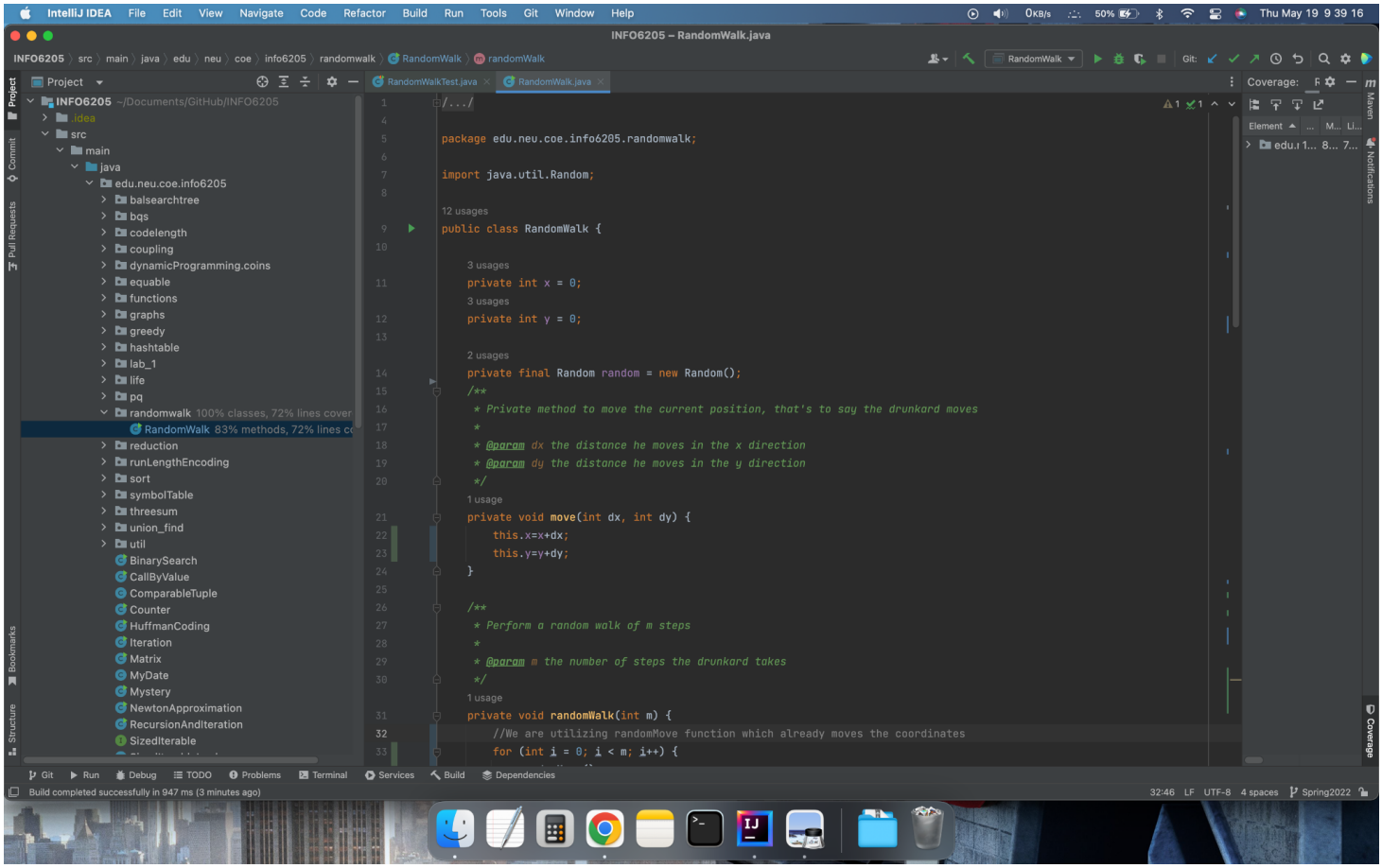
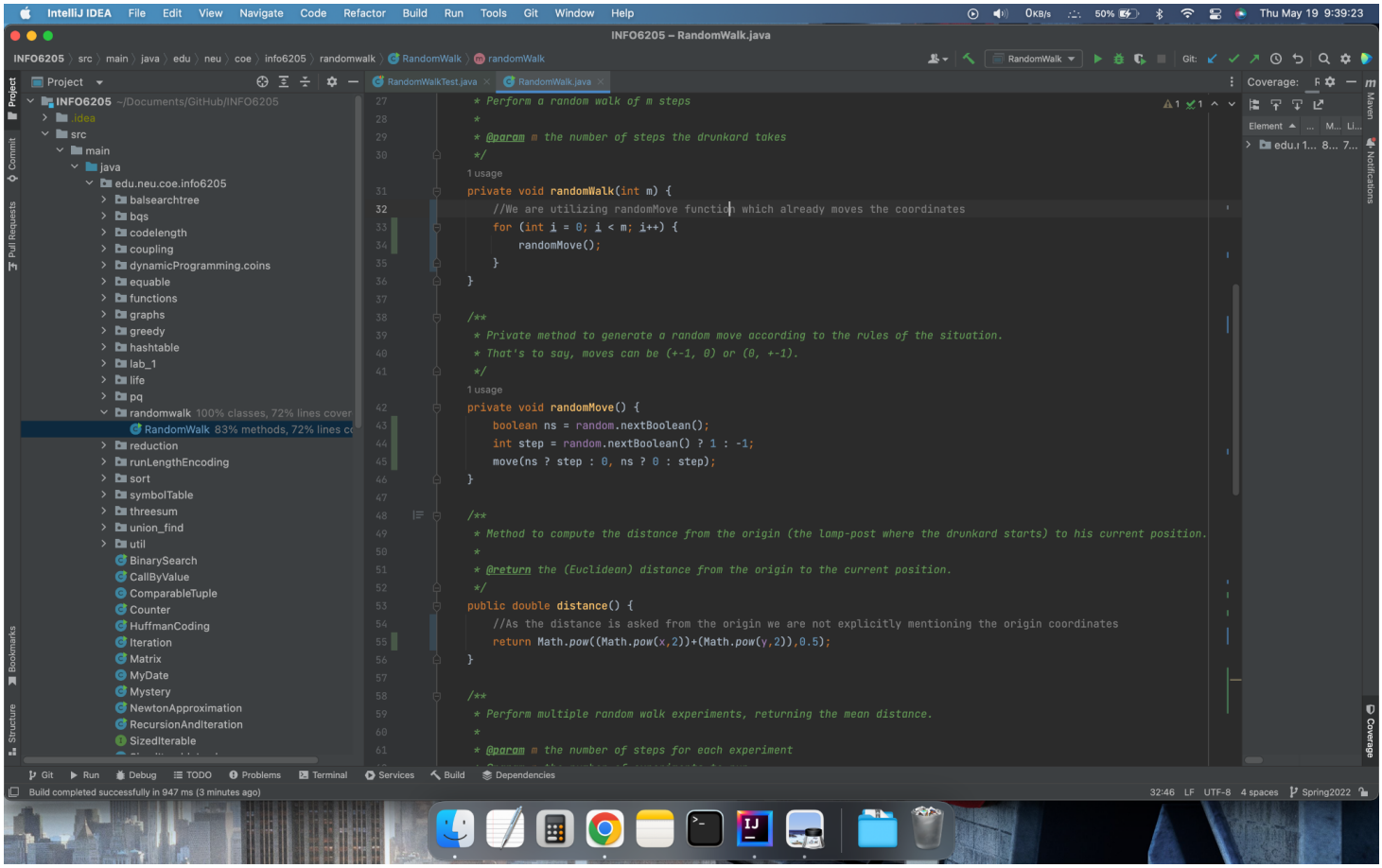
INFO6205 - Program Structures and Algorithms

