Chaman Betrabet (002784662)

Program Structures & Algorithms
Assignment 1

Conclusion:

The distance (*d*) the man is from the lamp post after *m* number of steps is given by the square root of the sum of the squares of the distance he has travelled in the x and y directions.

As the man's steps are of equal length and randomly chosen from North, South, East and West, we can assume this is a 2D Random walk and hence the distance he has travelled in the x direction is equal to the number of steps he has taken East or West (minus the no. of steps he has taken West or East). The distance he has travelled in the y direction is equal to the number of steps he has taken North or South (minus the no. of steps he has taken South or North).

Therefore, d = √((m - the number of steps taken West/East)^2 + (m - the number of steps taken South/North)^2).

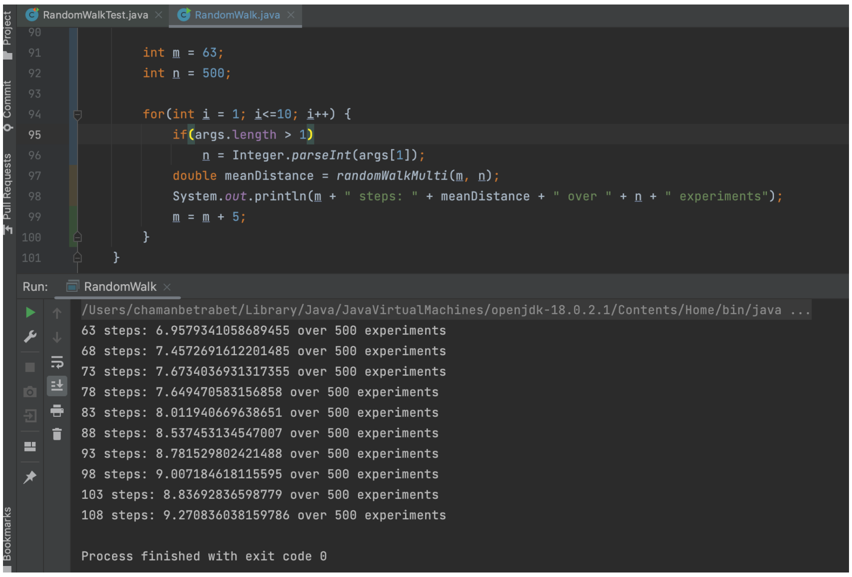
In general, as the direction of each step is randomly chosen, the expected value of the distance from the starting point after m steps is the square root of the number of steps taken, m plus the variance, hence we can conclude the distance is proportionate to the mean distance.

$d = \sqrt{n} + V$
where V = variance
$\therefore d \propto \sqrt{n}$

Evidence:

- **Output:** In order to test different values of no of steps taken (m) and observe the relation between steps and distance (d), few changes have been done to the Randomwalk.java file which has been attached along with the submission for reference. Below are the results:

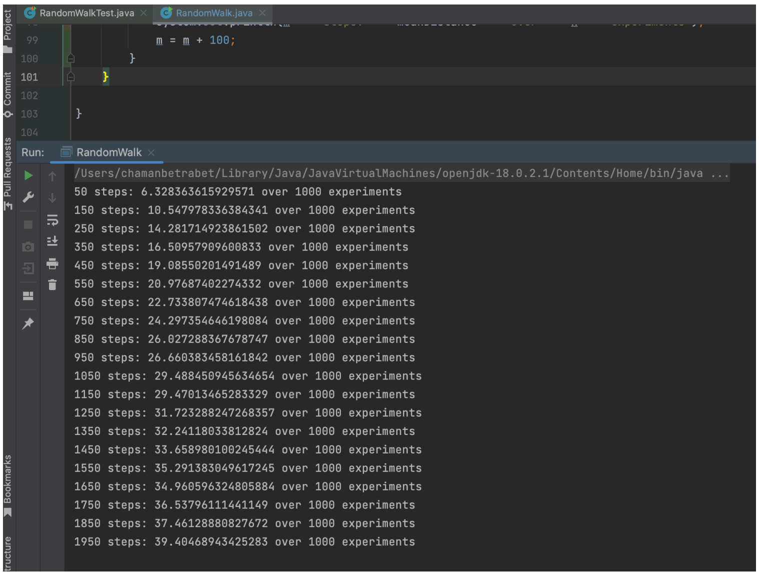
    1. Experiment 1: Number of experiments = 500 for 10 values of n:



```java

        int m = 63;
        int n = 500;

        for(int i = 1; i<=10; i++) {
            if(args.length > 1)
                n = Integer.parseInt(args[1]);
            double meanDistance = randomWalkMulti(m, n);
            System.out.println(m + " steps: " + meanDistance + " over " + n + " experiments");
            m = m + 5;
        }
    }
```

```
/Users/chamanbetrabet/Library/Java/JavaVirtualMachines/openjdk-18.0.2.1/Contents/Home/bin/java ...
63 steps: 6.9579341058689455 over 500 experiments
68 steps: 7.4572691612201485 over 500 experiments
73 steps: 7.6734036931317355 over 500 experiments
78 steps: 7.649470583156858 over 500 experiments
83 steps: 8.011940669638651 over 500 experiments
88 steps: 8.537453134547007 over 500 experiments
93 steps: 8.781529802421488 over 500 experiments
98 steps: 9.007184618115595 over 500 experiments
103 steps: 8.83692836598779 over 500 experiments
108 steps: 9.270836038159786 over 500 experiments

Process finished with exit code 0
```

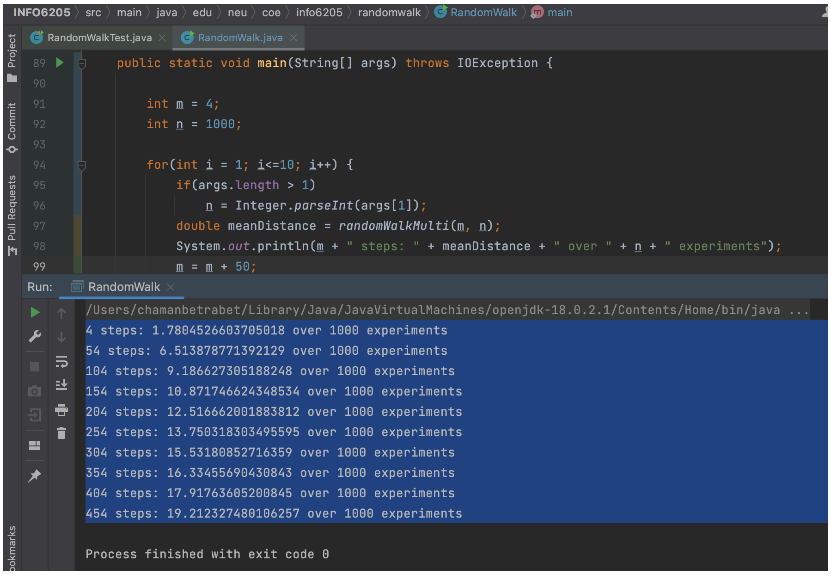
2. Experiment 2: Number of experiments = 1000 for 20 values of n:



```java
            m = m + 100;
        }
    }

}

