

Assignment 1 – Random Walk

Course: INFO6205

Professor: Robin Hillyard

Student Name: Natarajan Lekshmi Narayana Pillai

NUID: 002766033

Conclusion

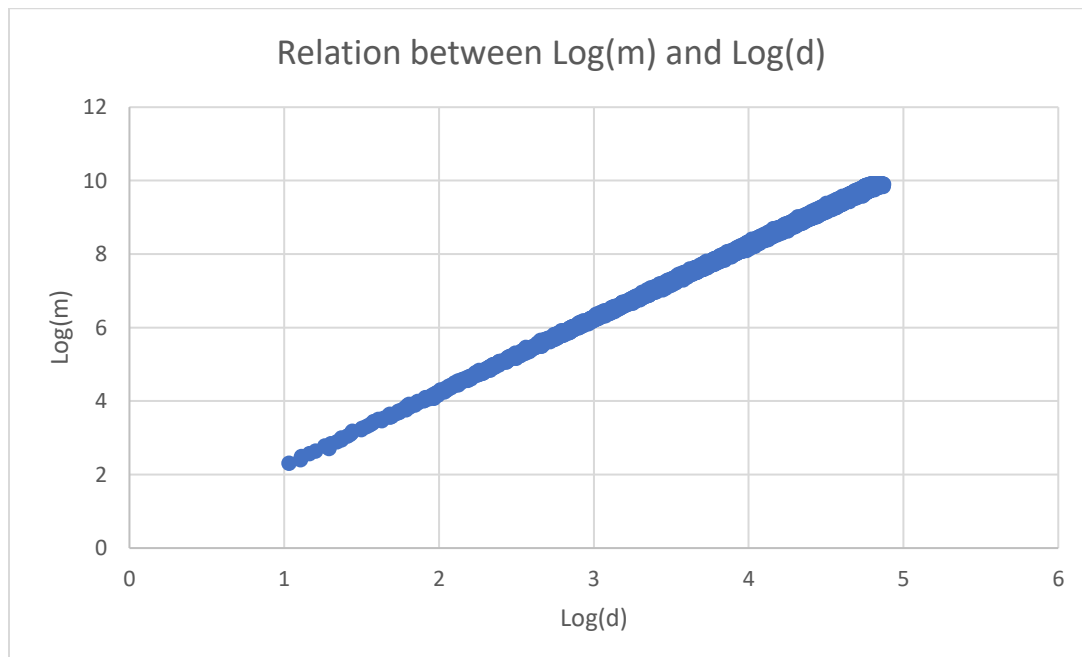
As seen from the data and the graphs, there is a proportional relation between “m” and “d”, where “m” is the number of steps and “d” is the distance from the starting point.

To find the relation between “m” and “d”, I imagined the equation to be $m = cd^x$. From here I put log on both sides to get $\text{Log}(m) = x\text{Log}(d) + \text{Log}(c)$. Assuming that the line is passing through the origin, $\text{Log}(c)$ will become irrelevant, and the equation will simplify to its final form $x = \text{Log}(m)/\text{Log}(d)$, where x is the ratio.

