Program Structures and Algorithms
Spring 2023(SEC 3)

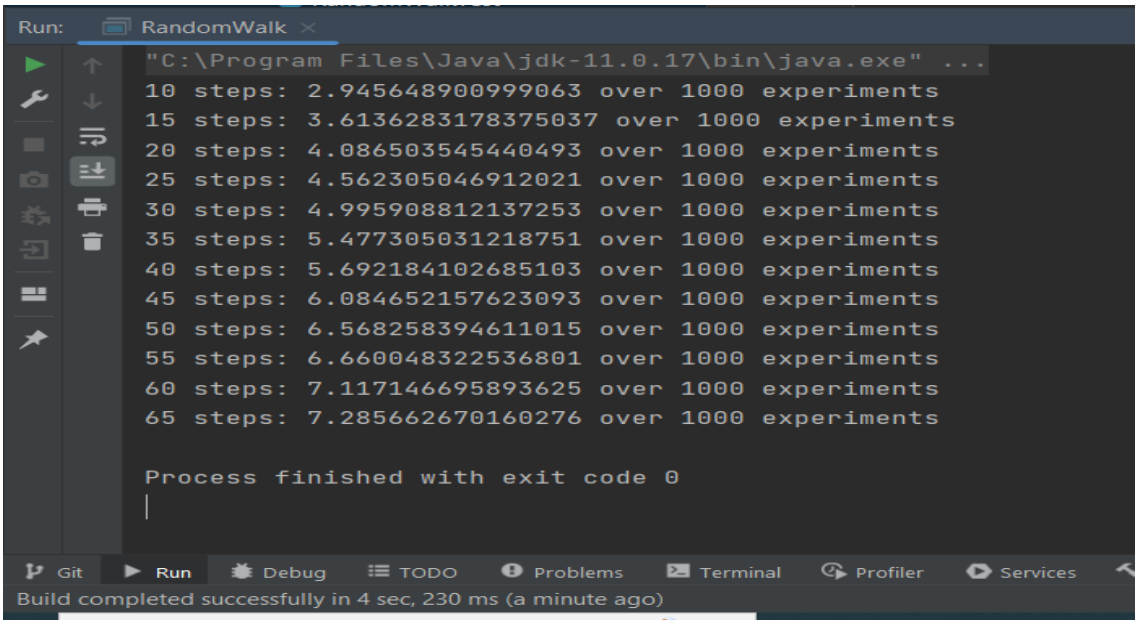
NAME: Subbu Manickam
NUID: 002768764

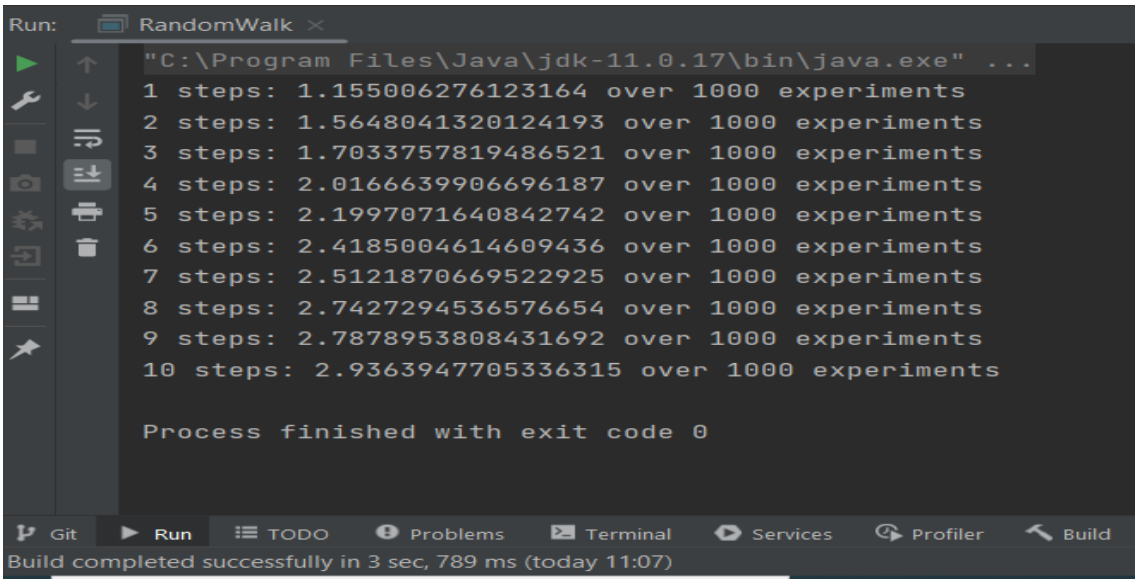
Task: Assignment 1 - Random Walk – To find the relationship between d and m

Relationship Conclusion: The total mean distance (d) is directly proportional to the square root of number of steps (m).

d = k (m)^0.5, where k is any constant

Evidence to support that conclusion: As the number of steps (m) increases the mean distance (d) increases as a product of the square root of the value.



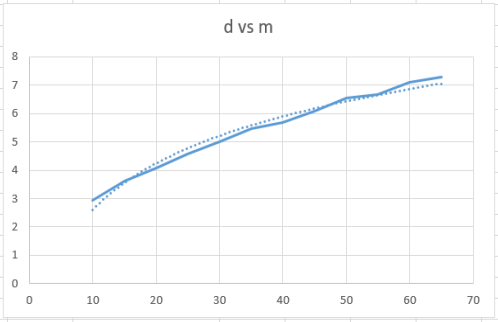


Graphical Representation: The below image shows the comparison of the two parameters d and m in a graphical format.

- The correlation coefficient (~1), confirms that m and d have a direct proportionality
- The graph resembles a logarithmic plot
- By converting the graph to (log m) vs (log d), we get a linear curve
- The calculated slope of the plot is 0.485 which is approximately 0.5
- Logarithmic slope relation equation is x = k (y)^m, where k is a constant and m is the slope of the graph
- Thus d = k (m)^0.5, where k is any constant

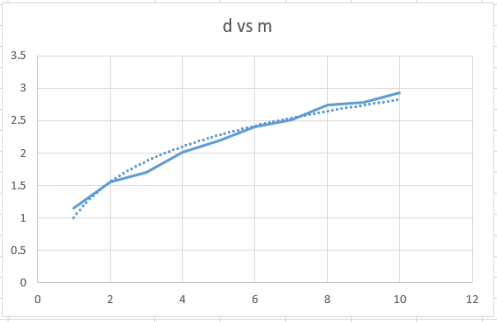
m	d
10	2.94
15	3.61
20	4.08
25	4.56
30	4.99
35	5.47
40	5.69
45	6.08
50	6.56
55	6.66
60	7.11
65	7.28

Correlation Coefficient	Logarithmic slope
0.991904408	0.506328



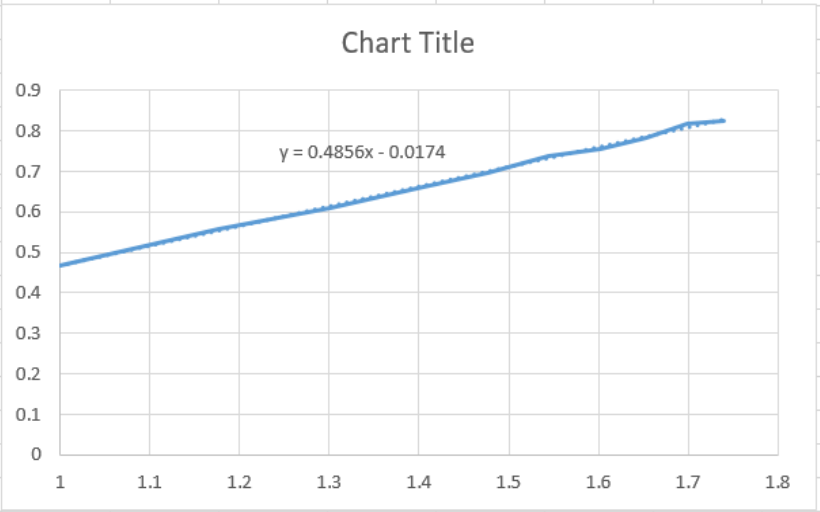
m	d
1	1.15
2	1.56
3	1.7
4	2.01
5	2.19
6	2.41
7	2.51
8	2.74
9	2.78
10	2.93

Correlation Coefficient	Logarithmic slope
0.984332313	0.439912



Conclusion d ~ m^0.5

log m	log d
1	0.468347
1.176091	0.557507
1.30103	0.61066
1.39794	0.658965
1.477121	0.698101
1.544068	0.737987
1.60206	0.755112
1.653213	0.783904
1.69897	0.816904
1.740363	0.823474



Unit Tests Screenshot:

Code Snippet:

```
private void move(int dx, int dy) {
    // FIXME do move by replacing the following code
     x += dx;
     y += dy;
    // END
}
```

```
private void randomWalk(int m) {
    // FIXME
    while(m>=0) {
        randomMove();
        m--;
    }
    // END
}
```

```
public double distance() {
    // FIXME by replacing the following code
    double distance = Math.sqrt(((x * x) + (y * y)));
    return distance;
    // END
}
```