Table of Contents

Description

Look at Figure 1, taken from the *Feynman Lectures on Physics* (see acknowledgements).

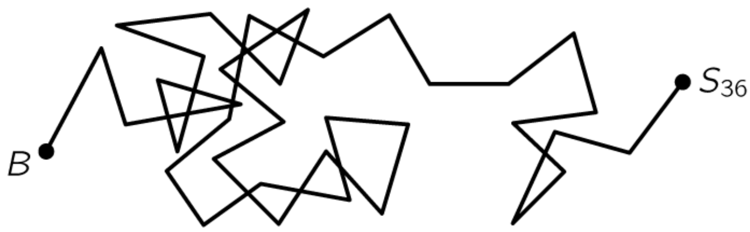

Figure 1: A 2D random walk of 36 steps of length l.

The figure represents the path that a randomly moving particle might take inside a 2D space. This particular ‘model’, as you can see from reading Feynman’s 41st lecture , of volume 1 in his classic 3-volume series, assumes a continuous range of angles of collision (“bombarded from all side”), that kick the particle around. As Feynman says[1] in that lecture:

> *“Let us consider how the position of a jiggling particle should change with time, for very long times compared with the time between “kicks.” Consider a little Brownian movement particle which is jiggling about because it is bombarded on all sides by irregularly jiggling water molecules. Query: After a given length of time, how far away is it likely to be from where it began?”*

---

[1] Yes, he ‘**says**’. He didn’t write the lectures, he just made notes before class then these were transcribed. I’d encourage you to listen to some of the lectures or parts of them, whichever take your fancy; the lectures and the recordings, where science is concerned. are a great treasure of the 20th Century. https://www.feynmanlectures.caltech.edu/

In this lab, we are not going to implement a continuous model[2]. It will be a discrete **2D space, a grid**… again, not unlike the line of cells in the Nagel and Rasmussen car model (or the 1D random walk), but a 2D grid for a wondering particle. Let's rephrase the problem:

Consider a particle in a 2D grid, centred at an origin (the middle of the grid). At each time step, the particle is 'kicked' to a new position of length *l = 1*. As we are in a grid, and as the length can only be 1, we can picture the possible moves the particle can take from its current position - *i,j*:

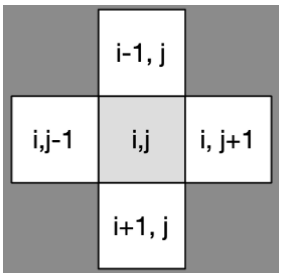

Figure 2: von Neuman 'neighbourhood'.

As time unfolds, the 'particle' (use a 'graphic' for that), output on square grid of size *L*, will be seen to drift as the following example shows:

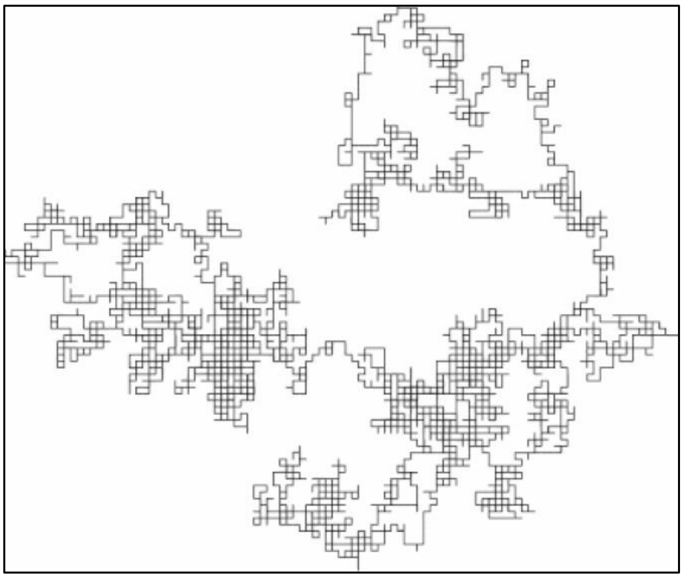

Figure 3: Accumulative result of a random walk constrained to a von Neuman 'hop' at time t.

[2] …but if you were interested in seeing how that can be done, take a look at 145-148 in Landau et al (2007)

Task

Implement a 2D random walk, with von Neuman neighbourhood constraints, and output your results to the console. Use whatever symbols you want to represent the particle, and the empty space, and and use colours, if you like

Tips

Graphics

You could take a look at the latest Nagel and Rasmussen car model implementation for how to include colour output to the console, or just use basic prints of your chosen symbols… up to you.

Random walk tip 1

Notice that as you are outputting the whole path (rather than just the current position, as in the previous coin tossing example, storing the current position alone isn't enough. You could use a list to store the path.

Random walk tip 2

Notice, as the particle moves around it will often cross over its own path. Hence, if you use a list, you will have duplicate values of positions.

Random walk tip 3

We are only plotting the graphics of the path, rather than doing anything with the path itself. So, maybe start with a list representation, and check if it is true that you get duplicates. Otherwise, to reduce the need for duplicates, you could store the current position in some other data type. Check out the Python3 **set**.

Acknowledgements/References

1) Figure 1: https://www.feynmanlectures.caltech.edu/I_41.html

2) Feynman Lectures on Physics: https://www.feynmanlectures.caltech.edu/

3) Landau, R.H, Paez, M. J., Bordeianu, C.C (2007) *Computational Physics: Problem Solving with Computers*. WILEY-VCH, Weinheim.


~~~~
~~~~