Program Structures and Algorithms
Spring 2024

NAME: Rahul Pothirendi
NUID: 002889957
GITHUB LINK: https://github.com/Pothirendirahul/INFO6205.git

Task: Analyzing the Relationship between Number of Steps (m) and Mean Distance

Relationship Conclusion: The relationship between the number of steps (m) and the mean distance appears to follow a pattern where the mean distance increases as the number of steps increases.
the relationship between the number of steps (m) and the mean distance from the starting point can be modeled using the Euclidean distance. The Euclidean distance is a measure of straight-line distance between two points in space. The proposed model suggests that the mean distance (Mean Distance Mean Distance) covered by a random walk tends to increase with the square root of the number of steps. Mathematically, this relationship is expressed as:
Mean Distance=k(square root of m) m

Evidence to support that conclusion: After running the experiments for at least six values of m and ten times each (n = 10), the following observations were made:

Observation for m = 10;
- **Mean Distance = 10 steps: 2.636233567652581 over 10 experiments**

Observation for m = 20;
- **Mean Distance = 20 steps: 4.08745070881268 over 10 experiments**

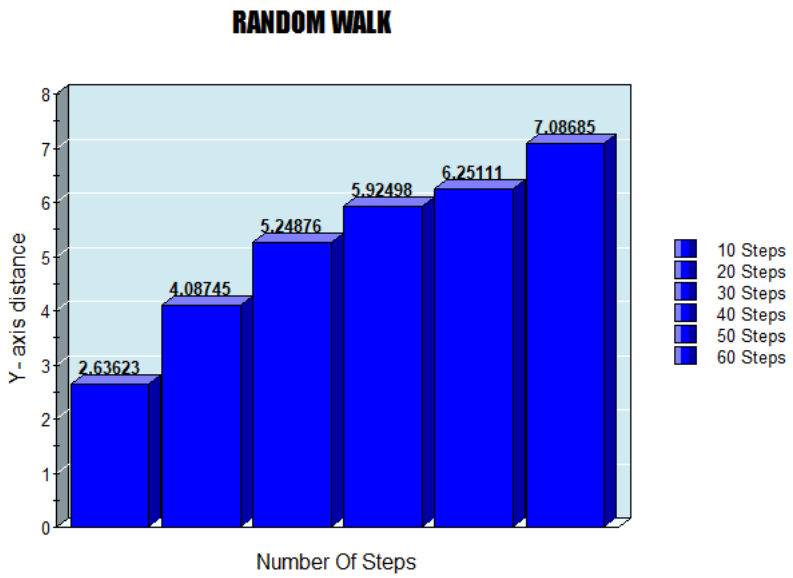
Observation for m = 30;
- **Mean Distance = 30 steps: 5.248769708295801 over 10 experiments**

Observation for m = 40;
- **Mean Distance = 40 steps: 5.92498413294314 over 10 experiments**

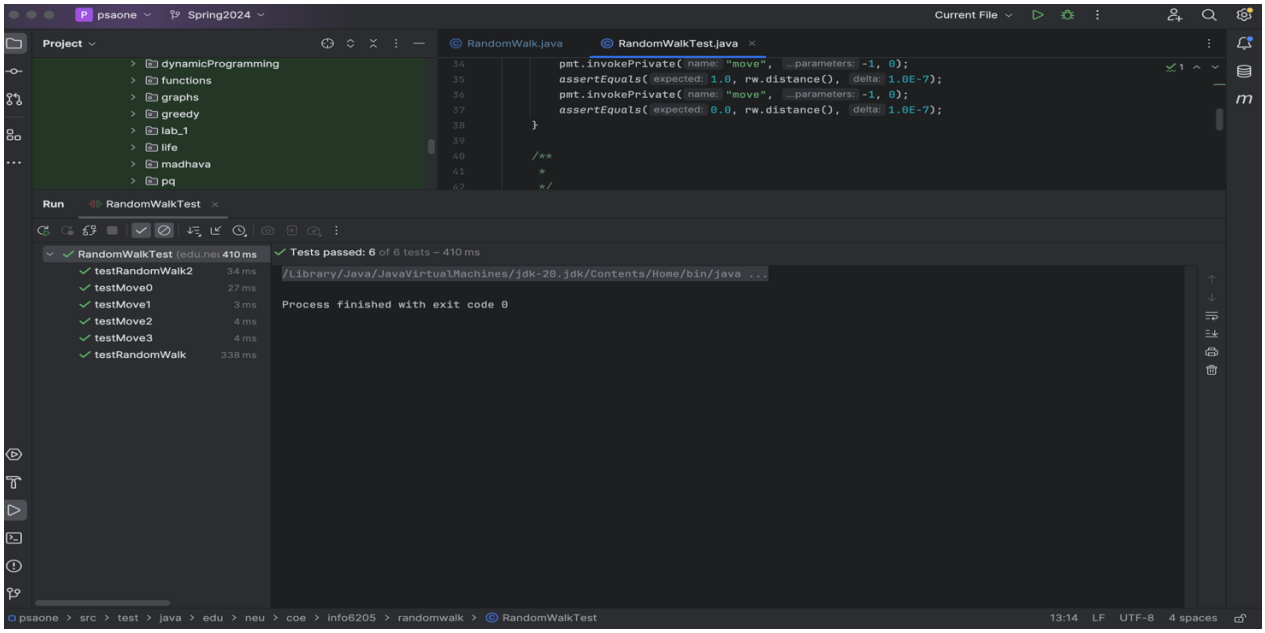
Observation for m = 50;
- **Mean Distance = 50 steps: 6.251111813206244 over 10 experiments**

