Program Structure and Algorithm

Assignment 1 Random Walk

Shouting Lyu

001822094

Following the instruction of the lecture, I designed my experiments and implemented it in my code. First, after many tries, I found that the final positions of the program vacillate unstably (Figure 1). Hence, from my perspective, it is more reasonable to use the average distance calculated from the whole test for a certain n while analyzing the results. Generally, I did several tests for different n and L. In details, to optimize the results, I run 5 times for each certain value of n (10, 20, 40, 80, 160, 320, 640, 1280, 2560, 5120)In addition, I repeated running the program with different L(1, 2, 4, 8, 16).

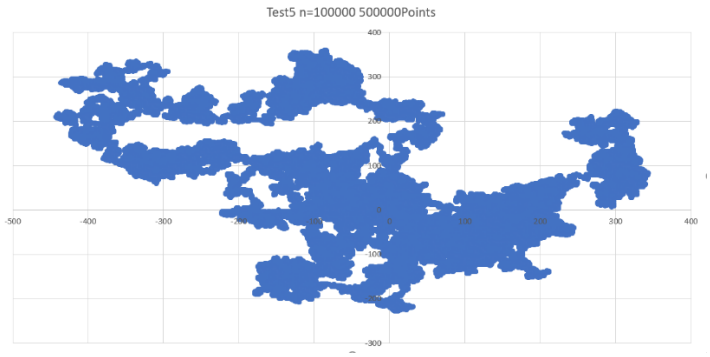

Figure 1

Then, the graphs below clearly show the relationships between n, L and distance. As shown in the Figure 2, the x axis represents times which means the value of n, and the y axis represents the average distance between the current place to the (0,0) and they are both under by logarithm transformation.

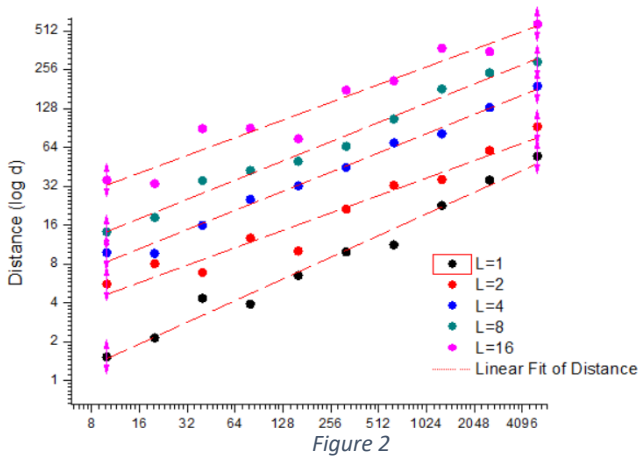

Figure 2

L	1	2	4	8	16
Slope	0.55	0.448	0.493	0.494	0.455
Intersection	-1.288	0.726	1.415	2.185	3.517
R-square	0.978	0.941	0.989	0.982	0.933
$2^{\text{intersection}}$	0.712	1.654	2.666	4.547	11.447

Table 1

Since the average slope is 0.488 and the intersection of the linear fit increases when L increases, the relationship should be $d = f(L) * N^{0.488}$. Then, calculating the $2^{intersection}$, we could get the f(L)=0.697L - 0.121. Because the intersection of linear fit of f(L) is small, we could make a theory that f(L) approximately monotonically increase with L.

In conclusion, the final relationship should be $d = (0.697L-0.121) * N^{0.488}$, if we make an approximation of it, then the relationship is $d = 0.7L * N^{0.5}$.

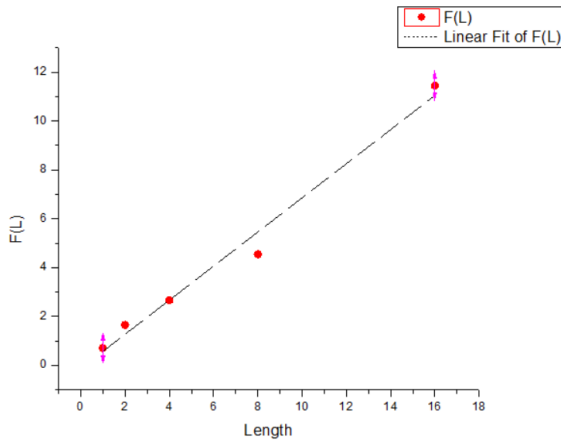

Figure 3

Equation	y = a + b		
Adj. R-Squa	0.97898		
		Value	Standard Err
F(L)	Intercept	-0.1219	0.42115
F(L)	Slope	0.6979	0.051

Table 2

Finally, there are some evidence of passing the Junit tests shown below.

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

    ArrayList<ArrayList<Double[]>> testList = new ArrayList<>();
    //JUnit Test
    RandomWalkTest.testMove1();
    RandomWalkTest.testMove2();
    RandomWalkTest.testMove3();
    RandomWalkTest.testRandomWalk();
    // Writing Result
    int times = 10;
    for (int i = 0; i < 10; i++) {
        System.out.println("Test" + i + ": n=" + times + "\n");
        for (int j = 0; j < 5; j++) {
            RandomWalk randomTest = new RandomWalk();
            try {
                File file = new File("d:\\test_" +(i+1)+ "_n_" +times+".txt");
                FileWriter out = new FileWriter(file,true);
                for (Double[] test1 : randomTest.randomWalk(times, 16)) {
                    out.write(test1[0] + " ");
                    out.write(test1[1] + " ");
                    out.write(test1[2] + "\n");
                }
                out.close();
            } catch (IOException o) {
                o.getStackTrace();
            }
        }
        times = times*2;
    }
```

```
Output - Assignment1 (run) × Start Page × DisplayFormData.java ×
run:
Test0: n=10

Test1: n=20

Test2: n=40

Test3: n=80

Test4: n=160

Test5: n=320

Test6: n=640

Test7: n=1280

Test8: n=2560

Test9: n=5120

BUILD SUCCESSFUL (total time: 1 second)
```