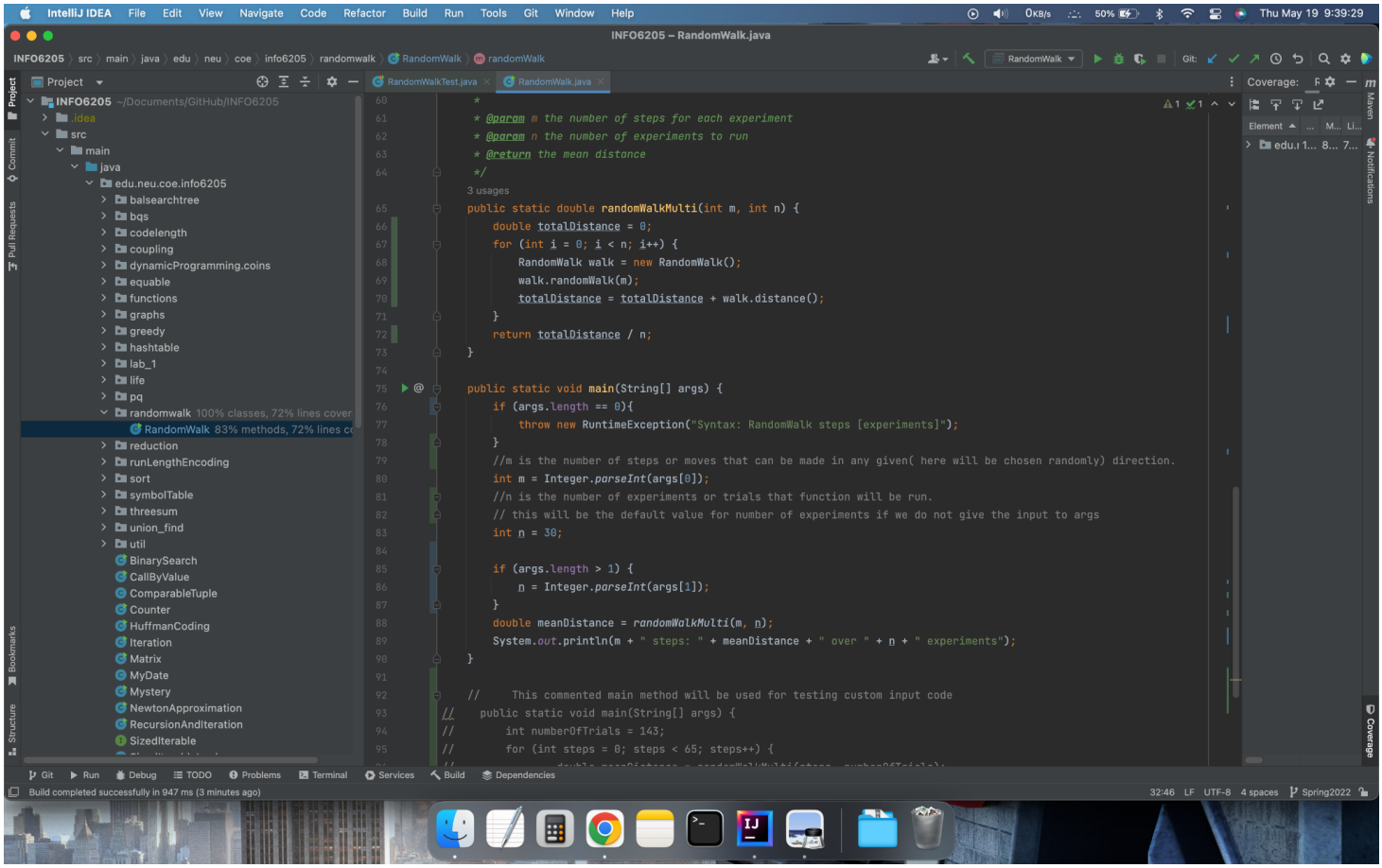
Assignment 01 - Random Walk Algorithm

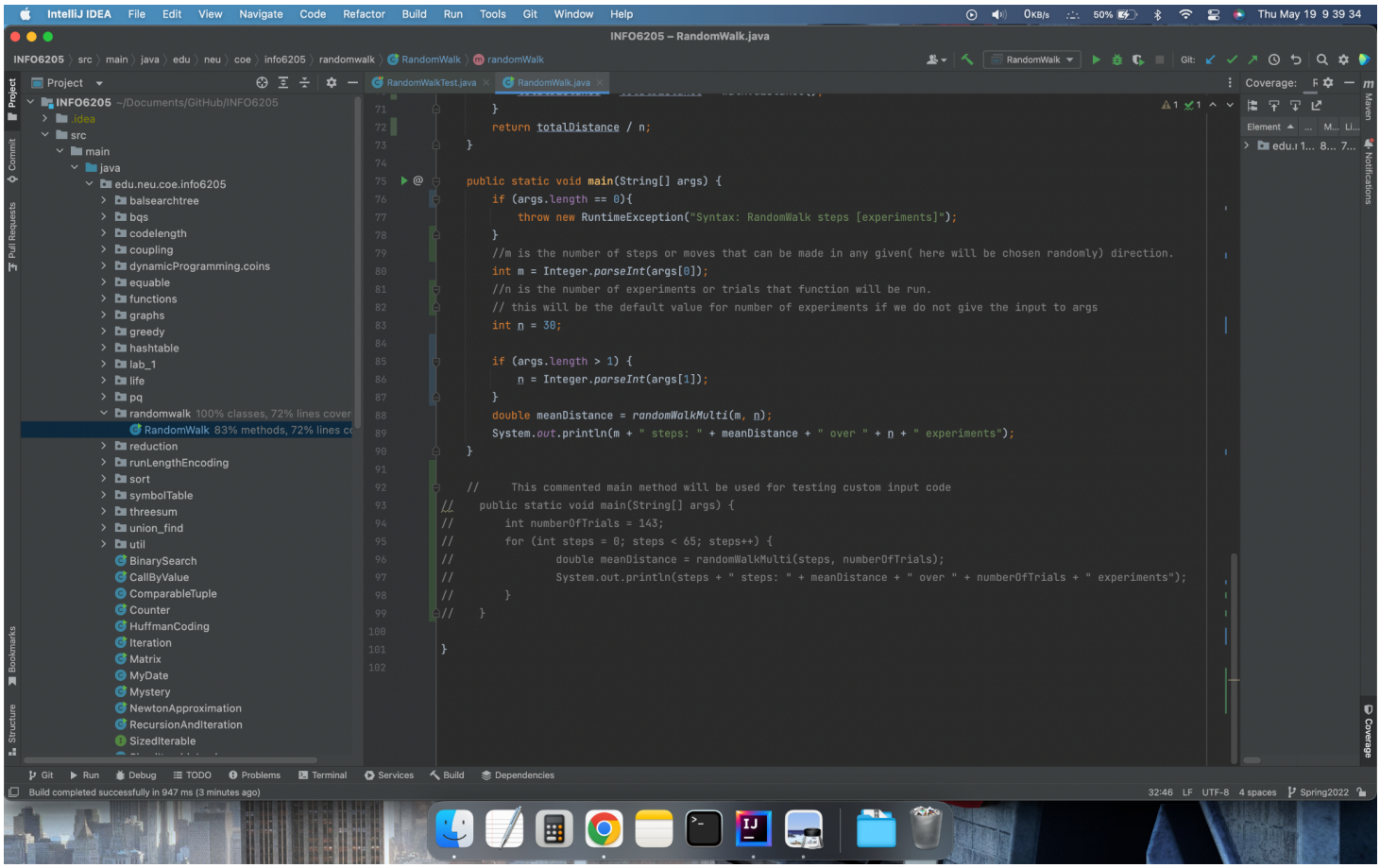
Implemented Functions marked as “FIXME” in RandomWalk.java class file

- **move(int dx,int dy)**
- **randomWalk(int m)**
- **distance()**
- **Main method in RandomWalk.java for custom test cases**