```

```
/Users/chamanbetrabet/Library/Java/JavaVirtualMachines/openjdk-18.0.2.1/Contents/Home/bin/java ...
50 steps: 6.328363615929571 over 1000 experiments
150 steps: 10.547978336384341 over 1000 experiments
250 steps: 14.281714923861502 over 1000 experiments
350 steps: 16.50957909600833 over 1000 experiments
450 steps: 19.08550201491489 over 1000 experiments
550 steps: 20.97687402274332 over 1000 experiments
650 steps: 22.733807474618438 over 1000 experiments
750 steps: 24.297354646198084 over 1000 experiments
850 steps: 26.027288367678747 over 1000 experiments
950 steps: 26.660383458161842 over 1000 experiments
1050 steps: 29.488450945634654 over 1000 experiments
1150 steps: 29.47013465283329 over 1000 experiments
1250 steps: 31.723288247268357 over 1000 experiments
1350 steps: 32.24118033812824 over 1000 experiments
1450 steps: 33.658980100245444 over 1000 experiments
1550 steps: 35.291383049617245 over 1000 experiments
1650 steps: 34.960596324805884 over 1000 experiments
1750 steps: 36.53796111441149 over 1000 experiments
1850 steps: 37.46128880827672 over 1000 experiments
1950 steps: 39.40468943425283 over 1000 experiments
```

3. Experiment 3: Number of experiments = 1000 for 10 values of n:



INFO6205 〉 src 〉 main 〉 java 〉 edu 〉 neu 〉 coe 〉 info6205 〉 randomwalk 〉 RandomWalk 〉 main
RandomWalkTest.java
RandomWalk.java
Project
Commit
Pull Requests
Run:
RandomWalk
ookmarks

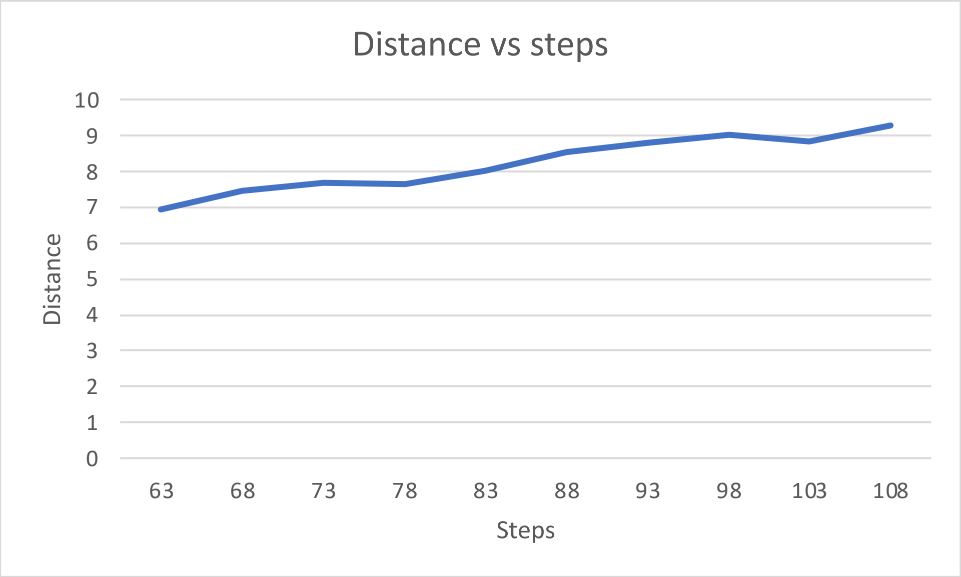
```java
    public static void main(String[] args) throws IOException {

        int m = 4;
        int n = 1000;

        for(int i = 1; i<=10; i++) {
            if(args.length > 1)
                n = Integer.parseInt(args[1]);
            double meanDistance = randomWalkMulti(m, n);
            System.out.println(m + " steps: " + meanDistance + " over " + n + " experiments");
            m = m + 50;
```

