Program Structures and Algorithms (INFO 6205) Summer 2022
NUID-002959612

Assignment 1-Random Walk

Tasks:

- Find the distance between origin and the destination of the drunk man who takes a series of steps of the same length 1 meter. The direction of these steps is randomly chosen from North, South, East, or West.
- Fill the missing code
- Find the relationship between d and n (Evidence to support that relationship)
- Screen shot of the unit tests all passing

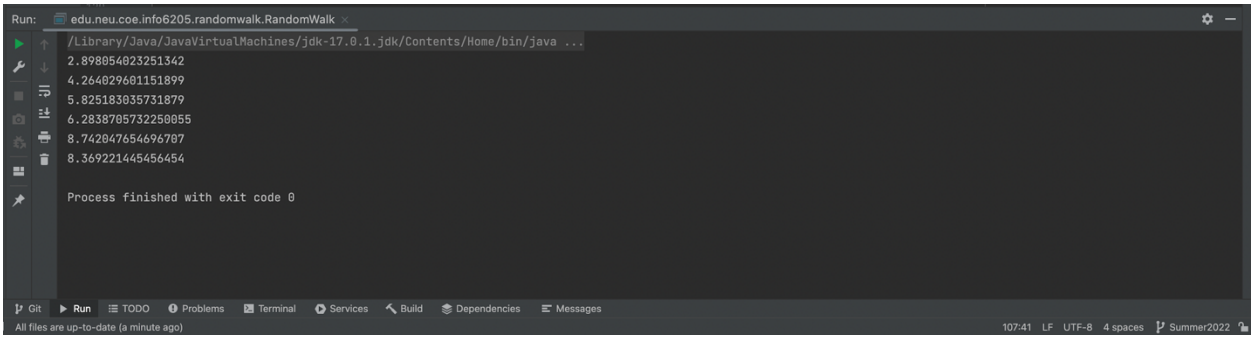
How d varies with n?

d- It is the Euclidean Distance of the man from the lamp-post.
N- It is the number of steps.
N values taken for the experiment are [10,20,40,60,80,100] and each experiment is run 30 times.

Relationship Conclusion

D = SQRT(N) + value (uncertain)
We can conclude that D is equal to the square root of the number of steps taken by the drunk man and (plus or minus) an uncertain value.

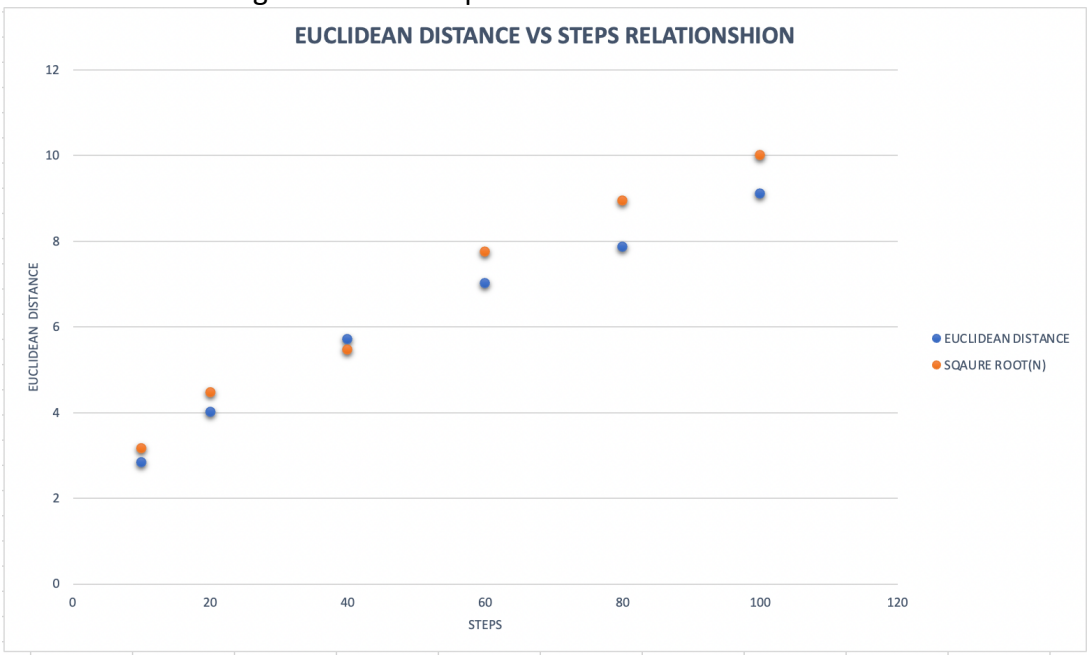
Sample Output:



STEPS AND DISTANCES (RUN 10 TIMES OVER DIFFERENT RANDOM VALUES)
Evidence to support the relationship conclusion given above.

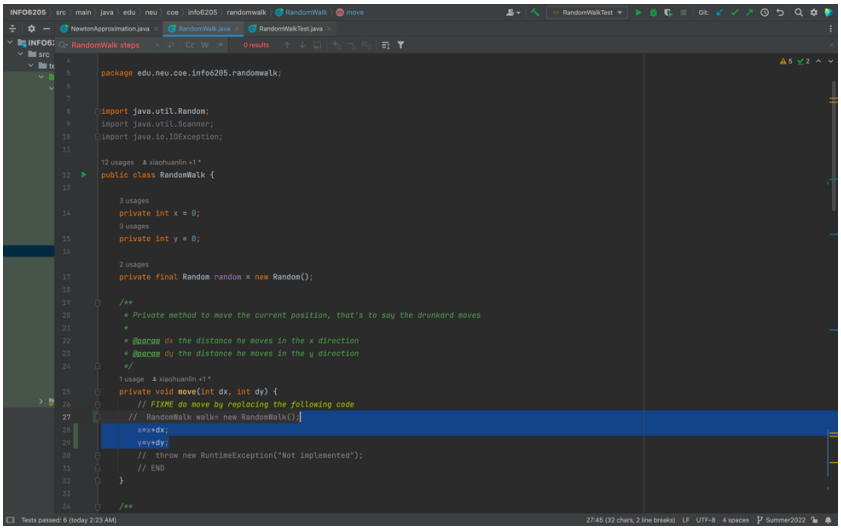
Steps	Cycles	Euclidean d1	Euclidean d2	Euclidean d3	Euclidean d4	Euclidean d5		
10	30	3.099622542	2.898054023	3.019621543	2.5477328	3.165573627		
20	30	3.939096292	4.264029601	4.548163301	4.275434853	4.035686087		
40	30	6.025669708	5.825183036	5.441492937	5.54916118	5.196864348		
60	30	7.727427487	6.283870573	7.388004497	5.707405826	7.625510459		
80	30	8.255620969	8.742047655	7.747258414	7.161248654	7.267029743		
100	30	8.866979387	8.369221445	11.34697337	9.104792196	8.780477414		
Steps	Cycles	Euclidean d6	Euclidean d7	Euclidean d8	Euclidean d9	Euclidean d10	Average	SQRT(N)
10	30	2.644306011	2.717343365	2.418926517	3.078280704	2.764354565	2.83538	3.16
20	30	3.638269901	4.189548749	3.477408686	3.812270561	3.986244456	4.016615249	4.47
40	30	6.654849288	5.583436179	5.084190426	5.978391931	5.825531728	5.716477076	5.47
60	30	7.458038653	6.722234882	7.497974307	6.378466944	7.335232965	7.012416659	7.745
80	30	7.640238535	9.613004865	8.024220993	6.771200835	7.455827719	7.867769838	8.94
100	30	8.434106355	9.178138745	8.499741621	9.630806727	8.902925304	9.111416257	10

