

Assignment 3 (Benchmark)

Part 1

Screenshots of running Unit tests of Timer

```
INFO6205 src > test java > edu > neu > coe > info6205 > util > TimerTest > setup
no usages ▲ xiaohuanlin
@Before
public void setup() {
    pre = 0;
    run = 0;
    post = 0;
    result = 0;
}

TimerTest (edu.neu.coe.ir 2 sec 567 ms) /Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
    ✓ testPauseAndLapResume0 174ms
    ✓ testPauseAndLapResume1 316ms
    ✓ testLap 211ms
    ✓ testPause 207ms
    ✓ testStop 106ms
    ✓ testMilliseconds 105ms
    ✓ testRepeat1 125ms
    ✓ testRepeat2 241ms
    ✓ testRepeat3 611ms
    ✓ testRepeat4 368ms
    ✓ testPauseAndLap 103ms

Process finished with exit code 0

Tests passed: 11 (moments ago)
```

```
INFO6205 src > test java > edu > neu > coe > info6205 > util > TimerTest > setup
no usages ▲ xiaohuanlin
@Test
public void testStop() {
    final Timer timer = new Timer();
    GoToSleep(TENTH, which: 0);
    final double time = timer.stop();
    assertEquals(TENTH_DOUBLE, time, delta: 10);
    assertEquals(expected: 1, run);
    assertEquals(expected: 1, new PrivateMethodTester(timer).invokePrivate( name: "getLaps"));
}

no usages ▲ xiaohuanlin
@Test
public void testPauseAndLap() {
    final Timer timer = new Timer();
    final PrivateMethodTester privateMethodTester = new PrivateMethodTester(timer);
    GoToSleep(TENTH, which: 0);
    timer.pauseAndlap();
    final Long ticks = (Long) privateMethodTester.invokePrivate( name: "getTicks");
    assertEquals(TENTH_DOUBLE, actual: ticks / 1e6, delta: 12);
    assertFalse((Boolean) privateMethodTester.invokePrivate( name: "isRunning"));
    assertEquals(expected: 1, privateMethodTester.invokePrivate( name: "getLaps"));
}

no usages ▲ xiaohuanlin
@Test
public void testPauseAndLapResume0() {
    final Timer timer = new Timer();
    final PrivateMethodTester privateMethodTester = new PrivateMethodTester(timer);
    GoToSleep(TENTH, which: 0);
    timer.pauseAndlap();
    timer.resume();
    assertTrue((Boolean) privateMethodTester.invokePrivate( name: "isRunning"));
    assertEquals(expected: 1, privateMethodTester.invokePrivate( name: "getLaps"));
}

no usages ▲ xiaohuanlin

Tests passed: 11 (a minute ago)
```