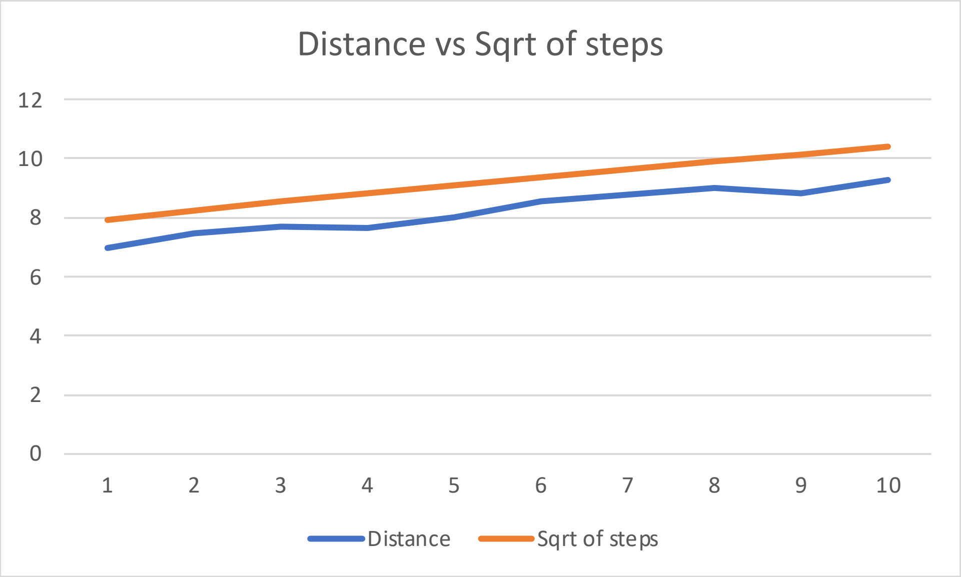
```
/Users/chamanbetrabet/Library/Java/JavaVirtualMachines/openjdk-18.0.2.1/Contents/Home/bin/java ...
4 steps: 1.7804526603705018 over 1000 experiments
54 steps: 6.513878771392129 over 1000 experiments
104 steps: 9.186627305188248 over 1000 experiments
154 steps: 10.871746624348534 over 1000 experiments
204 steps: 12.516662001883812 over 1000 experiments
254 steps: 13.750318303495595 over 1000 experiments
304 steps: 15.53180852716359 over 1000 experiments
354 steps: 16.33455690430843 over 1000 experiments
404 steps: 17.91763605200845 over 1000 experiments
454 steps: 19.212327480106257 over 1000 experiments

Process finished with exit code 0
```

- **Graphical representation:** In order to test different values of no of steps taken (m) and observe the relation between steps and distance (d), few changes have been done to the Randomwalk.java file which has been attached along with the submission for reference. Below are the results:

  1. Experiment 1:
     Number of experiments: 500
     10 different values of N

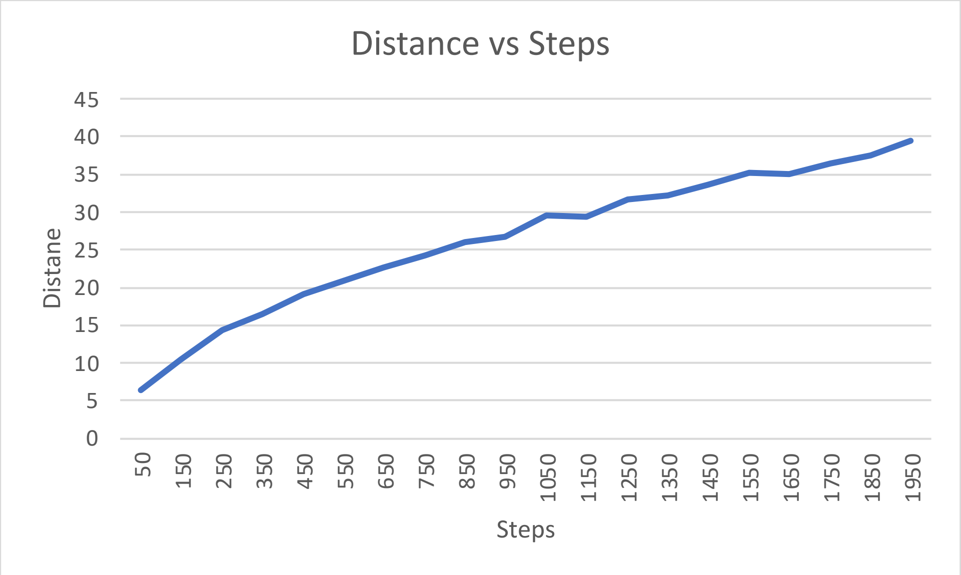
Steps	Distance	Sqrt of steps
63	6.95793411	7.937253933
68	7.45726916	8.246211251
73	7.67340369	8.544003745
78	7.64947058	8.831760866
83	8.01194067	9.110433579
88	8.53745313	9.38083152
93	8.7815298	9.643650761
98	9.00718462	9.899494937
103	8.83692837	10.14889157
108	9.27083604	10.39230485