Scatter Plot showing the relationship between EUCLIDEAN DISTANCE AND NO OF STEPS

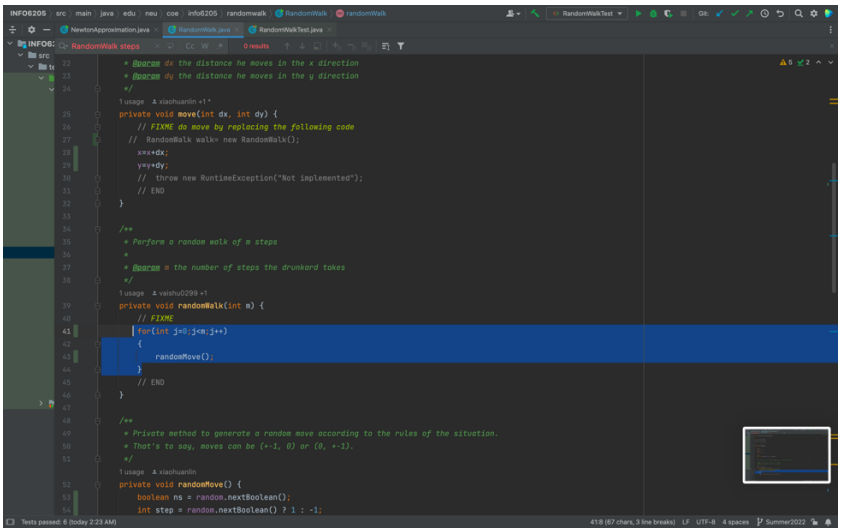


UNIT TEST CASES RESULTS

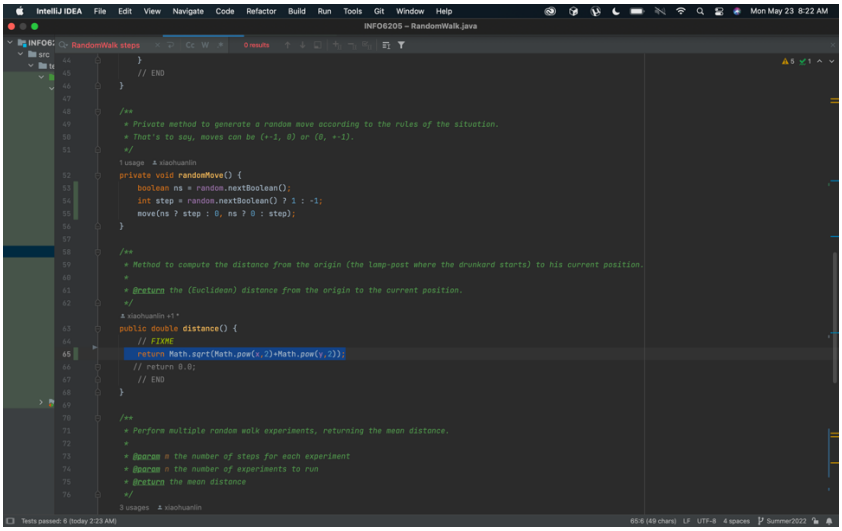
CODE



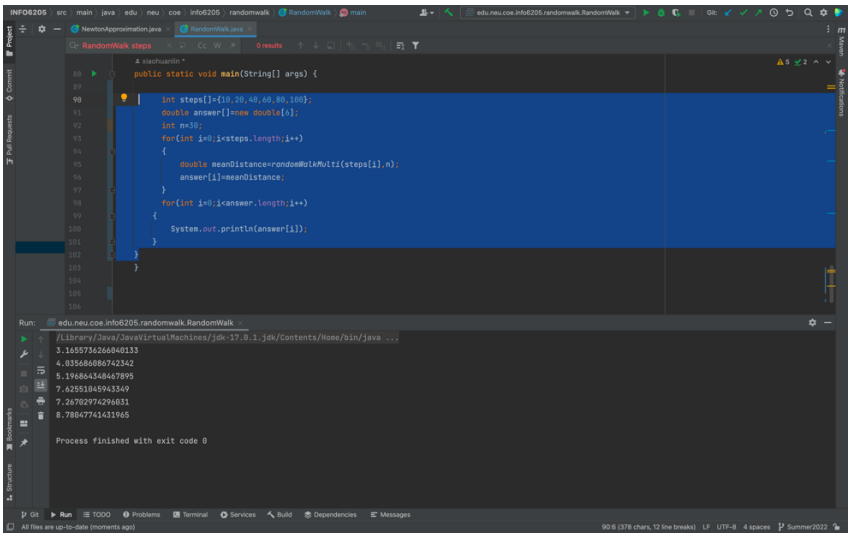
```java
package edu.neu.coe.info6205.randomwalk;


import java.util.Random;
import java.util.Scanner;
import java.io.IOException;

12 usages  xiaohuanlin +1 *
public class RandomWalk {

    3 usages
    private int x = 0;
    3 usages
    private int y = 0;

    2 usages
    private final Random random = new Random();

    /**
     * Private method to move the current position, that's to say the drunkard moves
     *
     * @param dx the distance he moves in the x direction
     * @param dy the distance he moves in the y direction
     */
    1 usage  xiaohuanlin +1 *
    private void move(int dx, int dy) {
        // FIXME do move by replacing the following code
      //  RandomWalk walk= new RandomWalk();
        x=x+dx;
        y=y+dy;
        //  throw new RuntimeException("Not implemented");
        // END
    }

    /**
```



```java
     * @param dx the distance he moves in the x direction
     * @param dy the distance he moves in the y direction
     */
    1 usage  xiaohuanlin +1 *
    private void move(int dx, int dy) {
        // FIXME do move by replacing the following code
      //  RandomWalk walk= new RandomWalk();
        x=x+dx;
        y=y+dy;
        //  throw new RuntimeException("Not implemented");
        // END
    }

    /**
     * Perform a random walk of m steps
     *
     * @param m the number of steps the drunkard takes
     */
    1 usage  vaishu0299 +1
    private void randomWalk(int m) {
        // FIXME
        for(int j=0;j<m;j++)
        {
            randomMove();
        }
        // END
    }

    /**
     * Private method to generate a random move according to the rules of the situation.
     * That's to say, moves can be (+-1, 0) or (0, +-1).
     */
    1 usage  xiaohuanlin
    private void randomMove() {
        boolean ns = random.nextBoolean();
        int step = random.nextBoolean() ? 1 : -1;
```



```java
        }
        // END
    }

    /**
     * Private method to generate a random move according to the rules of the situation.
     * That's to say, moves can be (+-1, 0) or (0, +-1).
     */
    1 usage  xiaohuanlin
    private void randomMove() {
        boolean ns = random.nextBoolean();
        int step = random.nextBoolean() ? 1 : -1;
        move(ns ? step : 0, ns ? 0 : step);
    }

    /**
     * Method to compute the distance from the origin (the lamp-post where the drunkard starts) to his current position.
     *
     * @return the (Euclidean) distance from the origin to the current position.
     */
    xiaohuanlin +1 *
    public double distance() {
        // FIXME
        return Math.sqrt(Math.pow(x,2)+Math.pow(y,2));
        // return 0.0;
        // END
    }

    /**
     * Perform multiple random walk experiments, returning the mean distance.
     *
     * @param m the number of steps for each experiment
     * @param n the number of experiments to run
     * @return the mean distance
     */
    3 usages  xiaohuanlin
```


INFO6205 src main java edu neu coe info6205 randomwalk RandomWalk main
edu.neu.coe.info6205.randomwalk.RandomWalk
Git:
NewtonApproximation.java
RandomWalk.java
RandomWalk steps
0 results
xiaohuanlin *
public static void main(String[] args) {
int steps[]={10,20,40,60,80,100};
double answer[]=new double[6];
int n=30;
for(int i=0;i<steps.length;i++)
{
double meanDistance=randomWalkMulti(steps[i],n);
answer[i]=meanDistance;
}
for(int i=0;i<answer.length;i++)
{
System.out.println(answer[i]);
}
}
}
Run:
edu.neu.coe.info6205.randomwalk.RandomWalk
/Library/Java/JavaVirtualMachines/jdk-17.0.1.jdk/Contents/Home/bin/java ...
3.1655736266040133
4.035686086742342
5.196864348467895
7.62551045943349
7.2670297429603 1
8.78047741431965
Process finished with exit code 0
Git
Run
TODO
Problems
Terminal
Services
Build
Dependencies
Messages
All files are up-to-date (moments ago)
90:6 (378 chars, 12 line breaks)
LF
UTF-8
4 spaces
Summer2022
Project
Commit
Pull Requests
Bookmarks
Structure
Maven
Notifications