INFO6205 src > test java > edu > neu > coe info6205 util TimerTest setup

```
51 13/05/22 Xiaohuanlin *
52 13/05/22 Xiaohuanlin * @Test
53 13/05/22 Xiaohuanlin *
54 13/05/22 Xiaohuanlin *
55 13/05/22 Xiaohuanlin *
56 13/05/22 Xiaohuanlin *
57 13/05/22 Xiaohuanlin *
58 13/05/22 Xiaohuanlin *
59 13/05/22 Xiaohuanlin *
60 13/05/22 Xiaohuanlin *
61 13/05/22 Xiaohuanlin *
62 13/05/22 Xiaohuanlin *
63 13/05/22 Xiaohuanlin *

no usages ▲ xiaohuanlin
64 13/05/22 Xiaohuanlin *
65 13/05/22 Xiaohuanlin * @Test
66 13/05/22 Xiaohuanlin *
67 13/05/22 Xiaohuanlin *
68 13/05/22 Xiaohuanlin *
69 13/05/22 Xiaohuanlin *
70 13/05/22 Xiaohuanlin *
71 13/05/22 Xiaohuanlin *
72 13/05/22 Xiaohuanlin *
73 13/05/22 Xiaohuanlin *
74 13/05/22 Xiaohuanlin *

no usages ▲ xiaohuanlin
75 13/05/22 Xiaohuanlin *
76 13/05/22 Xiaohuanlin * @Test
77 13/05/22 Xiaohuanlin *
78 13/05/22 Xiaohuanlin *
79 13/05/22 Xiaohuanlin *
80 13/05/22 Xiaohuanlin *
81 13/05/22 Xiaohuanlin *
82 13/05/22 Xiaohuanlin *
83 13/05/22 Xiaohuanlin *
84 13/05/22 Xiaohuanlin *
85 13/05/22 Xiaohuanlin * assertEquals(expected: 2, run);
86 13/05/22 Xiaohuanlin * assertEquals(expected: 1, run);

no usages ▲ xiaohuanlin
87 13/05/22 Xiaohuanlin *
88 13/05/22 Xiaohuanlin * @Test
89 13/05/22 Xiaohuanlin *
90 13/05/22 Xiaohuanlin *
91 13/05/22 Xiaohuanlin *
92 13/05/22 Xiaohuanlin *
93 13/05/22 Xiaohuanlin *
94 13/05/22 Xiaohuanlin *
95 13/05/22 Xiaohuanlin *
96 13/05/22 Xiaohuanlin *

no usages ▲ xiaohuanlin
97 13/05/22 Xiaohuanlin *
98 13/05/22 Xiaohuanlin * @Test
99 13/05/22 Xiaohuanlin *
100 13/05/22 Xiaohuanlin *
101 13/05/22 Xiaohuanlin *
102 13/05/22 Xiaohuanlin *
103 13/05/22 Xiaohuanlin *
104 13/05/22 Xiaohuanlin *
105 13/05/22 Xiaohuanlin *
106 13/05/22 Xiaohuanlin *
107 13/05/22 Xiaohuanlin *
108 13/05/22 Xiaohuanlin *
109 13/05/22 Xiaohuanlin *
110 13/05/22 Xiaohuanlin *

no usages ▲ xiaohuanlin
111 13/05/22 Xiaohuanlin *
112 13/05/22 Xiaohuanlin * @Test
113 13/05/22 Xiaohuanlin *
114 13/05/22 Xiaohuanlin *
115 13/05/22 Xiaohuanlin *
116 13/05/22 Xiaohuanlin *
117 13/05/22 Xiaohuanlin *
118 13/05/22 Xiaohuanlin *
119 13/05/22 Xiaohuanlin *
120 13/05/22 Xiaohuanlin *
```

Tests passed: 11 (2 minutes ago)

INFO6205 src > test java > edu > neu > coe info6205 util TimerTest testPauseAndLapResume1

```
87 13/05/22 Xiaohuanlin *
88 13/05/22 Xiaohuanlin * @Test
89 13/05/22 Xiaohuanlin *
90 13/05/22 Xiaohuanlin *
91 13/05/22 Xiaohuanlin *
92 13/05/22 Xiaohuanlin *
93 13/05/22 Xiaohuanlin *
94 13/05/22 Xiaohuanlin *
95 13/05/22 Xiaohuanlin *
96 13/05/22 Xiaohuanlin *

no usages ▲ xiaohuanlin
97 13/05/22 Xiaohuanlin *
98 13/05/22 Xiaohuanlin * @Test
99 13/05/22 Xiaohuanlin *
100 13/05/22 Xiaohuanlin *
101 13/05/22 Xiaohuanlin *
102 13/05/22 Xiaohuanlin *
103 13/05/22 Xiaohuanlin *
104 13/05/22 Xiaohuanlin *
105 13/05/22 Xiaohuanlin *
106 13/05/22 Xiaohuanlin *
107 13/05/22 Xiaohuanlin *
108 13/05/22 Xiaohuanlin *
109 13/05/22 Xiaohuanlin *
110 13/05/22 Xiaohuanlin *

no usages ▲ xiaohuanlin
111 13/05/22 Xiaohuanlin *
112 13/05/22 Xiaohuanlin * @Test
113 13/05/22 Xiaohuanlin *
114 13/05/22 Xiaohuanlin *
115 13/05/22 Xiaohuanlin *
116 13/05/22 Xiaohuanlin *
117 13/05/22 Xiaohuanlin *
118 13/05/22 Xiaohuanlin *
119 13/05/22 Xiaohuanlin *
120 13/05/22 Xiaohuanlin *
```