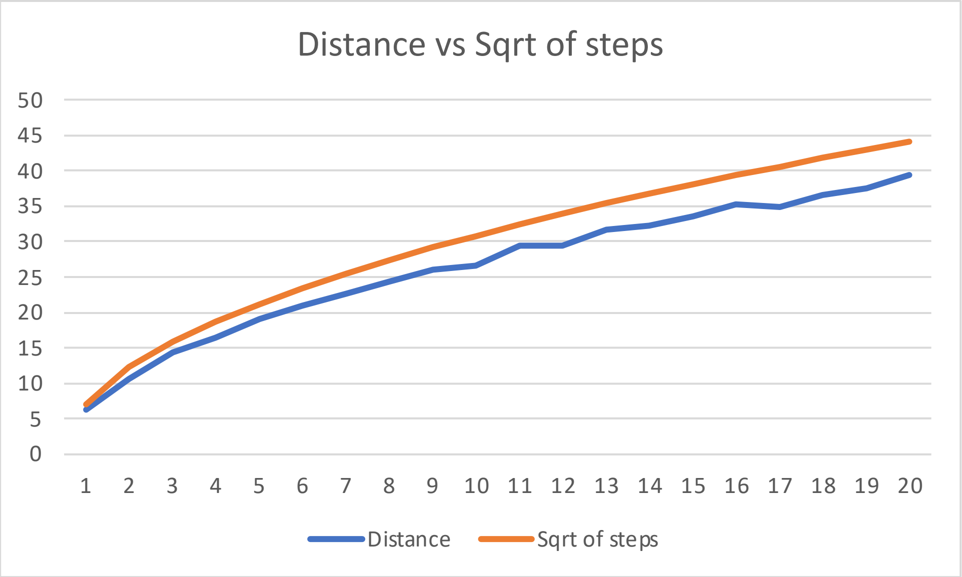

Distance vs steps
Distance
10
9
8
7
6
5
4
3
2
1
0
63
68
73
78
83
88
93
98
103
108
Steps


Distance vs Sqrt of steps
12
10
8
6
4
2
0
1
2
3
4
5
6
7
8
9
10
Distance
Sqrt of steps