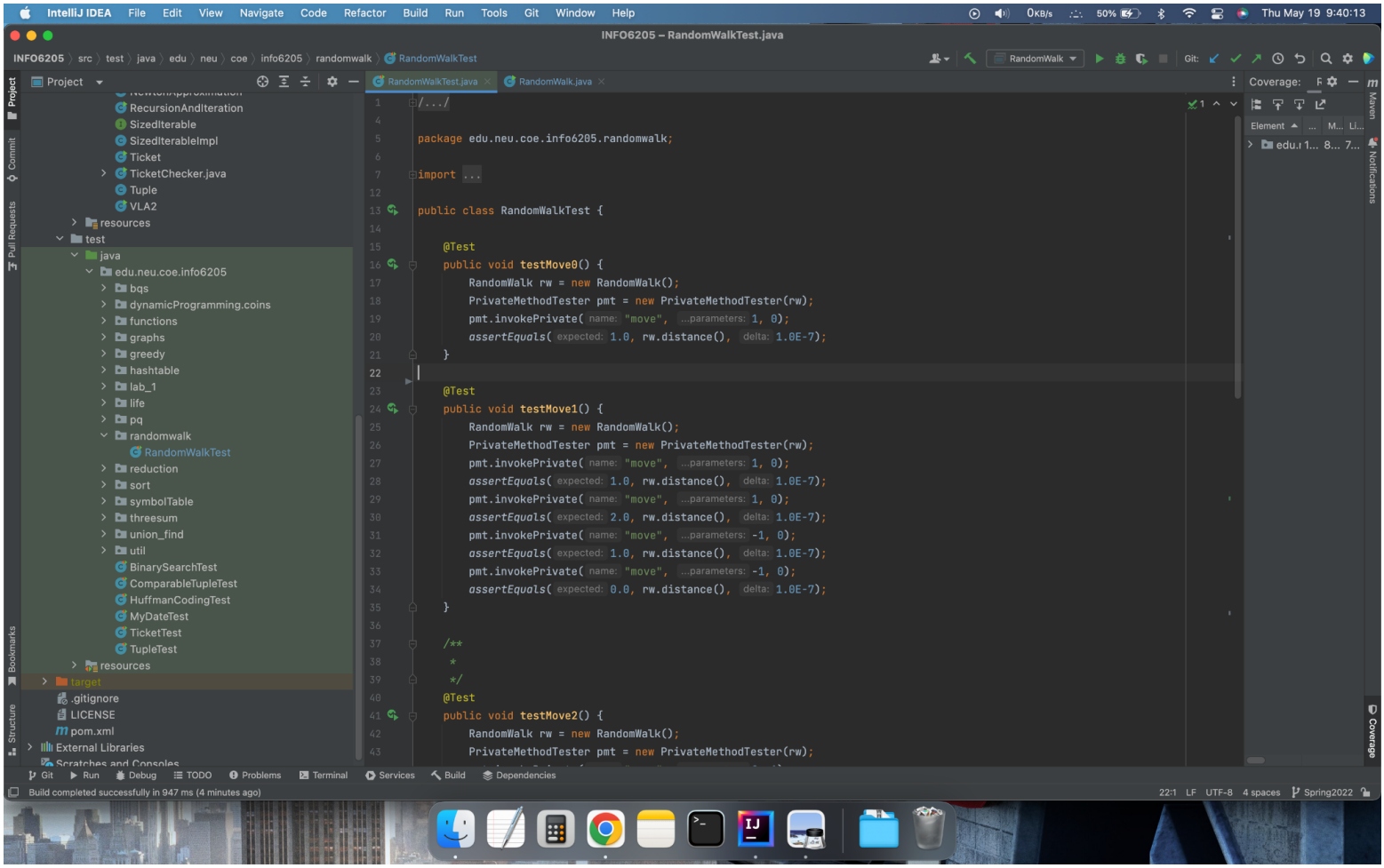


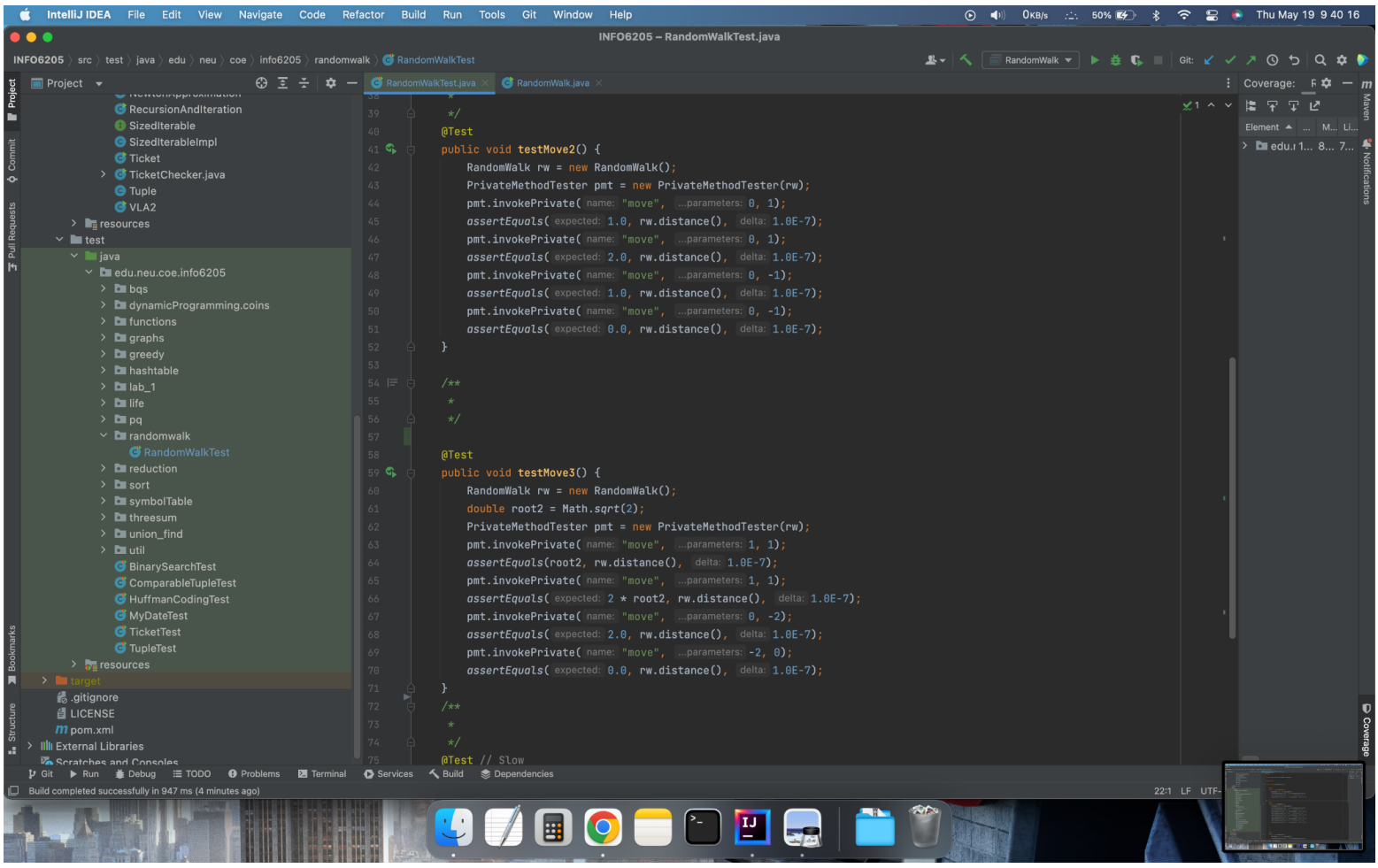

IntelliJ IDEA File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Thu May 19 9:39:23
INFO6205 – RandomWalk.java
INFO6205 src main java edu neu coe info6205 randomwalk RandomWalk randomWalk
RandomWalkTest.java
RandomWalk.java
Project
INFO6205 ~/Documents/GitHub/INFO6205
.idea
src
main
java
edu.neu.coe.info6205
balsearchtree
bqs
codelength
coupling
dynamicProgramming.coins
equable
functions
graphs
greedy
hashtable
lab_1
life
pq
randomwalk 100% classes, 72% lines cover
RandomWalk 83% methods, 72% lines co
reduction
runLengthEncoding
sort
symbolTable
threesum
union_find
util
BinarySearch
CallByValue
ComparableTuple
Counter
HuffmanCoding
Iteration
Matrix
MyDate
Mystery
NewtonApproximation
RecursionAndIteration
SizedIterable
* Perform a random walk of m steps
* @param m the number of steps the drunkard takes
1 usage
private void randomWalk(int m) {
//We are utilizing randomMove function which already moves the coordinates
for (int i = 0; i < m; i++) {
randomMove();
}
}
/**
* 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
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.
*/
public double distance() {
//As the distance is asked from the origin we are not explicitly mentioning the origin coordinates
return Math.pow((Math.pow(x,2))+(Math.pow(y,2)),0.5);
}
/**
* Perform multiple random walk experiments, returning the mean distance.
*
* @param m the number of steps for each experiment
Coverage:
Element
edu.n 1... 8... 7...
Git Run Debug TODO Problems Terminal Services Build Dependencies
Build completed successfully in 947 ms (3 minutes ago)
32:46 LF UTF-8 4 spaces Spring2022



