

Program Structures and Algorithms Fall 2022(SEC 06)

NAME: Olasunkanmi Olayinka
NUID: 001512266

Task:

- Implement the code for the experiment
- Deduce the relationship between the distance(d) and number of steps(n) taken

Relationship Conclusion:

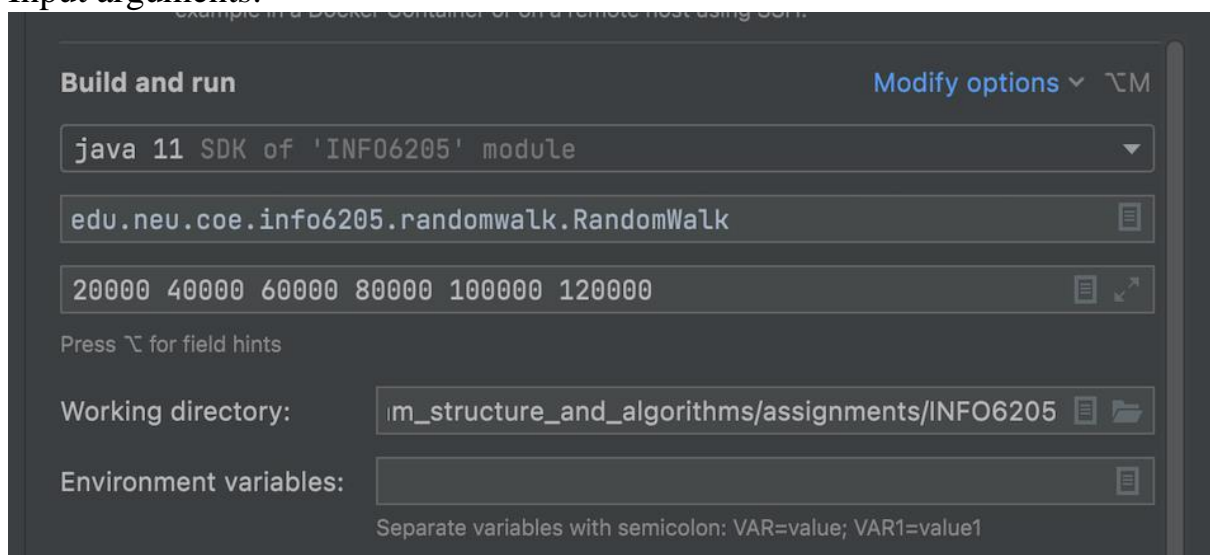
Distance(d) $\approx k * \sqrt{\text{no of steps}(n)}$

I can conclude that as the number of experiments increases $d \approx k * \sqrt{n}$, where k is a constant, distance(d) does go above or below it sometimes, but it usually revolves around this value. Also, distance(d) tends to increase with respect to the number of steps (n).

Evidence to support that conclusion:

The tables show the distance(d) from the lamp post, number of steps(n) taken, sqrt(n) and number of experiments to see the relationship between d and n.

Input arguments:



Test over 25 experiments:

```

Run: RandomWalk
/Library/Java/JavaVirtualMachines/jdk-11.0.12.jdk/Contents/Home/bin/java ...
20000 steps: 117.3354127004303 over 25 experiments--> sqrt of 20000steps: 141.4213562373095
40000 steps: 153.86510883589602 over 25 experiments--> sqrt of 40000steps: 200.0
60000 steps: 205.00973620822018 over 25 experiments--> sqrt of 60000steps: 244.94897427831782
80000 steps: 257.63063164699315 over 25 experiments--> sqrt of 80000steps: 282.842712474619
100000 steps: 276.70724966741193 over 25 experiments--> sqrt of 100000steps: 316.22776601683796
120000 steps: 311.6795746318287 over 25 experiments--> sqrt of 120000steps: 346.41016151377545

Process finished with exit code 0

```

no of steps(n)	distance(d)	sqrt of (n)	no of experiments
20000	117.3354127	141.4213562	25
40000	153.8651008	200	25
60000	205.0097362	244.9489743	25
80000	257.6306316	282.8427125	25
100000	276.7072497	316.227766	25
120000	311.6795746	346.4101615	25

Test over 50 experiments:

```

Run: RandomWalk
/Library/Java/JavaVirtualMachines/jdk-11.0.12.jdk/Contents/Home/bin/java ...
20000 steps: 134.19656689816267 over 50 experiments--> sqrt of 20000steps: 141.4213562373095
40000 steps: 178.68978263583864 over 50 experiments--> sqrt of 40000steps: 200.0
60000 steps: 211.72720455438946 over 50 experiments--> sqrt of 60000steps: 244.94897427831782
80000 steps: 248.01673424348323 over 50 experiments--> sqrt of 80000steps: 282.842712474619
100000 steps: 288.6709126245887 over 50 experiments--> sqrt of 100000steps: 316.22776601683796
120000 steps: 314.3531468499198 over 50 experiments--> sqrt of 120000steps: 346.41016151377545

Process finished with exit code 0

