Arnold's cat map

Background

An important feature of chaos is **space folding**. This process allows several key aspects of chaos, most importantly divergence of nearby trajectories (the "butterfly effect"). This is a difficult concept, but we are going to look at a nice simple example of a space folding map known as "Arnold's cat map".

The idea of Arnold's cat map is to stretch space out and then refold it in on itself in a way that fills it back in. Consider a coordinate in 2d in the unit square (x, y). The Arnold cat map takes this coordinate and moves it according to

$$(x_{\text{new}}, y_{\text{new}}) = (2x + y, x + y) \mod 1 \equiv M(x, y)$$

The part in the brackets does the stretching of space, and then the "mod 1" places the coordinate back onto the unit square. Taking the four corners of the unit square, $p_1 = (0,0)$, $p_2 = (1,0)$, $p_3 = (0,1)$, $p_4 = (1,1)$ we see

$$M(p_1) = (2 \times 0 + 0,0 + 0) \mod 1 = (0,0)$$

$$M(p_2) = (2 \times 1 + 0,1 + 0) \mod 1 = (2,1) \bmod 1 = (0,1)$$

$$M(p_3) = (2 \times 0 + 1,0 + 1) \mod 1 = (1,1) \bmod 1 = (1,1)$$

$$M(p_4) = (2 \times 1 + 1,1 + 1) \mod 1 = (3,1) \bmod 1 = (1,0)$$

And hopefully you can see that the original square will be mapped back onto a square but with some amount of mixing!

An interesting application of the cat map is to images. If you have an $N \times N$ image, then instead of mod 1, you use mod N and integer coordinates. One still observes space folding as needed, but the surprising feature (because it is discrete) is that after some number of iterations the image will return!

Task

Take an $N \times N$ image. Go through each pixel and move the value of the pixel (either grey scale or RGB) and place it in a new position according to the cat map. If the current pixel is at (x, y), then

$$(x_{\text{new}}, y_{\text{new}}) = (2x + y, x + y) \mod N$$

Remembering that this is the coordinate, but you move the colour itself.

Once you have moved all the pixels according to this, you have applied the map once! Then repeat this many times and see if you can spot when it matches back up to the original image.