Tests passed: 11 (2 minutes ago)

```

INFO6205 > src > test > java > edu > neu > coe > info6205 > util > TimerTest > testPauseAndLapResume
insertionSortTest.java <-- InsertionSort.java <-- threessum/Source.java <-- ThreeSum.java <-- ThreeSumQuadratic.java <-- elementary/Source.java <-- InsertionSortOpt.java <-- TicketTest.java <-- TimerTest.java

122 13/05/22 Xiaohuanlin *
123 13/05/22 Xiaohuanlin *
124 13/05/22 Xiaohuanlin *
125 13/05/22 Xiaohuanlin *

126 13/05/22 Xiaohuanlin *
127 13/05/22 Xiaohuanlin * @Test // Slow
128 13/05/22 Xiaohuanlin *
129 13/05/22 Xiaohuanlin *
130 13/05/22 Xiaohuanlin *
131 13/05/22 Xiaohuanlin *
132 13/05/22 Xiaohuanlin *
133 13/05/22 Xiaohuanlin *
134 13/05/22 Xiaohuanlin *
135 13/05/22 Xiaohuanlin *
136 13/05/22 Xiaohuanlin *
137 13/05/22 Xiaohuanlin *
138 13/05/22 Xiaohuanlin *
139 13/05/22 Xiaohuanlin *
140 13/05/22 Xiaohuanlin *
141 13/05/22 Xiaohuanlin *
142 13/05/22 Xiaohuanlin *
143 13/05/22 Xiaohuanlin *

144 13/05/22 Xiaohuanlin *
145 13/05/22 Xiaohuanlin * @Test // Slow
146 13/05/22 Xiaohuanlin *
147 13/05/22 Xiaohuanlin *
148 13/05/22 Xiaohuanlin *
149 13/05/22 Xiaohuanlin *
150 13/05/22 Xiaohuanlin *
151 13/05/22 Xiaohuanlin *
152 13/05/22 Xiaohuanlin *
153 13/05/22 Xiaohuanlin *
154 13/05/22 Xiaohuanlin *
155 13/05/22 Xiaohuanlin *

no usages ▾ xiaohuanlin
@Test // Slow
public void testRepeat3() {
    final Timer timer = new Timer();
    final int zzz = 20;
    final double mean = timer.repeat( n: 10, () -> zzz, t -> {
        GoToSleep(t, which: 0);
        return null;
    }, t -> {
        GoToSleep(t, which: -1);
        return t;
    }, t -> GoToSleep( mSecs: 10, which: 1));
    assertEquals( expected: 10, new PrivateMethodTester(timer).invokePrivate( name: "getLaps"));
    assertEquals(zzz, mean, delta: 6);
    assertEquals( expected: 10, run);
    assertEquals( expected: 10, pre);
    assertEquals( expected: 10, post);
}

no usages ▾ xiaohuanlin
@Test // Slow
public void testRepeat4() {
    final Timer timer = new Timer();
    final int zzz = 20;
    final double mean = timer.repeat( n: 10,
        () -> zzz, // supplier
        t -> { // function
            result = t;
            GoToSleep( mSecs: 10, which: 0);
            return null;
        }, t -> { // pre-function
            GoToSleep( mSecs: 10, which: -1);
        }
)
}

```

Part 2 Screenshots of running Unit tests of Insertion Sort

```

INFO6205 > src > test > java > edu > neu > coe > info6205 > sort > elementary > InsertionSortTest > sort1
InsertionSortTest.java <-- InsertionSort.java <-- threessum/Source.java <-- ThreeSum.java <-- ThreeSumQuadratic.java <-- elementary/Source.java <-- InsertionSortOpt.java <-- TicketTest.java <-- TimerTest.java

Run: InsertionSortTest
Tests passed: 6 of 6 tests - 93ms
Process finished with exit code 0