```

no of steps(n)	distance(d)	sqrt of (n)	no of experiments
20000	134.1965669	141.4213562	50
40000	178.6897826	200	50
60000	211.7272046	244.9489743	50
80000	248.0167342	282.8427125	50
100000	288.6709126	316.227766	50
120000	314.3531468	346.4101615	50

Test over 100 experiments:

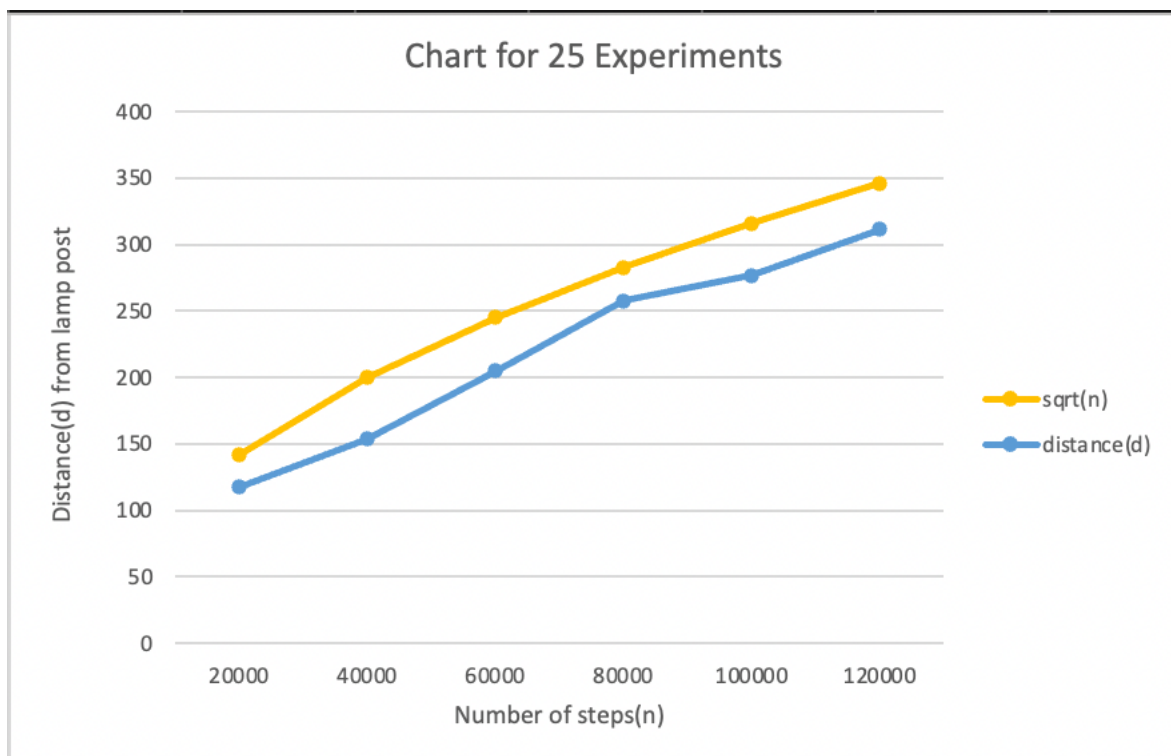
```
Run: RandomWalk
/Library/Java/JavaVirtualMachines/jdk-11.0.12.jdk/Contents/Home/bin/java ...
20000 steps: 124.5207095 over 100 experiments--> sqrt of 20000 steps: 141.4213562373095
40000 steps: 186.0858196 over 100 experiments--> sqrt of 40000 steps: 200.0
60000 steps: 210.6506523 over 100 experiments--> sqrt of 60000 steps: 244.94897427831782
80000 steps: 264.0135175 over 100 experiments--> sqrt of 80000 steps: 282.842712474619