Observation for m = 60;
- **Mean Distance = 60 steps: 7.086855629626768 over 10 experiments**

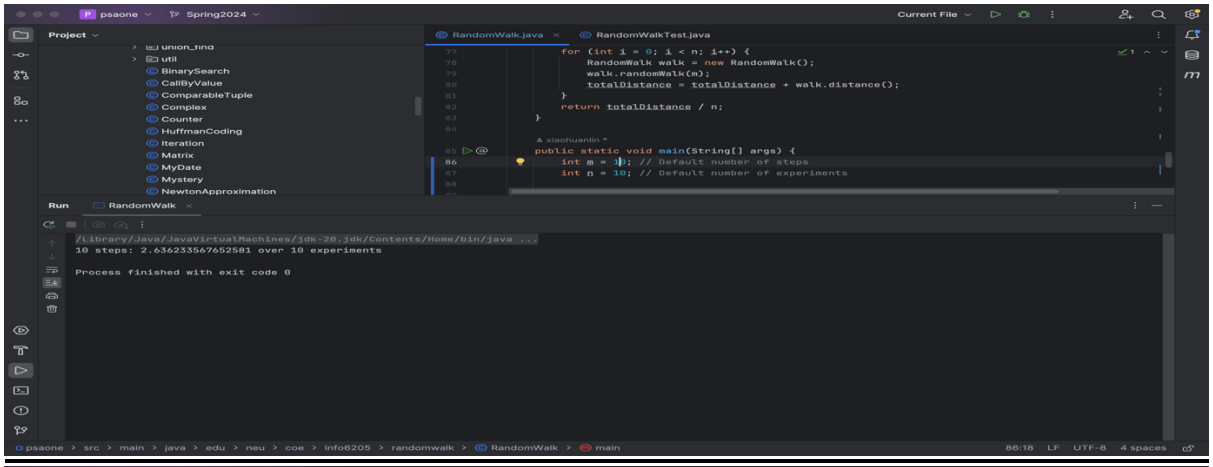