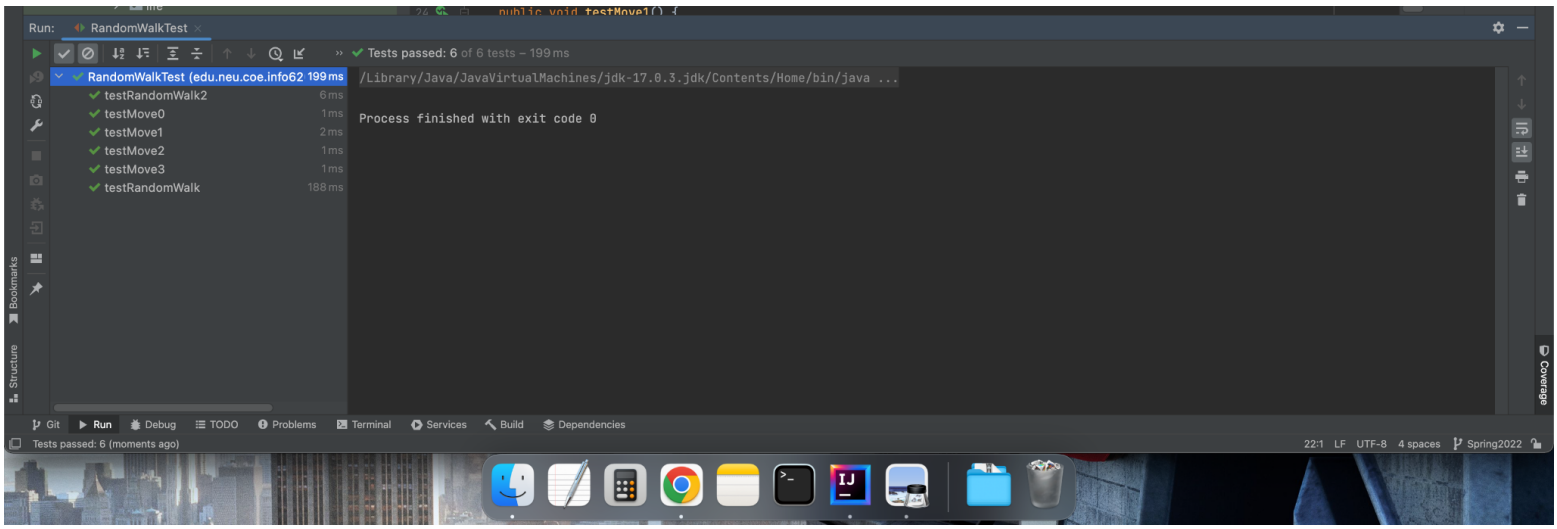


All Test Cases Are Running for the implemented code - Attached > Test Cases + Test Results

