuns
InsertionSortTest.java Timer.java TimerTest.java BenchmarkTest.java StatPack.java Benchmark_Timer.java InsertionSort.java InstrumentedHelper.java
/.../
package edu.neu.coe.info6205.util;
import ...
/ALL/
public class BenchmarkTest {
int pre = 0;
int run = 0;
int post = 0;
@Test // Slow
public void testWaitPeriods() throws Exception {
Run: BenchmarkTest
Tests passed: 2 of 2 tests – 1 sec 477 ms
BenchmarkTest (edu.neu.coe.info62 1 sec 477 ms
testWaitPeriods 1 sec 477 ms
getWarmupRuns 0 ms
D:\SOFTWARE\Java\bin\java.exe ...
2021-09-24 16:33:45 INFO Benchmark_Timer - Begin run: testWaitPeriods with 2 runs
Process finished with exit code 0
Git Run TODO Problems Debug Messages Terminal Build Dependencies
Event Log
Tests passed: 2 (moments ago)
58:6 CRLF UTF-8 4 spaces Fall2021
ENG IN 16:33 24-09-2021


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Timer.class - InsertionSortTest.java
INFO6205 › src › test › java › edu › neu › coe › info6205 › sort › elementary › InsertionSortTest › sort0
InsertionSortTest.java
Timer.java
TimerTest.java
StatPack.java
Benchmark_Timer.java
InsertionSort.java
InstrumentedHelper.java
Config.java
config.ini
@SuppressWarnings("ALL")
public class InsertionSortTest {
@Test
public void sort0() throws Exception {
final List<Integer> list = new ArrayList<>();
list.add(1);
list.add(2);
list.add(3);
list.add(4);
Integer[] xs = list.toArray(new Integer[0]);
final Config config = ConfigTest.setupConfig(instrumenting: "true", seed: "0", inversions: "1", cutoff: "", interimInversions: "");
Helper<Integer> helper = HelperFactory.create(description: "InsertionSort", list.size(), config);
helper.init(list.size());
final PrivateMethodTester privateMethodTester = new PrivateMethodTester(helper);
Run: InsertionSortTest
Tests passed: 6 of 6 tests – 158 ms
InsertionSortTest (edu.neu.coe.info6205: 158 ms
testMutatingInsertionSort 134 ms
sort0 8 ms
sort1 0 ms
sort2 0 ms
sort3 16 ms
testStaticInsertionSort 0 ms
D:\SOFTWARE\Java\bin\java.exe ...
2021-09-24 16:32:09 DEBUG Config - Config.get(helper, instrument) = true
2021-09-24 16:32:09 DEBUG Config - Config.get(helper, seed) = 0
2021-09-24 16:32:09 DEBUG Config - Config.get(instrumenting, copies) = true
2021-09-24 16:32:09 DEBUG Config - Config.get(instrumenting, swaps) = true
2021-09-24 16:32:09 DEBUG Config - Config.get(instrumenting, compares) = true
2021-09-24 16:32:09 DEBUG Config - Config.get(instrumenting, inversions) = 1
2021-09-24 16:32:09 DEBUG Config - Config.get(instrumenting, fixes) = true
2021-09-24 16:32:09 DEBUG Config - Config.get(instrumenting, hits) = true
2021-09-24 16:32:09 DEBUG Config - Config.get(helper, cutoff) =
Helper for InsertionSort with 4 elements
StatPack {hits: 9,880; copies: 0; inversions: 2,421; swaps: 2,421; fixes: 2,421; compares: 2,519}
StatPack {hits: 19,880; copies: 0; inversions: 4,950; swaps: 4,950; fixes: 4,950; compares: 4,950}
Git Run TODO Problems Debug Messages Terminal Build Dependencies
Event Log
Tests passed: 6 (moments ago)
29:72 CRLF UTF-8 4 spaces Fall2021
ENG IN
16:32 24-09-2021