Graphical Output for Reference

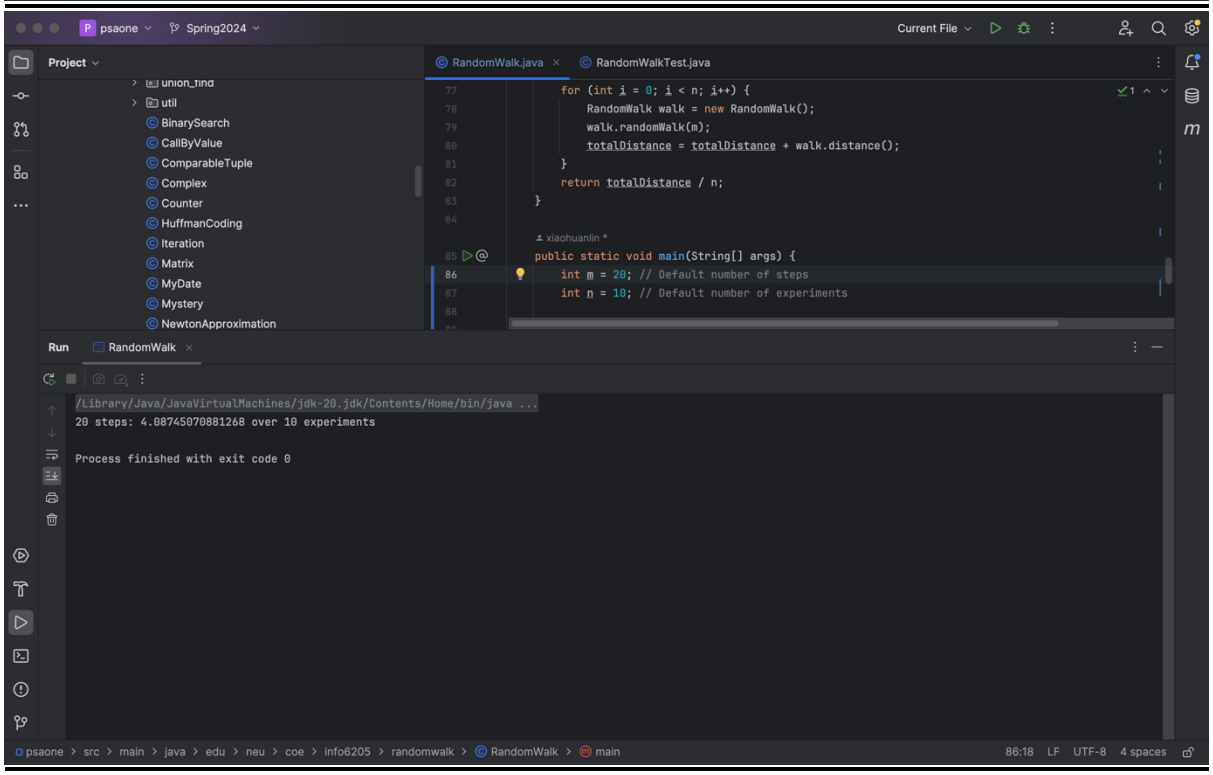


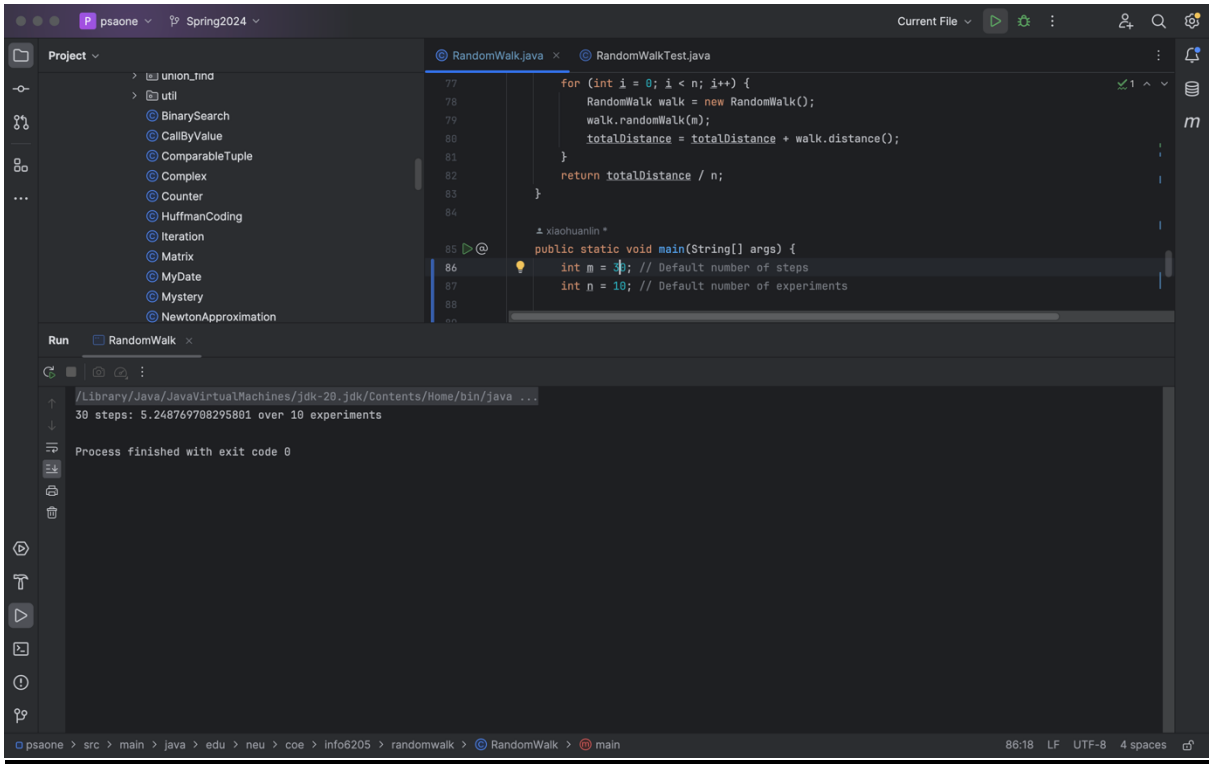
Unit Test Screenshots:

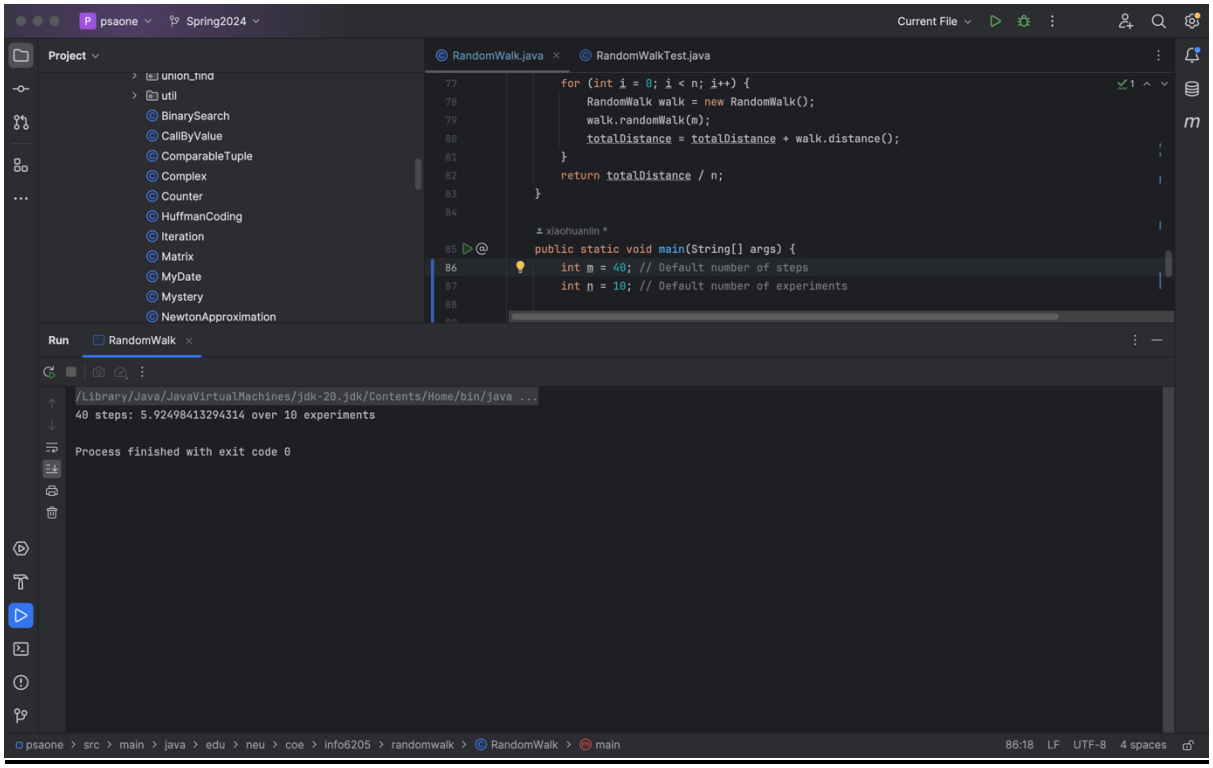


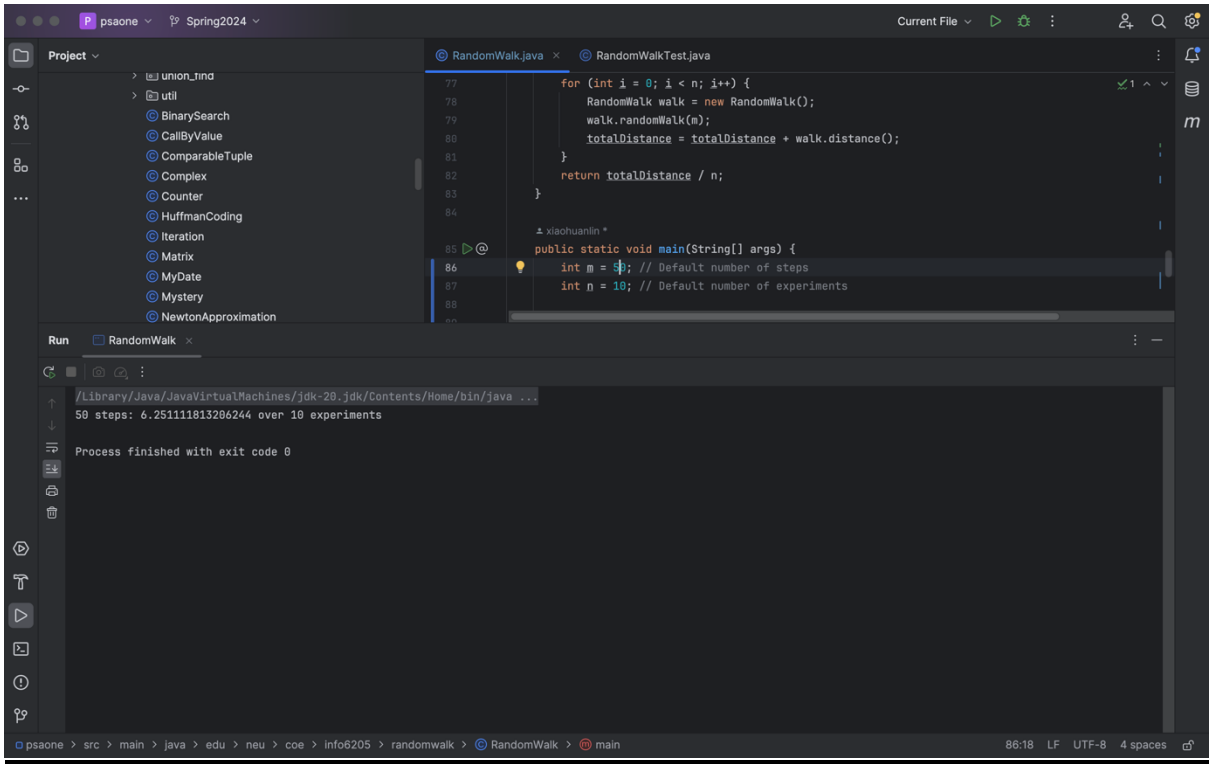
ScreenShots:-

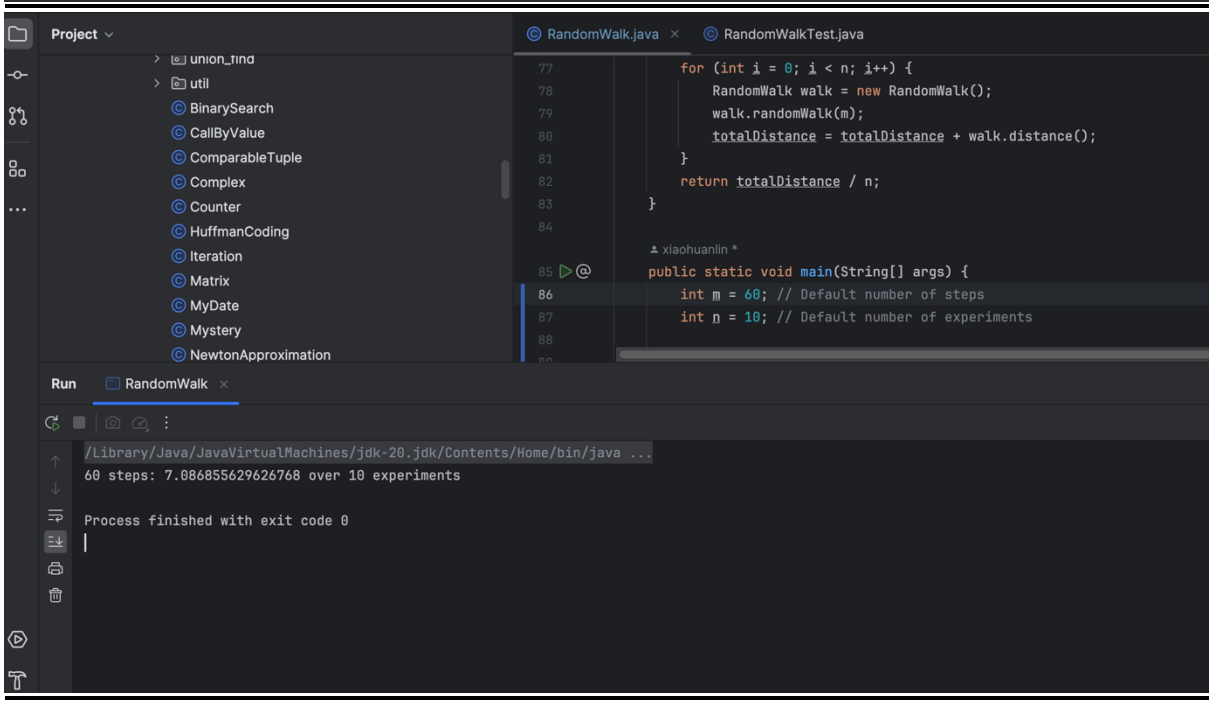





psaone
Spring2024
Current File
Project
union_find
util
BinarySearch
CallByValue
ComparableTuple
Complex
Counter
HuffmanCoding
Iteration
Matrix
MyDate
Mystery
NewtonApproximation
RandomWalk.java
RandomWalkTest.java
for (int i = 0; i < n; i++) {
RandomWalk walk = new RandomWalk();
walk.randomWalk(m);
totalDistance = totalDistance + walk.distance();
}
return totalDistance / n;
}
xiaohuanlin *
public static void main(String[] args) {
int m = 30; // Default number of steps
int n = 10; // Default number of experiments
Run
RandomWalk
/Library/Java/JavaVirtualMachines/jdk-20.jdk/Contents/Home/bin/java ...
30 steps: 5.248769708295801 over 10 experiments
Process finished with exit code 0
psaone > src > main > java > edu > neu > coe > info6205 > randomwalk > RandomWalk > main
86:18 LF UTF-8 4 spaces


psaone
Spring2024
Current File
Project
union_find
util
BinarySearch
CallByValue
ComparableTuple
Complex
Counter
HuffmanCoding