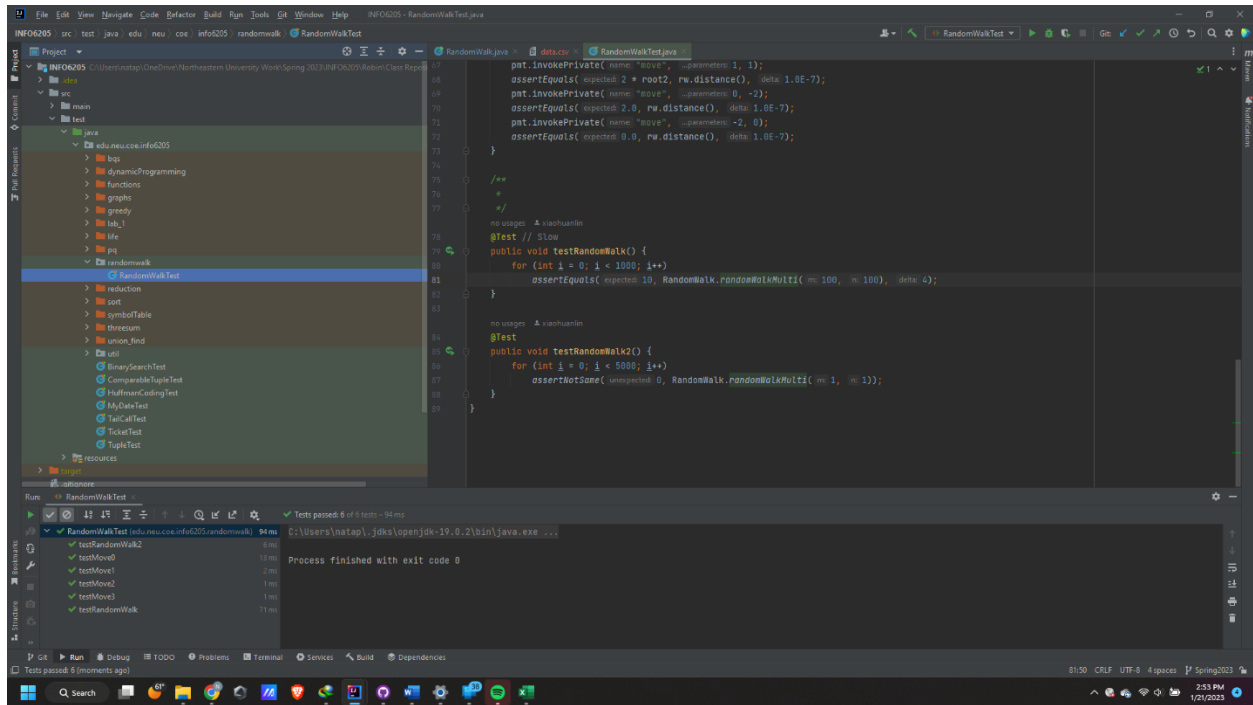
After calculating the average of the ratio using the values we got from the 20000 test cases, we get the ratio as 2.056861. Therefore, the final equation would be $m = d^2$, stating that the ratio (x) is equal to 2.

The relation between “m” and “d” can also be seen clearly when plotting $\log(m)$ by $\log(d)$, which produces a relatively straight line.

Evidence



Unit Test Passed Proof



The screenshot displays an IDE window for a project named 'INFO6205'. The left sidebar shows a project tree with a 'test' directory containing 'RandomWalkTest'. The main editor shows the source code for 'RandomWalkTest.java', which includes several assertions and two test methods: 'testRandomWalk()' and 'testRandomWalk2()'. The bottom panel shows the 'Run' output, indicating that all tests passed successfully.

```
INFO6205 - RandomWalkTest.java
67 pat.invokePrivate(name: "move", parameters: 1, 1);
68 assertEquals(expected: 2 * root2, rw.distance(), delta: 1.0E-7);
69 pat.invokePrivate(name: "move", parameters: 0, -2);
70 assertEquals(expected: 2.0, rw.distance(), delta: 1.0E-7);
71 pat.invokePrivate(name: "move", parameters: -2, 0);
72 assertEquals(expected: 0.0, rw.distance(), delta: 1.0E-7);
73 }
74
75 /**
76  *
77  */
78 no usages & xiaohuadin
79
80 @test // Slow
81 public void testRandomWalk() {
82     for (int i = 0; i < 1000; i++)
83         assertEquals(expected: 10, RandomWalk.randomWalkMulti(m: 100, n: 100), delta: 4);
84 }
85
86 no usages & xiaohuadin
87
88 @test
89 public void testRandomWalk2() {
90     for (int i = 0; i < 5000; i++)
91         assertEquals(expected: 0, RandomWalk.randomWalkMulti(m: 1, n: 1));
92 }
93 }
```

Run - RandomWalkTest

Tests passed: 6 of 6 tests - 94ms

Test Name	Duration
testRandomWalk	94ms
testRandomWalk2	6ms
testMove0	13ms
testMove1	2ms
testMove2	1ms
testMove3	1ms
testRandomWalk	71ms

Process finished with exit code 0

8:50 CRLF UTF-8 4 spaces Spring2023