2. Experiment 2:
   Number of experiments: 100
   20 different values of N

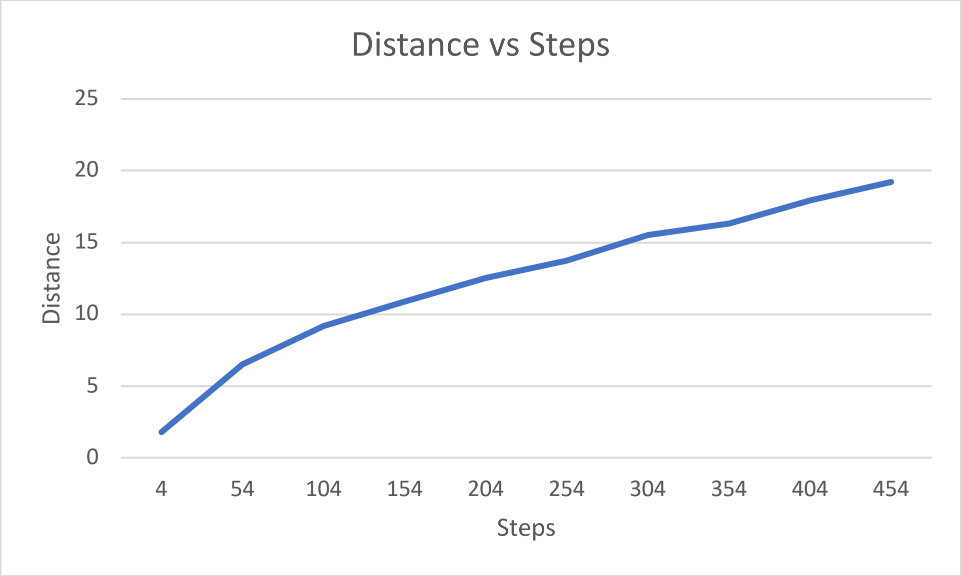
Steps	Distance	Sqrt of steps
50	6.32836362	7.07106781
150	10.5479783	12.2474487
250	14.2817149	15.8113883
350	16.5095791	18.7082869
450	19.085502	21.2132034
550	20.976874	23.4520788
650	22.7338075	25.4950976
750	24.2973546	27.3861279
850	26.0272884	29.1547595
950	26.6603835	30.82207
1050	29.4884509	32.4037035
1150	29.4701347	33.9116499
1250	31.7232882	35.3553391
1350	32.2411803	36.7423461
1450	33.6589801	38.0788655
1550	35.291383	39.3700394
1650	34.9605963	40.620192
1750	36.5379611	41.8330013
1850	37.4612888	43.0116263
1950	39.4046894	44.1588043

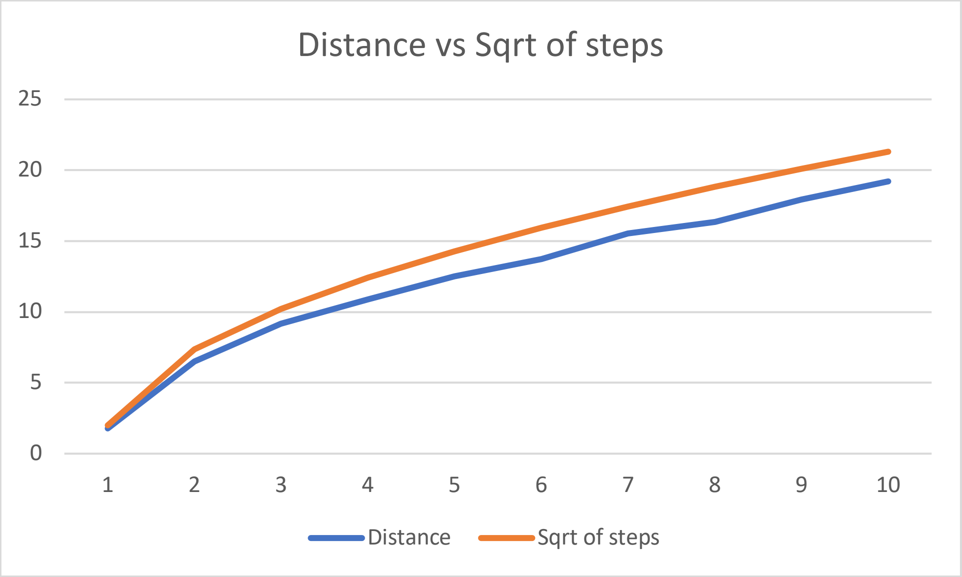



Distance vs Sqrt of steps
50
45
40
35
30
25
20
15
10
5
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Distance
Sqrt of steps

