Program Structures and Algorithms

Spring 2024

Name: Yu Lai
NUID: 002640845
GitHub: https://github.com/laiyumi/INFO6205

1 Task: RandomWalk

2 Conclusion

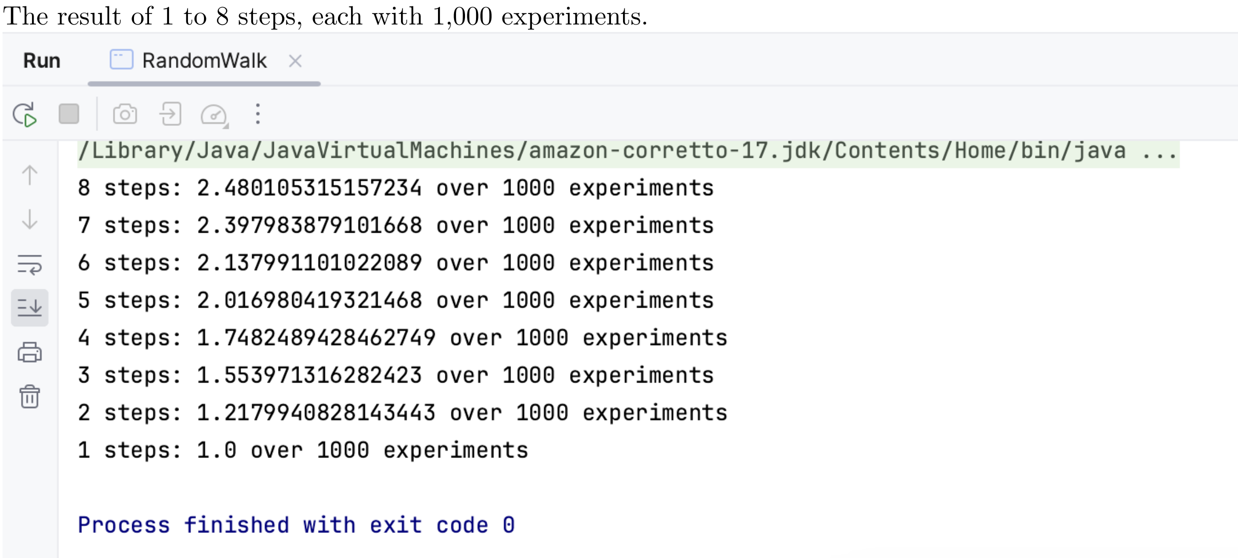
There exists a constant x, so that $d^x \propto m$. After calculating and testing, $x \approx 2.1$, thus, we could say

$$d^{2.1} \propto m$$

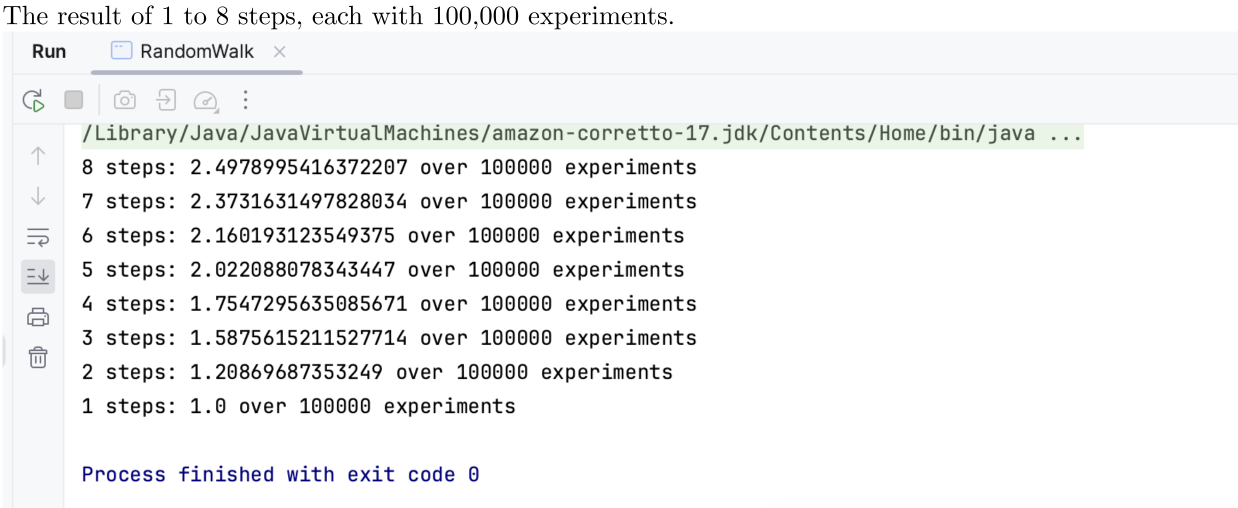
Apart from that, $0 <= d <= m$.

