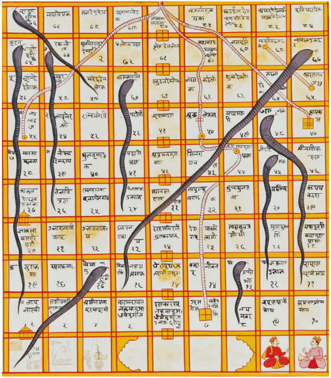
Games, graphs, and machines



August 30, 2024

The board

```
[ 0  1  2  3  4  5  6  7  8  9]
[10 11 12 13 14 15 16 17 18 19]
[20 21 22 23 24 25 26 27 28 29]
[30 31 32 33 34 35 36 37 38 39]
[40 41 42 43 44 45 46 47 48 49]
[50 51 52 53 54 55 56 57 58 59]
[60 61 62 63 64 65 66 67 68 69]
[70 71 72 73 74 75 76 77 78 79]
[80 81 82 83 84 85 86 87 88 89]
[90 91 92 93 94 95 96 97 98 99]
```

The transition matrix

We create the transition matrix.

```
A = matrix(QQ,100,100)
for i in range(0,100):
    for j in range(0,6):
        jump = i+j+1
        if (jump < 100):
            A[i,jump] = 1/6
        else:
            # We have to decide what to do if we cross 100.
            # Let us loop back to the beginning
            jump = jump - 100
            A[i,jump] = 1/6
```

The first row

Let us look at the first row.

```
pp_prob(A[0])
```

```
[0.00 0.17 0.17 0.17 0.17 0.17 0.17 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
```

The 96th row

```
pp_prob(A[96])
```

```
[0.17 0.17 0.17 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.17 0.17 0.17]
```

Steady state

What will happen to the powers?

Steady state

Let us verify.

```
pp_prob((A^(10000))[0])
```

```
[0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010]
[0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010]
[0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010]
[0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010]
[0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010]
[0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010]
[0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010]
[0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010]
[0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010]
[0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010 0.010]
```

Let us put some ladders/snakes

```
put_ladder(13,60,A)
```

Let us look at the result.

```
pp_prob(A[10])
```

```
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.17 0.17 0.00 0.17 0.17 0.17 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.17 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
[0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
```

Some more ladders/snakes

```
put_ladder(80,20,A)
put_ladder(96,63,A)
put_ladder(52,87,A)
put_ladder(21,35,A)
```

Steady state

```
pp_prob((A^(1000))[0])
```

```
[0.0093 0.0088 0.0081 0.0095 0.0093 0.0092 0.0090 0.0090 0.0090
[0.0091 0.0091 0.0091   0.00 0.0076 0.0073 0.0070 0.0067 0.0063
[ 0.020   0.00 0.0077 0.0078 0.0080 0.0083 0.0087 0.0068 0.0079
[0.0079 0.0079 0.0078 0.0077 0.0079  0.017 0.0093 0.0096 0.0099
[ 0.011  0.011  0.010  0.010  0.010  0.010  0.010  0.010  0.010
[ 0.010  0.010   0.00 0.0087 0.0084 0.0080 0.0076 0.0072 0.0066
[ 0.017 0.0090 0.0091  0.022  0.012  0.013  0.014  0.013  0.014
[ 0.013  0.013  0.014  0.014  0.014  0.014  0.014  0.014  0.014
[  0.00  0.011  0.011  0.011  0.010 0.0094 0.0087  0.021  0.012
[ 0.012  0.012  0.013  0.014  0.012  0.013   0.00  0.011  0.010
```

Start somewhere else?

```
pp_prob((A^(1000))[1])
```

```
[0.0093 0.0088 0.0081 0.0095 0.0093 0.0092 0.0090 0.0090 0.0090
[0.0091 0.0091 0.0091   0.00 0.0076 0.0073 0.0070 0.0067 0.0063
[ 0.020   0.00 0.0077 0.0078 0.0080 0.0083 0.0087 0.0068 0.0079
[0.0079 0.0079 0.0078 0.0077 0.0079  0.017 0.0093 0.0096 0.0099
[ 0.011  0.011  0.010  0.010  0.010  0.010  0.010  0.010  0.010
[ 0.010  0.010   0.00 0.0087 0.0084 0.0080 0.0076 0.0072 0.0066
[ 0.017 0.0090 0.0091  0.022  0.012  0.013  0.014  0.013  0.014
[ 0.013  0.013  0.014  0.014  0.014  0.014  0.014  0.014  0.014
[  0.00  0.011  0.011  0.011  0.010 0.0094 0.0087  0.021  0.012
[ 0.012  0.012  0.013  0.014  0.012  0.013   0.00  0.011  0.010
```

Start somewhere else?

```
pp_prob((A^(1000))[35])
```

```
[0.0093 0.0088 0.0081 0.0095 0.0093 0.0092 0.0090 0.0090 0.0090
[0.0091 0.0091 0.0091   0.00 0.0076 0.0073 0.0070 0.0067 0.0063
[ 0.020   0.00 0.0077 0.0078 0.0080 0.0083 0.0087 0.0068 0.0079
[0.0079 0.0079 0.0078 0.0077 0.0079  0.017 0.0093 0.0096 0.0099
[ 0.011  0.011  0.010  0.010  0.010  0.010  0.010  0.010  0.010
[ 0.010  0.010   0.00 0.0087 0.0084 0.0080 0.0076 0.0072 0.0066
[ 0.017 0.0090 0.0091  0.022  0.012  0.013  0.014  0.013  0.014
[ 0.013  0.013  0.014  0.014  0.014  0.014  0.014  0.014  0.014
[  0.00  0.011  0.011  0.011  0.010 0.0094 0.0087  0.021  0.012
[ 0.012  0.012  0.013  0.014  0.012  0.013   0.00  0.011  0.010
```

How to establish the Perron–Frobenius property?

- Want: some n such that there is a path of length n between every two states.

How to establish the Perron–Frobenius property?

- Want: some n such that there is a path of length n between every two states.

- Check:
  1. There is a path between every two states (strongly connected).
  2. Pick a vertex v. Suppose there are directed cycles based at v of length a and b such that

     $$\gcd(a, b) = 1.$$

     Then the Perron–Frobenius hypothesis is satisfied.

How to establish the Perron–Frobenius property?

- Want: some n such that there is a path of length n between every two states.

- Check:
    1. There is a path between every two states (strongly connected).
    2. Pick a vertex v. Suppose there are directed cycles based at v of length a and b such that

    $$\gcd(a, b) = 1.$$

    Then the Perron–Frobenius hypothesis is satisfied.

Why does this work?

Theorem
Let a, b *be positive integers such that* $\gcd(a, b) = 1$*. Any integer* $> ab$ *can be written as a sum of* a*'s and* b*'s.*

Why does this work?

Theorem
Let a, b be positive integers such that $\gcd(a, b) = 1$. Any integer $> ab$ can be written as a sum of a's and b's.

Consequence:

Let n be the number of vertices.
Take any $N > ab + 2n$.
Can find a path of length N between pair of vertices.