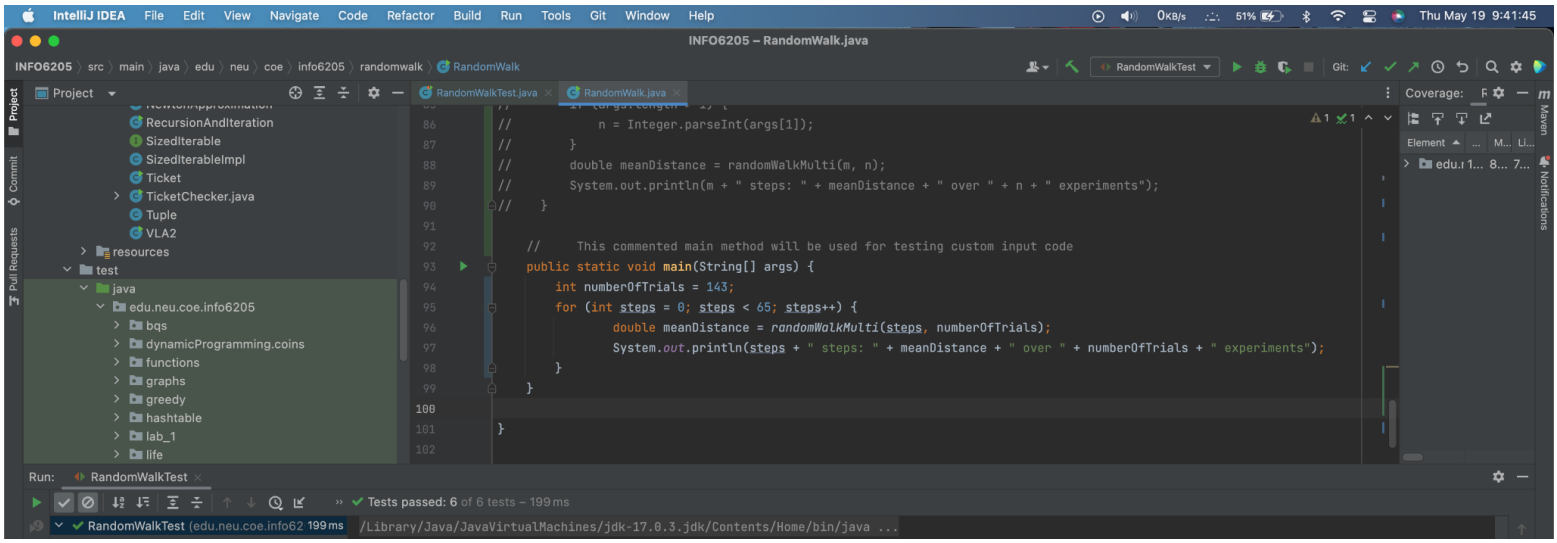




All Test Cases were passed

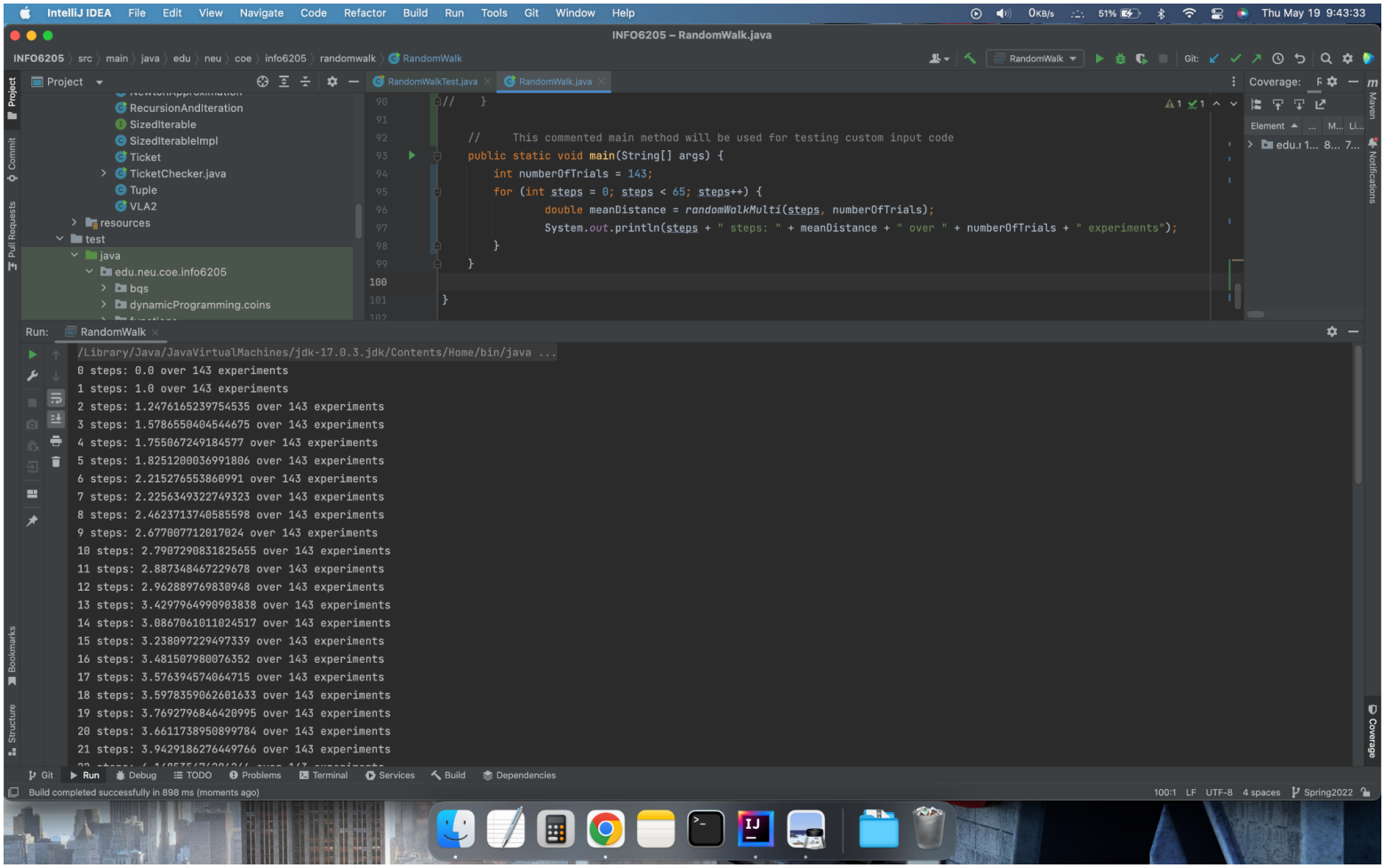