```

Test Case	Time (ms)	Helper for InsertionSort with 4 elements	StatPack {hits: ...}
testMutatingInsertionSort	77ms	Helper for InsertionSort with 4 elements	{hits: 9,880, normalized=21.454; copies: 0, normalized=0.000; inversions: 2,421, normalized=5.257; swaps: 2,421, normalized=10.749}
sort0	4ms		{hits: 19,880, normalized=42.995; copies: 0, normalized=0.000; inversions: 4,950, normalized=10.749; swaps: 4,950, normalized=10.749}
sort1	6ms		
sort2	3ms		
sort3	2ms		
testStaticInsertionSort	1ms		

```
INFO6205 src > test > java > edu > neu > coe > info6205 > sort > elementary > InsertionSortTest > sort1
no usages ▲ xiaohuanlin +1
@Text
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 = Config.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);
    final StatPack statPack = (StatPack) privateMethodTester.invokePrivate(name: "getStatPack");
    SortWithHelper<Integer> sorter = new InsertionSort<~>(helper);
    sorter.preProcess(xs);
    Integer[] ys = sorter.sort(xs);
    assertTrue(helper.sorted(ys));
    sorter.postProcess(ys);
    final int compares = (int) statPack.getStatistics(InstrumentedHelper.COMPARES).mean();
    assertEquals(expected: list.size() - 1, compares);
    final int inversions = (int) statPack.getStatistics(InstrumentedHelper.INVERSIONS).mean();
    assertEquals(expected: 0L, inversions);
    final int fixes = (int) statPack.getStatistics(InstrumentedHelper.FIXES).mean();
    assertEquals(inversions, fixes);
}
no usages ▲ xiaohuanlin
@Text
public void sort1() throws Exception {
    final List<Integer> list = new ArrayList<>();
    list.add(3);
    list.add(4);
    list.add(2);
    list.add(1);
    Integer[] xs = list.toArray(new Integer[0]);
    BaseHelper<Integer> helper = new BaseHelper<>(description: "InsertionSort", xs.length, Config.load(InsertionSortTest.class));
}
Tests passed: 6 (4 minutes ago)
50:10 LF UTF-8 4 spaces Spring2023
```

```
INFO6205 src > test > java > edu > neu > coe > info6205 > sort > elementary > InsertionSortTest > sort2
no usages ▲ xiaohuanlin +1
@Text
public void testStaticInsertionSort() throws IOException {
    final List<Integer> list = new ArrayList<>();
    list.add(3);
    list.add(4);
    list.add(2);
    list.add(1);
    Integer[] xs = list.toArray(new Integer[0]);
    InsertionSort.sort(xs);
    assertTrue(condition: xs[0] < xs[1] && xs[1] < xs[2] && xs[2] < xs[3]);
}
no usages ▲ xiaohuanlin +1
@Text
public void sort2() throws Exception {
    final Config config = Config.setupConfig(instrumenting: "true", seed: "0", inversions: "1", cutoff: "", interimInversions: "");
    int n = 100;
    Helper<Integer> helper = HelperFactory.create(description: "InsertionSort", n, config);
    helper.init(n);
    final PrivateMethodTester privateMethodTester = new PrivateMethodTester(helper);
    final StatPack statPack = (StatPack) privateMethodTester.invokePrivate(name: "getStatPack");