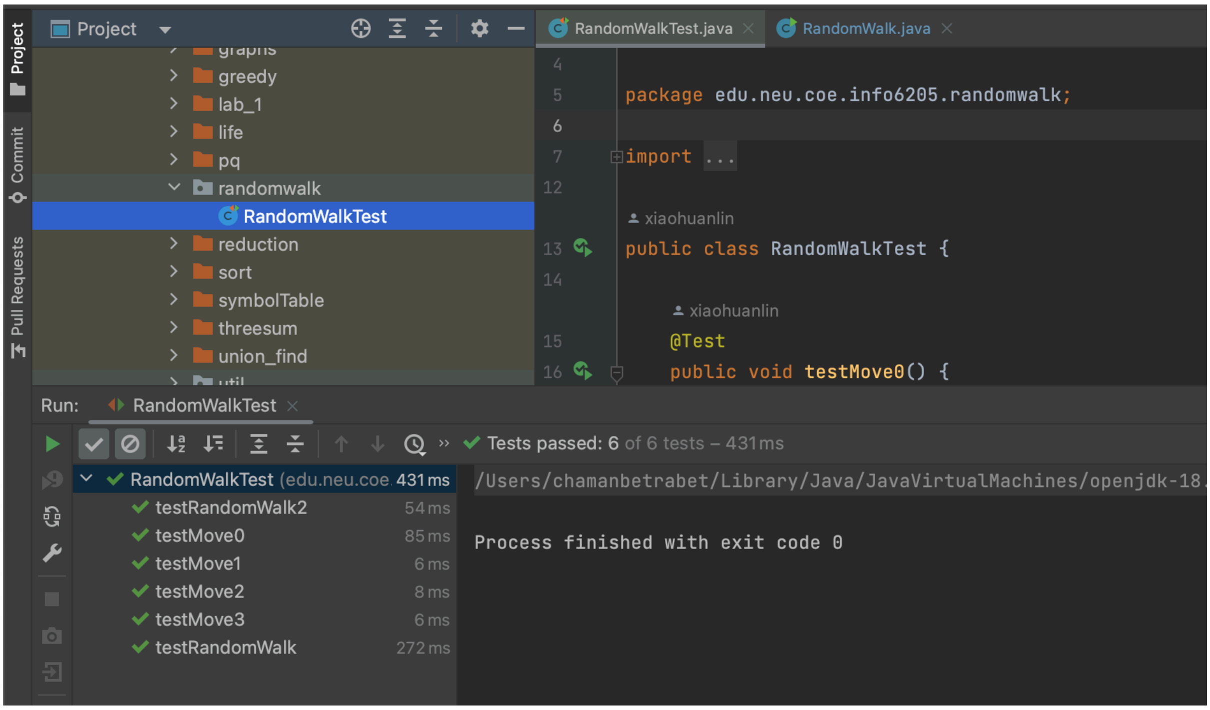
3. Experiment 3:
   Number of experiments: 100
   20 different values of N

Steps	Distance	Sqrt of steps
4	1.78045266	2
54	6.51387877	7.34846923
104	9.18662731	10.198039
154	10.8717466	12.4096736
204	12.516662	14.2828569
254	13.7503183	15.9373775
304	15.5318085	17.4355958
354	16.3345569	18.8148877
404	17.9176361	20.0997512
454	19.2123275	21.3072758





Unit test: Unit test results are as shown below:



```
package edu.neu.coe.info6205.randomwalk;

import ...

xiaohuanlin
public class RandomWalkTest {

    xiaohuanlin
    @Test
    public void testMove0() {
```

Tests passed: 6 of 6 tests – 431 ms

```
RandomWalkTest (edu.neu.coe 431 ms
  testRandomWalk2          54 ms
  testMove0                85 ms
  testMove1                 6 ms
  testMove2                 8 ms
  testMove3                 6 ms
  testRandomWalk          272 ms
```

```
/Users/chamanbetrabet/Library/Java/JavaVirtualMachines/openjdk-18

Process finished with exit code 0
```