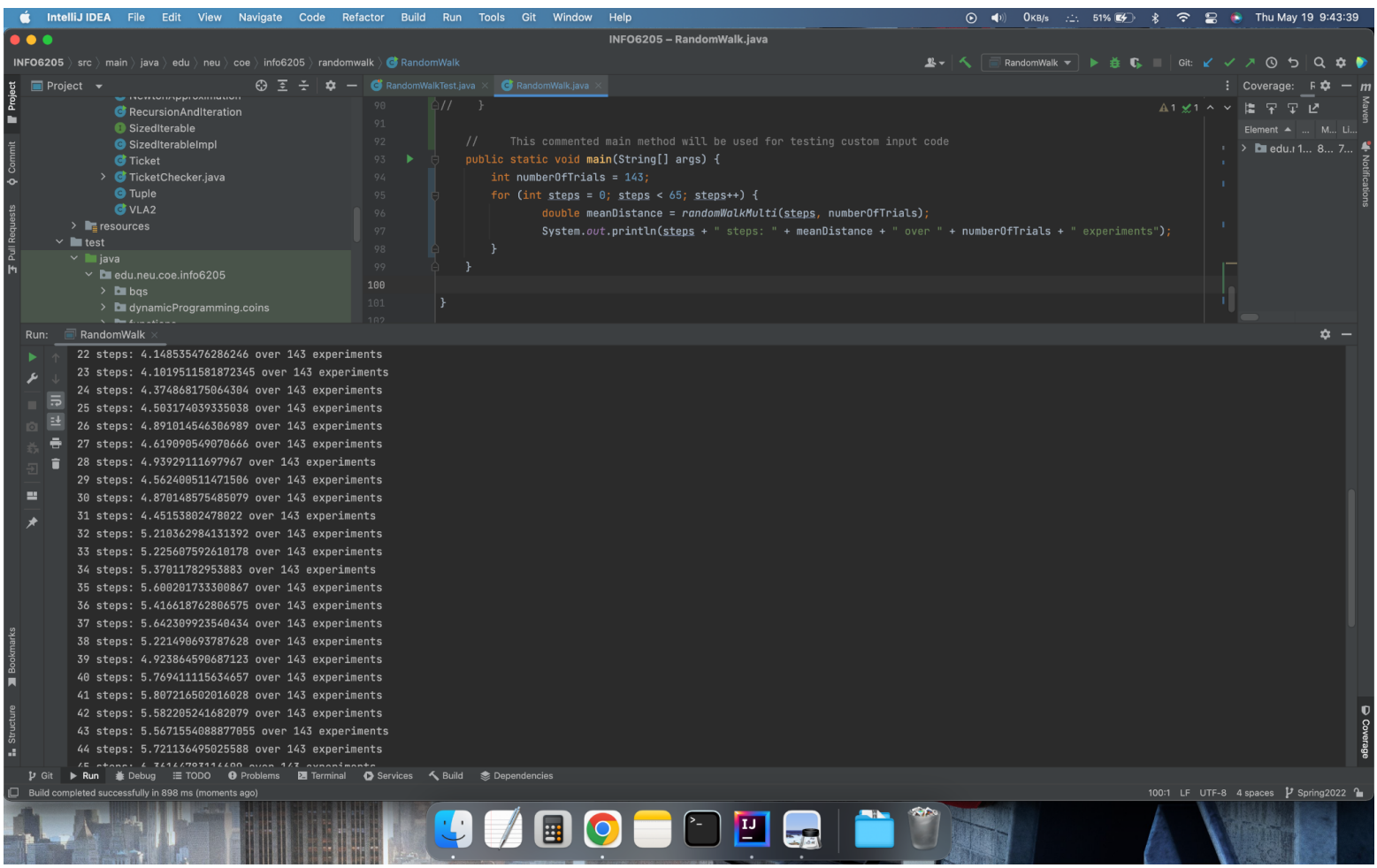
Modified Main Method to run custom test cases

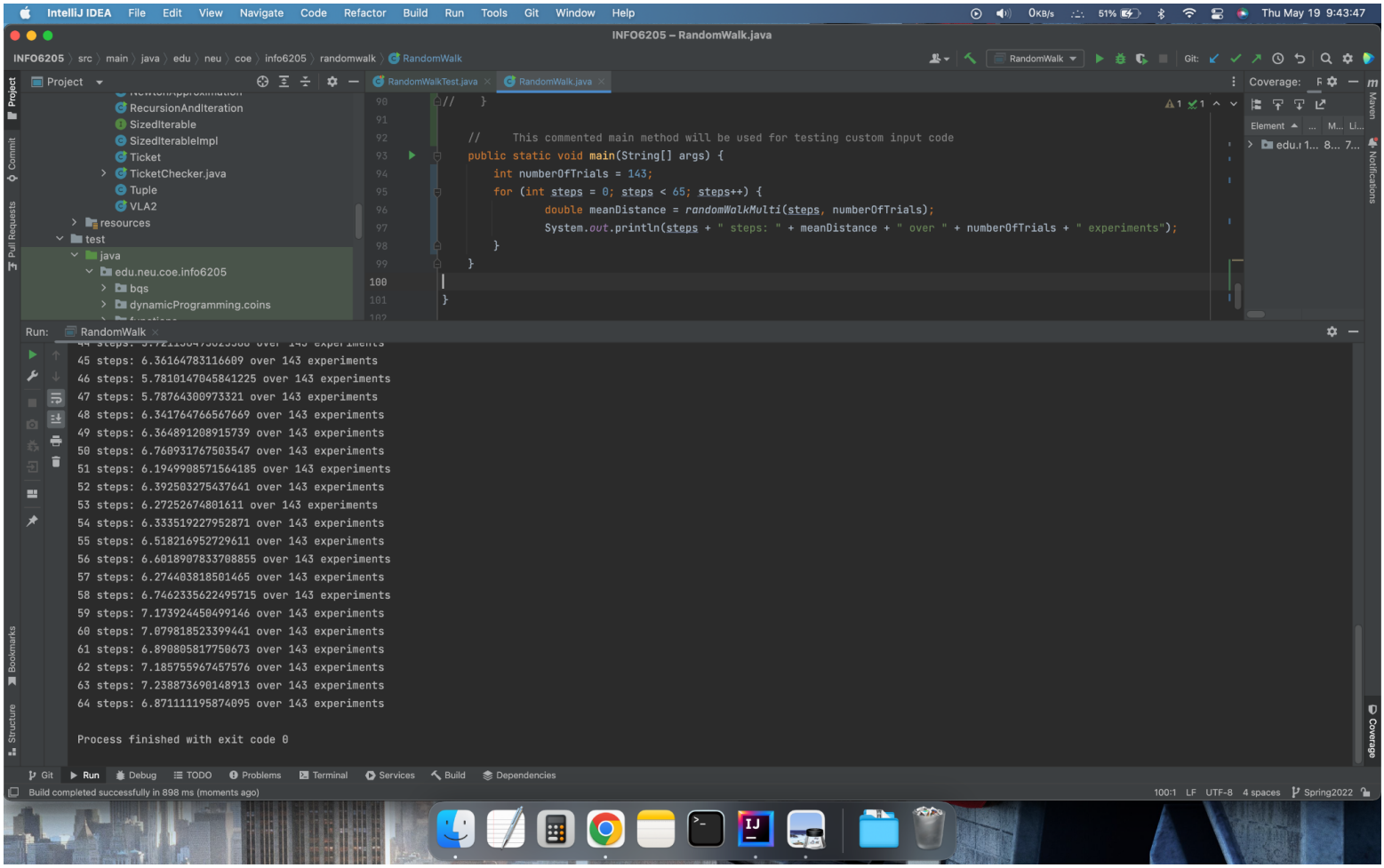


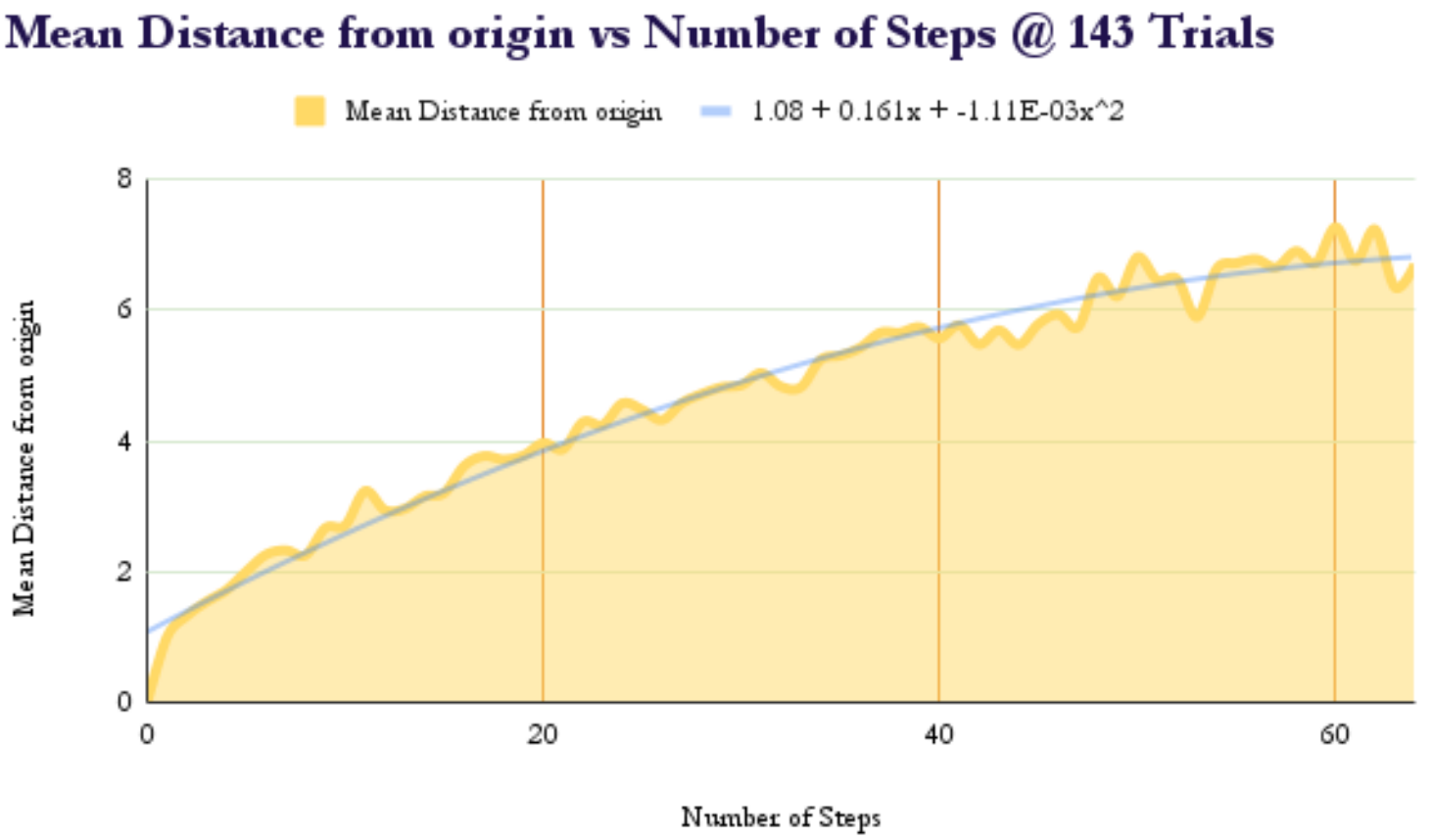
Console Output of Main method with custom tests running for steps ranging from 0 to 65 and with each steps count running for 143 Trials (Arbitrarily chosen)

Attached are Excel Files with the data and also the curve showing the trend of the mean distance