    Integer[] xs = helper.random(Integer.class, r -> r.nextInt(bound: 1000));
    SortWithHelper<Integer> sorter = new InsertionSort<~>(helper);
    sorter.preProcess(xs);
    Integer[] ys = sorter.sort(xs);
    assertTrue(helper.sorted(ys));
    sorter.postProcess(ys);
    final int compares = (int) statPack.getStatistics(InstrumentedHelper.COMPARES).mean();
    // NOTE: these are supposed to match within about 12%.
    // Since we set a specific seed, this should always succeed.
    // If we use true random seed and this test fails, just increase the delta a little.
    assertEquals(expected: 1.0, actual: 4.0 * compares / n / (n - 1), delta: 0.12);
    final int inversions = (int) statPack.getStatistics(InstrumentedHelper.INVERSIONS).mean();
    final int fixes = (int) statPack.getStatistics(InstrumentedHelper.FIXES).mean();
    System.out.println(statPack);
    assertEquals(inversions, fixes);
}
Tests passed: 6 (4 minutes ago)
100:77 LF UTF-8 4 spaces Spring2023
```

```

INFO6205 > src > test > java > edu > neu > coe > info6205 > sort > elementary > InsertionSortTest > sort2
  InsertionSortTest.java <--> InsertionSort.java <--> ThreeSum.java <--> ThreeSumQuadratic.java <--> elementary/Source.java <--> InsertionSortOpt.java <--> TicketTest.java <--> TimerTest.java
  111 13/05/22 Xiaohuanlin
  112 13/05/22 Xiaohuanlin
  113 13/05/22 Xiaohuanlin
  114 13/05/22 Xiaohuanlin
  115 13/05/22 Xiaohuanlin
  ...
  no usages + xiaohuanlin+1
  @Test
  public void sort3() throws Exception {
    final Config config = Config.setupConfig( instrumenting: "true", seed: "0", inversions: "1", cutoff: "", interimInversions: "" );
    int n = 100;
    Helper<Integer> helper = HelperFactory.create( description: "InsertionSort", n, config );
    helper.init();
    final PrivateMethodTester privateMethodTester = new PrivateMethodTester(helper);
    final StatPack statPack = (StatPack) privateMethodTester.invokePrivate( name: "getStatPack" );
    Integer[] xs = new Integer[n];
    for (int i = 0; i < n; i++) xs[i] = n - i;
    SortWithHelper<Integer> sorter = new InsertionSort<~>(helper);
    sorter.preProcess(xs);
    Integer[] ys = sorter.sort(xs);
    assertTrue(helper.sorted(ys));
    sorter.postProcess(ys);
    final int compares = (int) statPack.getStatistics(InstrumentedHelper.COMPARES).mean();
    // NOTE: these are supposed to match within about 12%.
    // Since we set a specific seed, this should always succeed.
    // If we use true random seed and this test fails, just increase the delta a little.
    assertEquals( expected: 4950, compares );
    final int inversions = (int) statPack.getStatistics(InstrumentedHelper.INVERSIONS).mean();
    final int fixes = (int) statPack.getStatistics(InstrumentedHelper.FIXES).mean();
    System.out.println(statPack);
    assertEquals(inversions, fixes);
  }
  ...
  no usages
  final static LazyLogger logger = new LazyLogger(InsertionSort.class);
}

