Student Name: **Hongyao Tao (001067209)**

INFO 6205
Program Structures & Algorithms
Spring 2021

Assignment No.1

- The relationship between *d* and *n*

d=n^0.4903

- **Evidence** to support that relationship

	A	B
1	steps	distance
2	0	0
3	50	5.750949
4	100	6.64702
5	150	10.54341
6	200	13.8183
7	250	12.91865
8	300	15.49494
9	350	17.94477
10	400	19.01446
11	450	19.94513
12	500	24.15901
13	550	21.8919
14	600	22.87556
15	650	18.49034
16	700	25.10559
17	750	21.70605
18	800	24.4073

	A	B
9984	499100	694.4834
9985	499150	637.4093
9986	499200	517.619
9987	499250	622.3012
9988	499300	630.264
9989	499350	519.0807
9990	499400	549.6287
9991	499450	601.1182
9992	499500	608.3727
9993	499550	672.5997
9994	499600	604.7464
9995	499650	662.4405
9996	499700	689.3644
9997	499750	678.0625
9998	499800	628.7192
9999	499850	615.8336
10000	499900	663.2058
10001	499950	659.7088

Figure 1. The data of steps and distance from random walk simulation

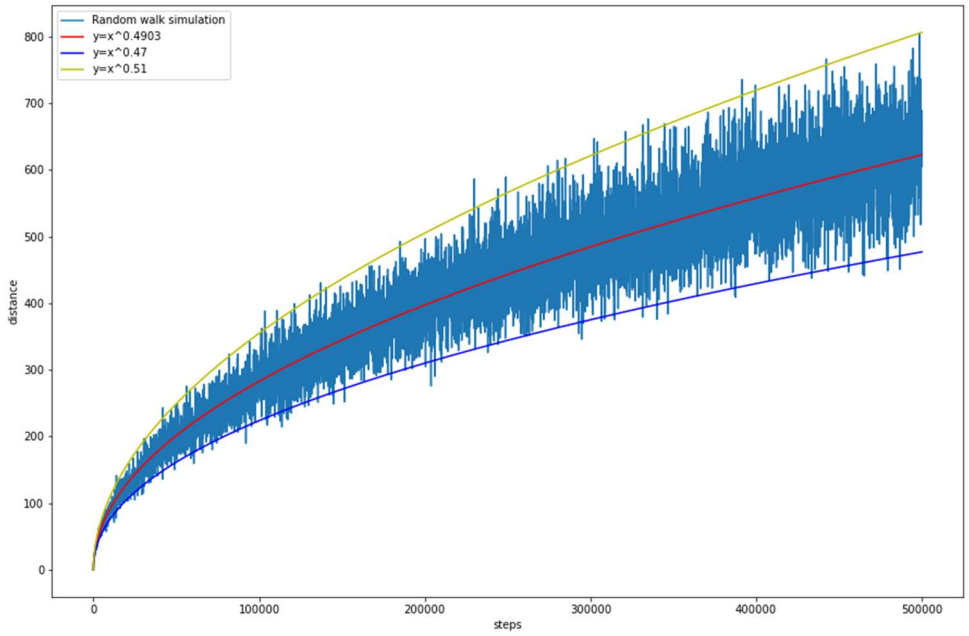

Figure 2. The line graph of random walk simulation

- *RandomWalk.java*

```
/**
 * 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
 */
private void move(int dx, int dy) {
    x+=dx;
    y+=dy;
}

/**
 * Perform a random walk of m steps
 *
 * @param m the number of steps the drunkard takes
 */
private void randomWalk(int m) {
    for (int i=0;i<m;i++) {
        randomMove();
    }
}
```

```
 */
public double distance() {
    double distance=Math.sqrt(x*x+y*y);
    return distance;
}


public static void main(String[] args) {
     List<Integer> list5=new ArrayList<Integer>();
     List<Integer> list2=new ArrayList<Integer>();
   List<Integer> list1=new ArrayList<Integer>();
   for (int i=0;i<500000;i+=50){
       list1.add(i);
   }
   writeToCSV(list1, testsCount: 30);

     for(int i=1;i<10;i++){
         list5.add((int)Math.pow(5,i));

     }
     for(int i=1;i<20;i++){
         list2.add((int)Math.pow(2,i));
     }

     System.out.println("--------5 times--------");
     walkFromList(list5,30);
     System.out.println("--------2 times--------");
     walkFromList(list2,30);
     System.out.println("--------1 times--------");
     walkFromList(list1,30);
}
```

```java
public static void walkFromList(List<Integer> steps,int testsCount){
    for (int i=0;i<steps.size();i++){
        int n = testsCount;
        int m=steps.get(i);
        double meanDistance = randomWalkMulti(m, n);
        System.out.println(m + "," + meanDistance);
          System.out.println(m + " steps: " + meanDistance + " over " + n + " experiments");
    }
}
public static void writeToCSV(List<Integer> steps,int testsCount){
    PrintWriter pw = null;
    try {
        pw = new PrintWriter(new File( pathname: "StepsAndDistance.csv"));
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
    StringBuilder header = new StringBuilder();
    String columnNamesList = "steps,distance";
    // No need give the headers Like: id, Name on builder.append
    header.append(columnNamesList +"\n");
    pw.write(header.toString());

    for (int i=0;i<steps.size();i++){
        int n = testsCount;
        int m=steps.get(i);
        double meanDistance = randomWalkMulti(m, n);
          System.out.println(m + " steps: " + meanDistance + " over " + n + " experiments");
        StringBuilder builder = new StringBuilder();
        builder.append(m+",");
        builder.append(meanDistance);
        builder.append('\n');
        if(i%100==0) System.out.println(m + " steps: " + meanDistance + " over " + n + " experiments");
        pw.write(builder.toString());
    }
    pw.close();
}
```

- A **screen shot** of the unit tests all passing

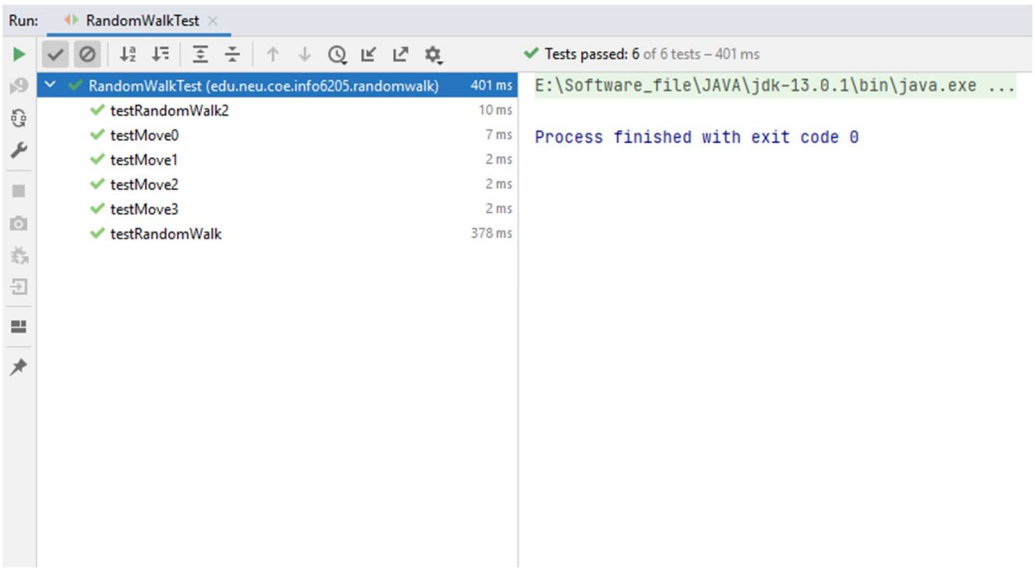

Figure 3. The unit test result

The result of RandomWalkTest showed that RandomWalk.java code is correct and passed all unites tests.