IntelliJ IDEA File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Thu May 19 9:43:33
INFO6205 – RandomWalk.java
INFO6205 src main java edu neu coe info6205 randomwalk RandomWalk
RandomWalkTest.java
RandomWalk.java
Project
RecursionAndIteration
SizedIterable
SizedIterableImpl
Ticket
TicketChecker.java
Tuple
VLA2
resources
test
java
edu.neu.coe.info6205
bqs
dynamicProgramming.coins
Coverage:
// }
// This commented main method will be used for testing custom input code
public static void main(String[] args) {
int numberOfTrials = 143;
for (int steps = 0; steps < 65; steps++) {
double meanDistance = randomWalkMulti(steps, numberOfTrials);
System.out.println(steps + " steps: " + meanDistance + " over " + numberOfTrials + " experiments");
}
}
}
Run: RandomWalk
/Library/Java/JavaVirtualMachines/jdk-17.0.3.jdk/Contents/Home/bin/java ...
0 steps: 0.0 over 143 experiments
1 steps: 1.0 over 143 experiments
2 steps: 1.2476165239754535 over 143 experiments
3 steps: 1.5786550404544675 over 143 experiments
4 steps: 1.755067249184577 over 143 experiments
5 steps: 1.8251200036991806 over 143 experiments
6 steps: 2.215276553860991 over 143 experiments
7 steps: 2.2256349322749323 over 143 experiments
8 steps: 2.4623713740585598 over 143 experiments
9 steps: 2.677007712017024 over 143 experiments
10 steps: 2.7907290831825655 over 143 experiments
11 steps: 2.887348467229678 over 143 experiments
12 steps: 2.962889769830948 over 143 experiments
13 steps: 3.4297964990903838 over 143 experiments
14 steps: 3.0867061011024517 over 143 experiments
15 steps: 3.238097229497339 over 143 experiments
16 steps: 3.481507980076352 over 143 experiments
17 steps: 3.576394574064715 over 143 experiments
18 steps: 3.5978359062601633 over 143 experiments
19 steps: 3.7692796846420995 over 143 experiments
20 steps: 3.6611738950899784 over 143 experiments
21 steps: 3.9429186276449766 over 143 experiments
Git Run Debug TODO Problems Terminal Services Build Dependencies
Build completed successfully in 898 ms (moments ago)
100:1 LF UTF-8 4 spaces Spring2022


IntelliJ IDEA File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Thu May 19 9:43:39
INFO6205 – RandomWalk.java
INFO6205 src main java edu neu coe info6205 randomwalk RandomWalk
RandomWalkTest.java
RandomWalk.java
Project
RecursionAndIteration
SizedIterable
SizedIterableImpl
Ticket
TicketChecker.java
Tuple
VLA2
resources
test
java
edu.neu.coe.info6205
bqs
dynamicProgramming.coins
Coverage:
// }
// This commented main method will be used for testing custom input code
public static void main(String[] args) {
int numberOfTrials = 143;
for (int steps = 0; steps < 65; steps++) {
double meanDistance = randomWalkMulti(steps, numberOfTrials);
System.out.println(steps + " steps: " + meanDistance + " over " + numberOfTrials + " experiments");
}
}
}
Run: RandomWalk
22 steps: 4.148535476286246 over 143 experiments
23 steps: 4.1019511581872345 over 143 experiments
24 steps: 4.374868175064304 over 143 experiments
25 steps: 4.503174039335038 over 143 experiments
26 steps: 4.891014546306989 over 143 experiments
27 steps: 4.619090549070666 over 143 experiments
28 steps: 4.93929111697967 over 143 experiments
29 steps: 4.562400511471506 over 143 experiments
30 steps: 4.870148575485079 over 143 experiments
31 steps: 4.45153802478022 over 143 experiments
32 steps: 5.210362984131392 over 143 experiments
33 steps: 5.225607592618178 over 143 experiments
34 steps: 5.37011782953883 over 143 experiments
35 steps: 5.600201733300867 over 143 experiments
36 steps: 5.416618762806575 over 143 experiments
37 steps: 5.642309923540434 over 143 experiments
38 steps: 5.221490693787628 over 143 experiments
39 steps: 4.923864590687123 over 143 experiments
40 steps: 5.769411115634657 over 143 experiments
41 steps: 5.807216502016028 over 143 experiments
42 steps: 5.582205241682079 over 143 experiments
43 steps: 5.5671554088877055 over 143 experiments
44 steps: 5.721136495025588 over 143 experiments
Git Run Debug TODO Problems Terminal Services Build Dependencies
Build completed successfully in 898 ms (moments ago)
100:1 LF UTF-8 4 spaces Spring2022


IntelliJ IDEA File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Thu May 19 9:43:47
INFO6205 – RandomWalk.java
RecursionAndIteration
SizedIterable
SizedIterableImpl
Ticket
TicketChecker.java
Tuple
VLA2
resources
test
java
edu.neu.coe.info6205
bqs
dynamicProgramming.coins
RandomWalkTest.java
RandomWalk.java
// }
// This commented main method will be used for testing custom input code
public static void main(String[] args) {
int numberOfTrials = 143;
for (int steps = 0; steps < 65; steps++) {
double meanDistance = randomWalkMulti(steps, numberOfTrials);
System.out.println(steps + " steps: " + meanDistance + " over " + numberOfTrials + " experiments");
}
}
}
Run: RandomWalk
45 steps: 6.36164783116609 over 143 experiments
46 steps: 5.7810147045841225 over 143 experiments
47 steps: 5.78764300973321 over 143 experiments
48 steps: 6.341764766567669 over 143 experiments
49 steps: 6.364891208915739 over 143 experiments
50 steps: 6.760931767503547 over 143 experiments
51 steps: 6.1949908571564185 over 143 experiments
52 steps: 6.392503275437641 over 143 experiments
53 steps: 6.27252674801611 over 143 experiments
54 steps: 6.333519227952871 over 143 experiments
55 steps: 6.518216952729611 over 143 experiments
56 steps: 6.6018907833708855 over 143 experiments
57 steps: 6.274403818501465 over 143 experiments
58 steps: 6.7462335622495715 over 143 experiments
59 steps: 7.173924450499146 over 143 experiments
60 steps: 7.079818523399441 over 143 experiments
61 steps: 6.890805817750673 over 143 experiments
62 steps: 7.185755967457576 over 143 experiments
63 steps: 7.238873690148913 over 143 experiments
64 steps: 6.871111195874095 over 143 experiments
Process finished with exit code 0
Git Run Debug TODO Problems Terminal Services Build Dependencies
Build completed successfully in 898 ms (moments ago)
100:1 LF UTF-8 4 spaces Spring2022


Mean Distance from origin vs Number of Steps @ 143 Trials
Mean Distance from origin
1.08 + 0.161x + -1.11E-03x^2
Mean Distance from origin
Number of Steps
0
2
4
6
8
0
20
40
60

Excel File

Assignment 01 Data

Conclusion

The mean distance showed a growth in small increments for the experiments run from 0 to 65 steps and each step running for 143 trials. From the graph above the trend also appears follows a curve

$$y = 1.08 + (0.161)(x) + (-1.11E-03)(x^2)$$

which is a second degree quadratic equation