

Program Structures and Algorithms
Spring 2024

NAME: Saurabh Srivastava

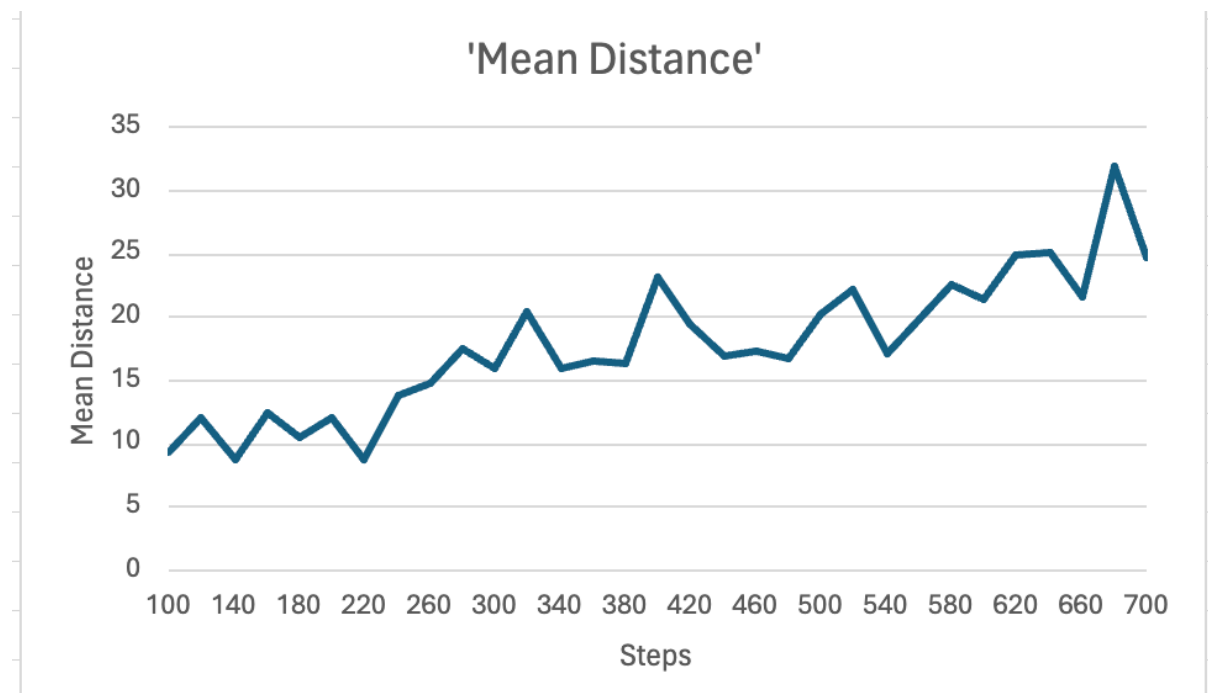
NUID: 002895225

GITHUB LINK: <https://github.com/ssaurabh760/INFO6205>

Task: The objective of our random walk experiments was to explore the relationship between the mean distance (d) and the number of steps (m) taken by a drunkard in a stochastic environment.

Relationship Conclusion: On an average, the equation that was derived is $d^2 = m$

Evidence to support that conclusion: A line chart with exact values of how d varies with m .



Unit Test Screenshots:

The first screenshot shows the `RandomWalk.java` file in an IDE. The code defines a `RandomWalk` class with a `main` method that takes command-line arguments and a `randomWalkMulti` method that performs multiple experiments and prints the mean distance.

```
101 int minSteps = Integer.parseInt(args[0]);
102 int maxSteps = Integer.parseInt(args[1]);
103 int stepSize = Integer.parseInt(args[2]);
104
105 int n = 10;
106 if (args.length > 3) n = Integer.parseInt(args[3]);
107
108 for (int m = minSteps; m <= maxSteps; m += stepSize) {
109     double meanDistance = randomWalkMulti(m, n);
110     System.out.println(m + " steps: " + meanDistance + " over " + n + " experiments");
111 }
112 }
```

The terminal output shows the results of running the program with arguments `100 700 20 10`:

```
saunabharivastava@Saurabhs-MacBook-Air randomwalk % java -cp . edu.neu.coe.info6205.randomwalk.RandomWalk 100 700 20 10
100 steps: 9.815622144778711 over 10 experiments
120 steps: 7.65683882276879 over 10 experiments
140 steps: 9.775767523799995 over 10 experiments
160 steps: 12.771828380417225 over 10 experiments
180 steps: 11.68223769685348 over 10 experiments
200 steps: 8.375839771146312 over 10 experiments
220 steps: 14.244231287312456 over 10 experiments
240 steps: 15.747671526665839 over 10 experiments
260 steps: 15.197849492854564 over 10 experiments
280 steps: 14.881593249938895 over 10 experiments
```

The second screenshot shows the `RandomWalkTest.java` file in the IDE. The code defines a `RandomWalkTest` class with a `testMove0` method that tests the `move` method of the `RandomWalk` class.

```
15 @Test
16 public void testMove0() {
17     RandomWalk rw = new RandomWalk();
18     PrivateMethodTester pmt = new PrivateMethodTester(rw);
19     pmt.invokePrivate( name: "move", ...parameters: 1, 0);
20     assertEquals( expected: 1.0, rw.distance(), delta: 1.0E-7);
21 }
22
23 /**
24  *
25  */
26 @Test
27 public void testMove1() {
28     RandomWalk rw = new RandomWalk();
29     PrivateMethodTester pmt = new PrivateMethodTester(rw);
30     pmt.invokePrivate( name: "move", ...parameters: 1, 0);
31     assertEquals( expected: 1.0, rw.distance(), delta: 1.0E-7);
32     pmt.invokePrivate( name: "move", ...parameters: 1, 0);
33     assertEquals( expected: 2.0, rw.distance(), delta: 1.0E-7);
34     pmt.invokePrivate( name: "move", ...parameters: -1, 0);
35 }
```

The Run window shows the results of running the tests:

```
Run RandomWalkTest
Tests passed: 6 of 6 tests - 394 ms
testRandomWalk2 39 ms
testMove0 62 ms
testMove1 6 ms
testMove2 6 ms
testMove3 7 ms
testRandomWalk 274 ms
Process finished with exit code 0
```