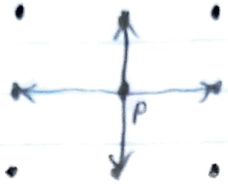
Project 2 - Self avoiding walks of len n

Say we have a $n \times n$ grid and an origin point p



the number of walks of length 0 is 1

the number of walks of length 1 is 4

The number of walks of length 2 is 16
The number of walks of length n is 4^n

Let's put some restrictions on these walks. Say p is the origin (0,0).
Any walk containing a point $> y=1$ or $< y=-1$ is a failure
Any walk containing at least 2 of the same points fails
All other walks succeed.

Create a DFA where each state is a walk (a set of points, in order).
If the walk contains $x > 1$ or $y < -1$, ignore it
If the walk contains 2 of the same, ignore it
Otherwise, count the walk.
Then, take that count & apply it to the next iteration.

This way, we keep a list of valid walks of length n, and use it to determine the list of valid walks of length n+1.