Problem F.2

Fractals

Due Date: 5/3/2019
Folder: `FinalProject`
File Name: `F2_Fractal_Name.py`
Points: 20 points

Problem Background

Start with a shape, say a triangle. Consider each point on the triangle to be defined by a set of vectors

$$S_0 = \{ \begin{bmatrix} x \\ y \end{bmatrix} : \text{the point } (x, y) \text{ is a point on the triangle}\}.$$

Different values for t correspond to different places on the triangle. To generate a fractal, we will perform 3 transformations on all of these vectors to generate a new set of points S_1. Here is one example of these transformations. For each vector $v \in S_0$, we generate 3 new vectors w_1, w_2, w_3

$$\begin{aligned} w_1 &= \begin{bmatrix} \frac{1}{4} & 0 \\ 0 & \frac{1}{4} \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ w_2 &= \begin{bmatrix} \frac{1}{4} & 0 \\ 0 & \frac{1}{4} \end{bmatrix} v + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} \\ w_3 &= \begin{bmatrix} \frac{1}{4} & 0 \\ 0 & \frac{1}{4} \end{bmatrix} v + \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \end{aligned}$$

All three of these new vectors are then added to a new set S_1. If this process is performed for all vectors $v \in S_0$, we generate a new set S_1. When all the points in S_1 are plotted we see three new triangles. We then perform this transformation yet again on all vectors in S_1 to generate a set S_2. By continuing this process repeatedly, we obtain a fractal image.

Your project will be to generate all the S_i sets up to some maximum value for i. The plot each one to see how the image changes. In general, the transformations shown above can be written as

$$\begin{aligned} w_1 &= A_1 v + b_1 \\ w_2 &= A_2 v + b_2 \\ w_3 &= A_3 v + b_3 \end{aligned}$$

where A_i are matrices and b_i are vectors. You will also be changing the matrices and vectors to see how it affects the fractal image. In addition, you will change the initial image to see how it affects the resulting fractal.

Program Criteria

Write a program that does the following:

- Has an input variable `N` to determine the 2^N number of bits on the circle.
- Determine how many binary circles have all distinct N length subsequences. Be sure to ignore rotations of the circle. (That is, all 2^N rotations of a circle with distinct N length subsequences will count as just one such binary circle.)
- Print out the number of binary circles, as well as the elements of the circles, starting at the top and going in clockwise order.

Deliverables

Place the following in a folder named `FinalProject` in your repository:

- A Python file `F2_Fractal_Name.py` that satisfies the program criteria.
- A PDF file `F1_Prob265_Name.pdf` that describes how your program works. This should be a description of how you went about solving this problem. You should go into some detail about your solution method, but I don't want to see something about every `if` statement and `for` loop.