100000 steps: 284.0113413 over 100 experiments--> sqrt of 100000 steps: 316.22776601683796
120000 steps: 330.8880472 over 100 experiments--> sqrt of 120000 steps: 346.41016151377545

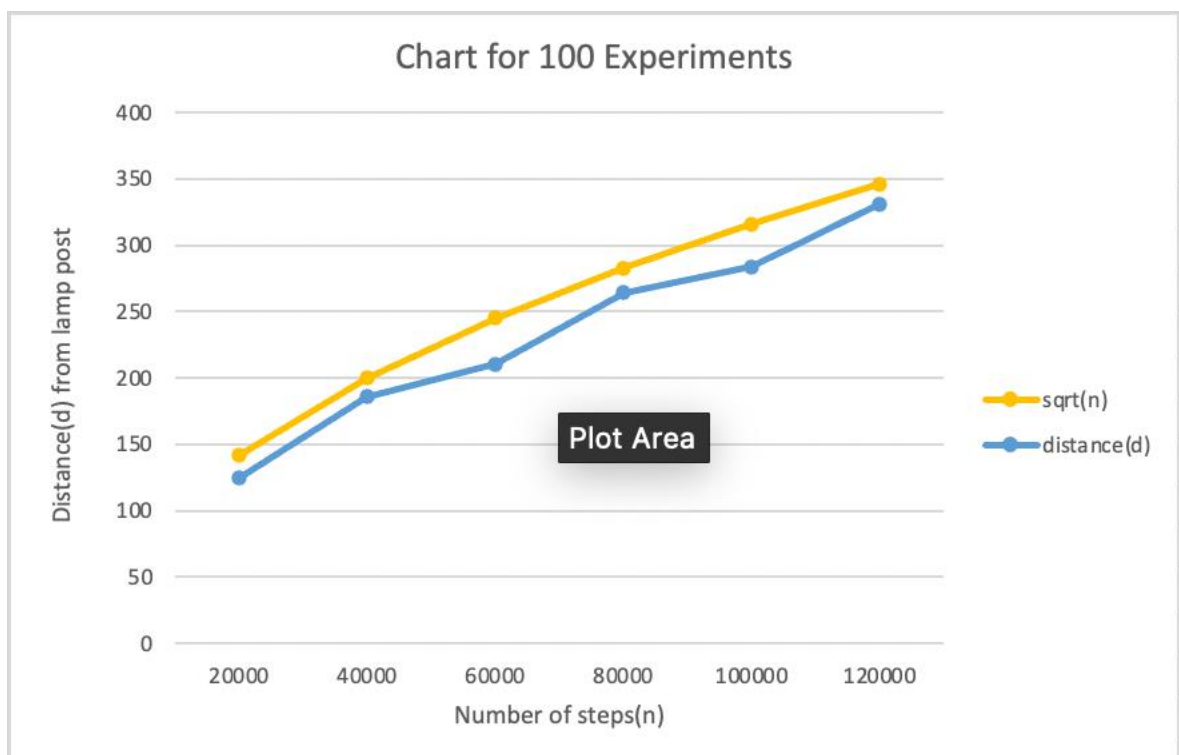
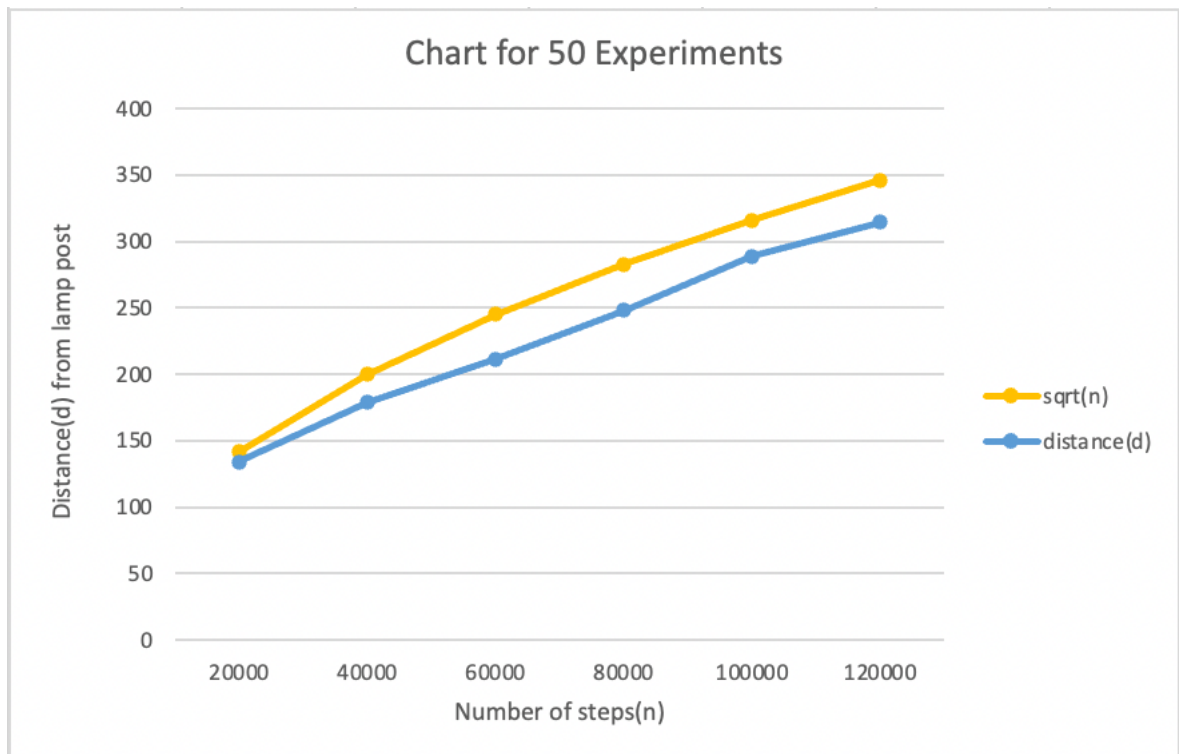
Process finished with exit code 0
```

no of steps(n)	distance(d)	sqrt of (n)	no of experiments
20000	124.5207095	141.4213562	100
40000	186.0858196	200	100
60000	210.6506523	244.9489743	100
80000	264.0135175	282.8427125	100
100000	284.0113413	316.227766	100
120000	330.8880472	346.4101615	100

Graphical Representation:

The line graphs below show us the relationship between the distance(d) moved and the square root of n. This gives us a better idea that $d \approx k \cdot \sqrt{n}$ and this is clearer with higher experiment number.





Unit Test Screenshots:

