

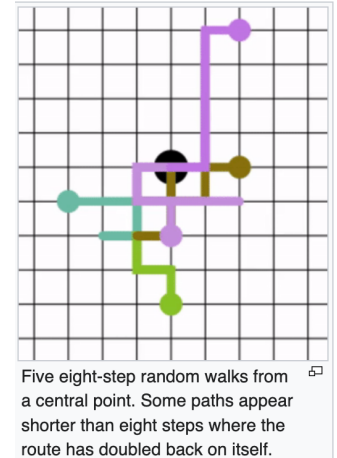
Random Walk

Objective: To solve a programming problem by decomposition and produce a histogram.

Background:

Imagine you're taking a stroll on a grid, where each step is like a little hop in one of the four cardinal directions: north, south, east, or west. We'll measure how far you've traveled by looking at the straight line from where you started to where you ended up.

Imagine we're taking an 8-step journey! We'll begin at a specific spot on the grid and move forward 8 times, each time in a direction chosen at random. Depending on how we move, the path might look like a straight line, a circle, or even loop back to right where we started! You can see five examples of these paths in the diagram on the right. The big dot is our starting point, and the smaller dots are where we end up. To figure out how long each path is, we use the distance formula $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.



To get a good idea, we'll run this random walk a million times and then average out the distances for each path. This is called a Monte Carlo simulation. Computers are great at making these calculations quick, precise, and reliable.

Because it's so fast, we'll do this for paths of all lengths, from just one step to twenty. We'll calculate the average distance for each one.

These average distances are just numbers, so it would look much better as a histogram, like the one below:

[illegible]

The ideas in this assignment are straightforward, but putting them into action can be a bit tricky. You'll need to design a data structure to store the information and develop an algorithm to calculate and show the histogram. If you try to do everything in one method, it'll become a tangled mess and hard to fix. The best approach is to break down the algorithm into smaller steps or what we call decomposition.

Assignment:

The project has three parts.

First, you must come up with a data structure to hold the path and distance information.

Second, you must come up with algorithms to perform the random walk, the Monte Carlo simulation, and to print a histogram.

Third, you must decompose your code into manageable methods.

A sample run:

```
% java RandomWalk
```

Welcome to RandomWalk

Calculating walks of length 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Walks of

Length	AveDist	
1	1.0000	*****
2	1.2078	*****
3	1.5892	*****
4	1.7523	*****
5	2.0194	*****
6	2.1615	*****
7	2.3750	*****
8	2.5050	*****
9	2.6834	*****
10	2.7986	*****
11	2.9607	*****
12	3.0688	*****
13	3.2136	*****
14	3.3172	*****
15	3.4492	*****
16	3.5453	*****
17	3.6691	*****
18	3.7635	*****
19	3.8795	*****
20	3.9652	*****

All Done!