

1. The conclusion about relationship between d and n:

$$\text{func}(d): E(d^2)=n$$
2. The evidence to support relationship:
 Making 500 times experiments and find out that the mean value of sqrt of distance infinitely approach the number of steps (*RandomWalkProof.java*):

```

Run: RandomWalkProof x
/Library/Java/JavaVirtualMachines/jdk1.8.0_261.jdk/Contents/Home/bin/java ...
1steps: mean value of sqrt of distance = 1.0
2steps: mean value of sqrt of distance = 1.952
3steps: mean value of sqrt of distance = 3.064
4steps: mean value of sqrt of distance = 4.056
5steps: mean value of sqrt of distance = 4.968
6steps: mean value of sqrt of distance = 5.94
7steps: mean value of sqrt of distance = 7.056
8steps: mean value of sqrt of distance = 7.576
9steps: mean value of sqrt of distance = 8.648
10steps: mean value of sqrt of distance = 10.224
11steps: mean value of sqrt of distance = 10.24
12steps: mean value of sqrt of distance = 11.944
13steps: mean value of sqrt of distance = 12.176
14steps: mean value of sqrt of distance = 14.148
15steps: mean value of sqrt of distance = 15.152
16steps: mean value of sqrt of distance = 15.796
17steps: mean value of sqrt of distance = 17.88
18steps: mean value of sqrt of distance = 17.692
19steps: mean value of sqrt of distance = 18.888
20steps: mean value of sqrt of distance = 20.432

Process finished with exit code 0

```

3. I complete the code in *RandomWalk.java* and *RandomWalkProof.java*
4. Screen Shot of Unit Test:

```

Project
├── codeLength
├── coupling
├── dynamicProgramming.coins
├── equable
├── functions
├── graphs
├── greedy
├── hashtable
├── lab_1
├── life
├── pq
├── randomwalk
├── RandomWalk
├── RandomWalkProof
├── reduction
├── runLengthEncoding
├── sort
├── symbolTable
├── threesum
├── union_find
├── util
├── BinarySearch
├── CallByValue
├── ComparableTuple
├── Counter
├── RandomWalk.java
├── RandomWalkProof.java
├── RandomWalkTest.java
├── ...
└── ...

RandomWalkTest.java
1 //...
2
3 package edu.neu.coe.info6205.randomwalk;
4
5 import ...
6
7
8
9
10
11
12
13 public class RandomWalkTest {
14
15     @Test
16     public void testMove0() {
17         RandomWalk rw = new RandomWalk();
18         PrivateMethodTester pmt = new PrivateMethodTester(rw);
19         pmt.invokePrivate( name: "move", ...parameters: 1, 0);
20         assertEquals( expected: 1.0, rw.distance(), delta: 1.0E-7);
21     }
22
23     /**
24      *
25      */
26
27     @Test
28     public void testMove1() {
29         RandomWalk rw = new RandomWalk();
30         PrivateMethodTester pmt = new PrivateMethodTester(rw);
31         pmt.invokePrivate( name: "move", ...parameters: 1, 0);
32     }
33 }

Run: RandomWalkTest
Tests passed: 6 of 6 tests - 302 ms
RandomWalkTest (edu.neu.coe) 302 ms
├── testRandomWalk2 11 ms
├── testMove0 2 ms
├── testMove1 3 ms
├── testMove2 2 ms
├── testMove3 1 ms
└── testRandomWalk 283 ms

Process finished with exit code 0

```