Iteration
Matrix
MyDate
Mystery
NewtonApproximation
RandomWalk.java
RandomWalkTest.java
for (int i = 0; i < n; i++) {
RandomWalk walk = new RandomWalk();
walk.randomWalk(m);
totalDistance = totalDistance + walk.distance();
}
return totalDistance / n;
}
xiaohuanlin *
public static void main(String[] args) {
int m = 40; // Default number of steps
int n = 10; // Default number of experiments
Run
RandomWalk
/Library/Java/JavaVirtualMachines/jdk-20.jdk/Contents/Home/bin/java ...
40 steps: 5.92498413294314 over 10 experiments
Process finished with exit code 0
psaone > src > main > java > edu > neu > coe > info6205 > randomwalk > RandomWalk > main
86:18 LF UTF-8 4 spaces


psaone
Spring2024
Current File
Project
RandomWalk.java
RandomWalkTest.java
union_find
util
BinarySearch
CallByValue
ComparableTuple
Complex
Counter
HuffmanCoding
Iteration
Matrix
MyDate
Mystery
NewtonApproximation
for (int i = 0; i < n; i++) {
RandomWalk walk = new RandomWalk();
walk.randomWalk(m);
totalDistance = totalDistance + walk.distance();
}
return totalDistance / n;
}
xiaohuanlin *
public static void main(String[] args) {
int m = 50; // Default number of steps
int n = 10; // Default number of experiments
Run
RandomWalk
/Library/Java/JavaVirtualMachines/jdk-20.jdk/Contents/Home/bin/java ...
50 steps: 6.251111813206244 over 10 experiments
Process finished with exit code 0
psaone > src > main > java > edu > neu > coe > info6205 > randomwalk > RandomWalk > main
86:18 LF UTF-8 4 spaces


Project
RandomWalk.java
RandomWalkTest.java
union_find
util
BinarySearch
CallByValue
ComparableTuple
Complex
Counter
HuffmanCoding
Iteration
Matrix
MyDate
Mystery
NewtonApproximation
for (int i = 0; i < n; i++) {
RandomWalk walk = new RandomWalk();
walk.randomWalk(m);
totalDistance = totalDistance + walk.distance();
}
return totalDistance / n;
}
xiaohuanlin *
public static void main(String[] args) {
int m = 60; // Default number of steps
int n = 10; // Default number of experiments
Run
RandomWalk
/Library/Java/JavaVirtualMachines/jdk-20.jdk/Contents/Home/bin/java ...
60 steps: 7.086855629626768 over 10 experiments
Process finished with exit code 0