```

Commit: 10/01/23 Akshaymata *
 Pull Requests: 13/05/22 Xiaohuanlin
 Bookmarks: 13/05/22 Xiaohuanlin
 Structure: 13/05/22 Xiaohuanlin
 Tests passed: 6 (6 minutes ago)

Part 3 Code Execution results

```

Run: InsertionSort <--> /Library/Java/JavaVirtualMachines/jdk-18.0.2.1.jdk/Contents/Home/bin/java ...
Array with size of 500 run 100 times:
Sorted array:0.06 (msec)
Random array:0.33 (msec)
Reverse array:0.46 (msec)
Partially ordered array:0.23 (msec)
Array with size of 1000 run 100 times:
Sorted array:0.05 (msec)
Random array:0.67 (msec)
Reverse array:1.28 (msec)
Partially ordered array:0.54 (msec)
Array with size of 2000 run 100 times:
Sorted array:0.06 (msec)
Random array:2.61 (msec)
Reverse array:5.05 (msec)
Partially ordered array:2.1 (msec)
Array with size of 4000 run 100 times:
Sorted array:0.06 (msec)
Random array:10.14 (msec)
Reverse array:20.21 (msec)
Partially ordered array:7.68 (msec)
Array with size of 8000 run 100 times:
Sorted array:0.15 (msec)
Random array:40.84 (msec)
Reverse array:81.55 (msec)
Partially ordered array:31.09 (msec)

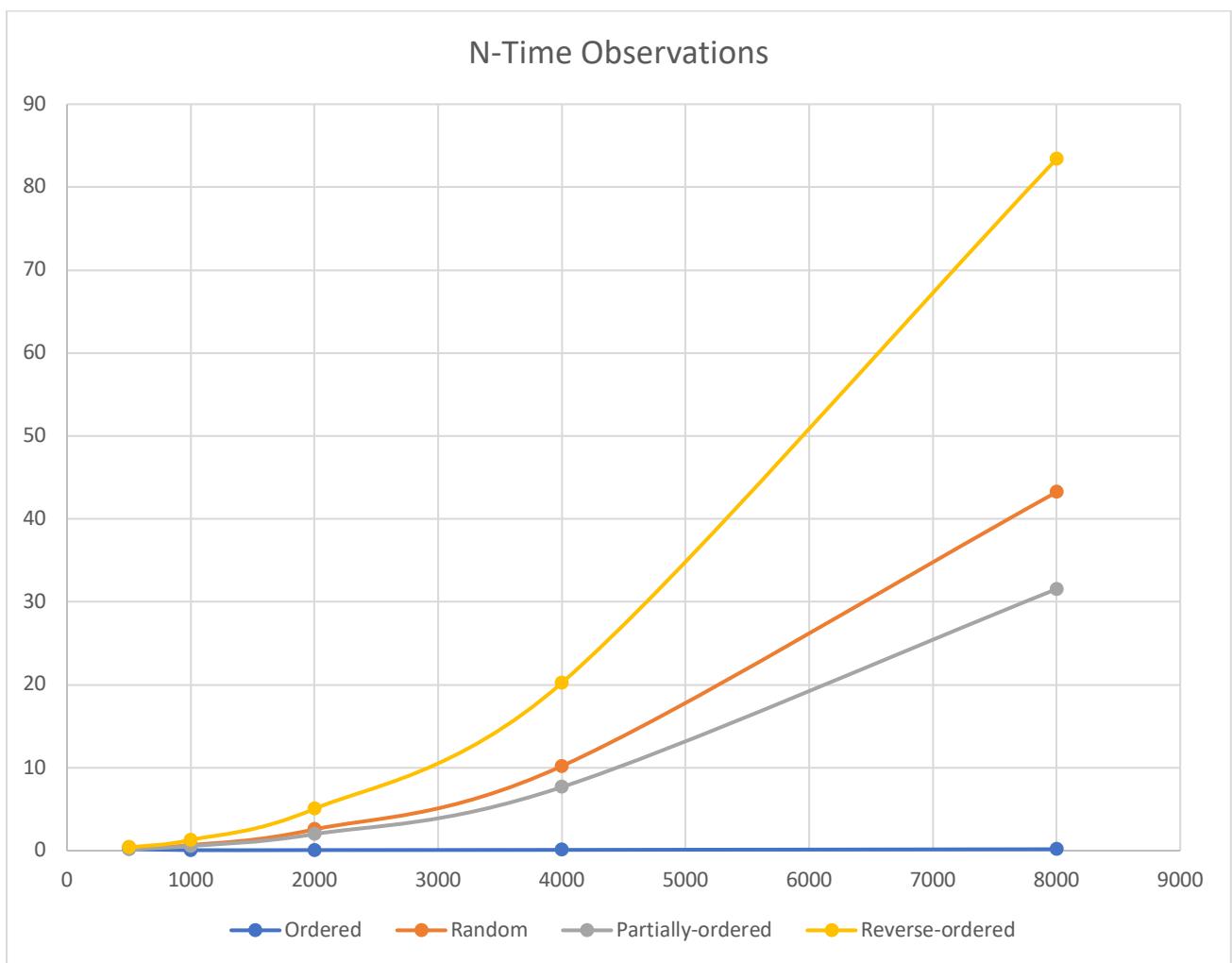
Process finished with exit code 0

```

Build completed successfully in 2 sec, 14 ms (2 minutes ago)

Timing Observations of all Ordered, Random, Partially-ordered and Reverse-ordered arrays

N (Array size)	Ordered Time	Random Time	Partially-ordered Time	Reverse-ordered Time
500	0.21	0.4	0.18	0.38
1000	0.06	0.67	0.56	1.29
2000	0.07	2.56	1.99	5.04
4000	0.09	10.19	7.66	20.22
8000	0.17	43.23	31.54	83.4



Graph

Conclusion

As per the graph, following is time taken by all the arrays in ascending order:

Ordered < Partially-Ordered < Random < Reverse-Ordered

Hence, sorted array is the best case of insertion sort while reverse-ordered array is the worst case of insertion sort.