3 Evidence

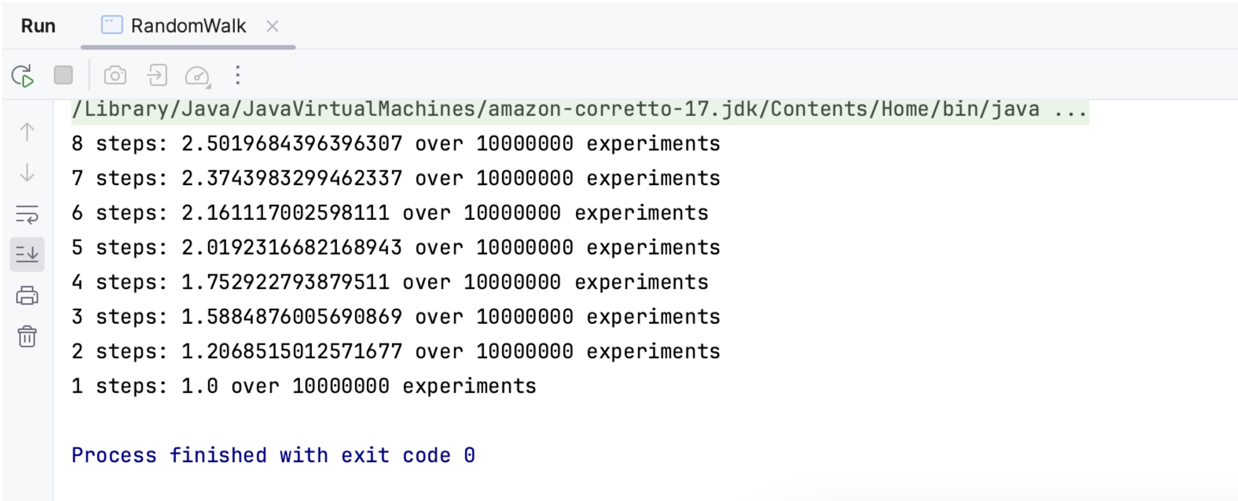
By running more experiments on different steps, it appears that as the quantity of experiments increases significantly, the average of m steps gravitates towards a specific value. As evident in the following screenshots, using the example of 8 steps, when the number of experiments increases from 1,000 to 10,000,000, the average distance increasingly approximates 2.50.
The result of 1 to 8 steps, each with 1,000 experiments.



The result of 1 to 8 steps, each with 100,000 experiments.



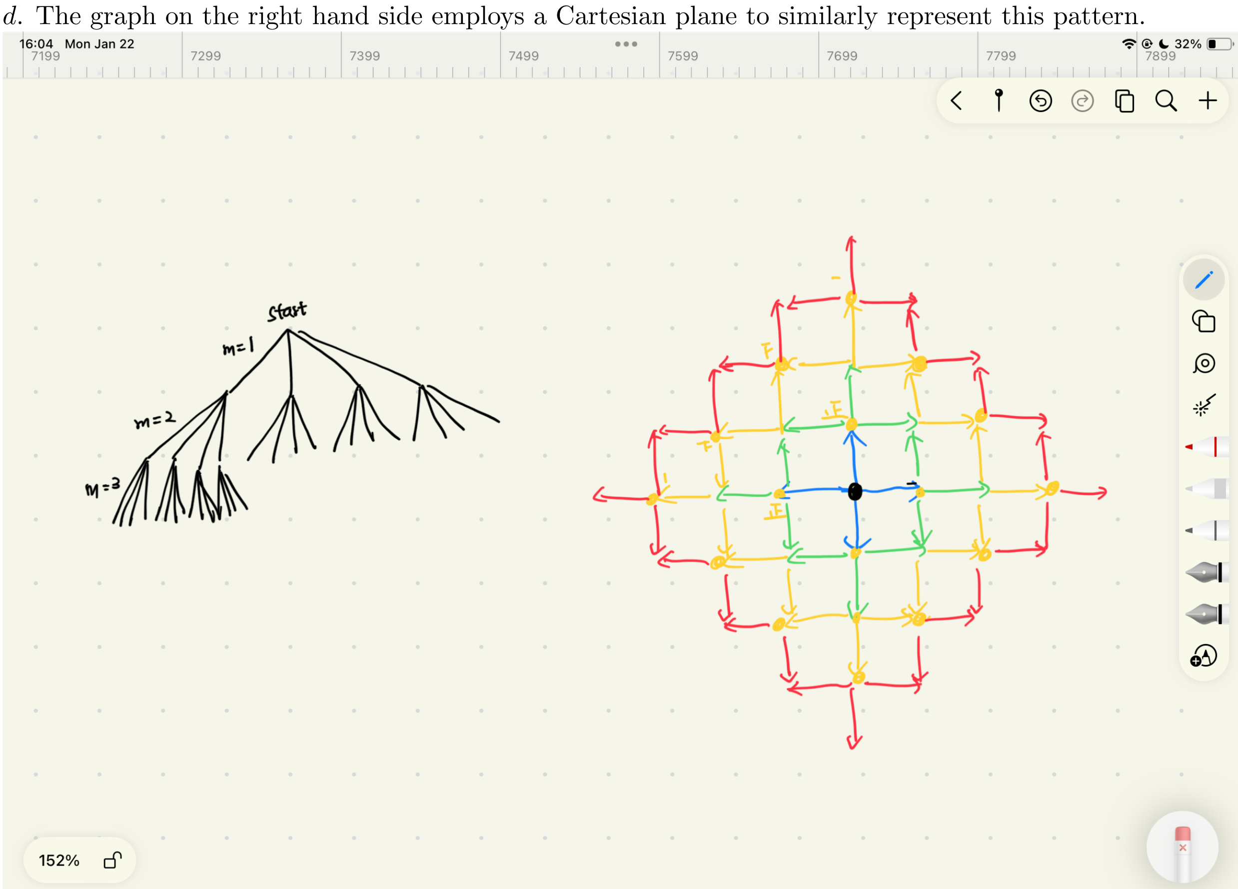
The result of 1 to 8 steps, each with 10,000,000 experiments.



```
Run    RandomWalk
/Library/Java/JavaVirtualMachines/amazon-corretto-17.jdk/Contents/Home/bin/java ...
8 steps: 2.5019684396396307 over 10000000 experiments
7 steps: 2.3743983299462337 over 10000000 experiments
6 steps: 2.161117002598111 over 10000000 experiments
5 steps: 2.0192316682168943 over 10000000 experiments
4 steps: 1.752922793879511 over 10000000 experiments
3 steps: 1.5884876005690869 over 10000000 experiments
2 steps: 1.2068515012571677 over 10000000 experiments
1 steps: 1.0 over 10000000 experiments

Process finished with exit code 0
```

I've utilized two graphs to identify the pattern. On the left, a tree diagram illustrates that every time m increases by one, the number of potential d scenarios experiences an exponential growth of four. For example, when m equals 1, there are four instances of d. When m equals 2, there are 4^2 which equals 16 instances of d. The graph on the right hand side employs a Cartesian plane to similarly represent this pattern.



Therefore, an exponential relationship appears to exist between d and m, which I've proposed as $C * d^x = m$, with C and x being constants. Simplifying, this expression can be written as $d^x \propto m$. Then we have:

$$log(d^x) = log(m)$$

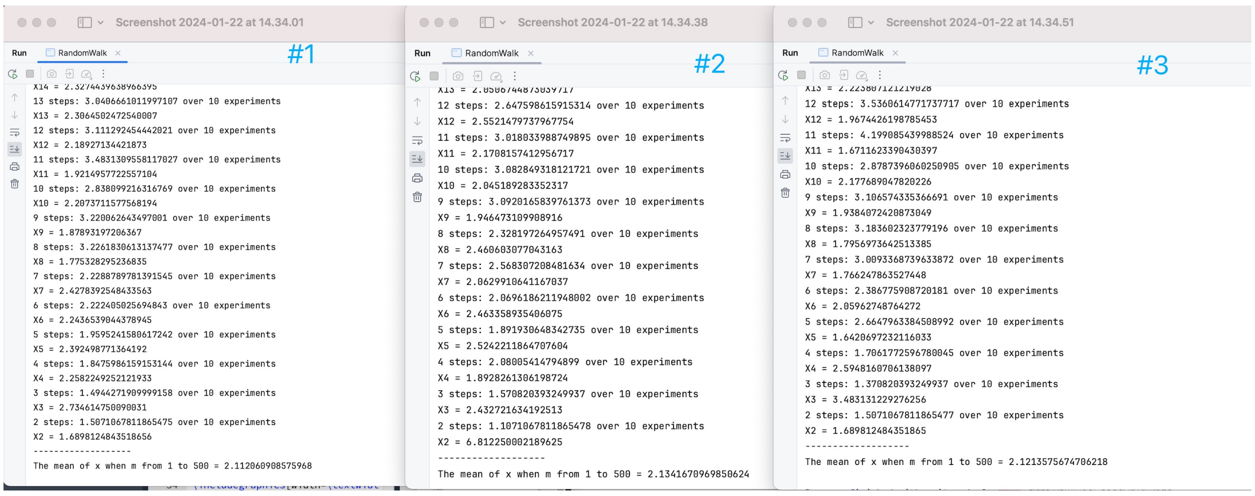
$$x * log(d) = log(m)$$

$$x = \frac{log(m)}{log(d)}$$

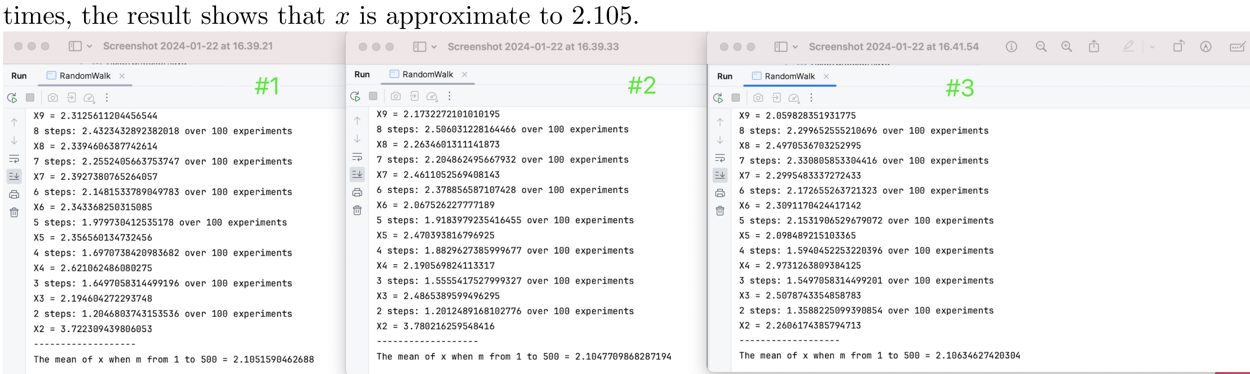
$$x = log(\frac{m}{d})$$

$$x = log_d(m)$$

Then, we run the code using m from $1 - 500$, each step with 10 experiments, to calculate the average of x. After conducting this test three times, it becomes apparent that the average of x is approximately 2.12.



Then we run another test, when m from $1-500$ and each m with 100 experiments. After conducting three times, the result shows that x is approximate to 2.105.



Thus, we deduces that $x \approx 2.1$, then $d^{2.1} \propto m$.

4 Code

Please see GitHub repository.

5 Unit Test Screenshots