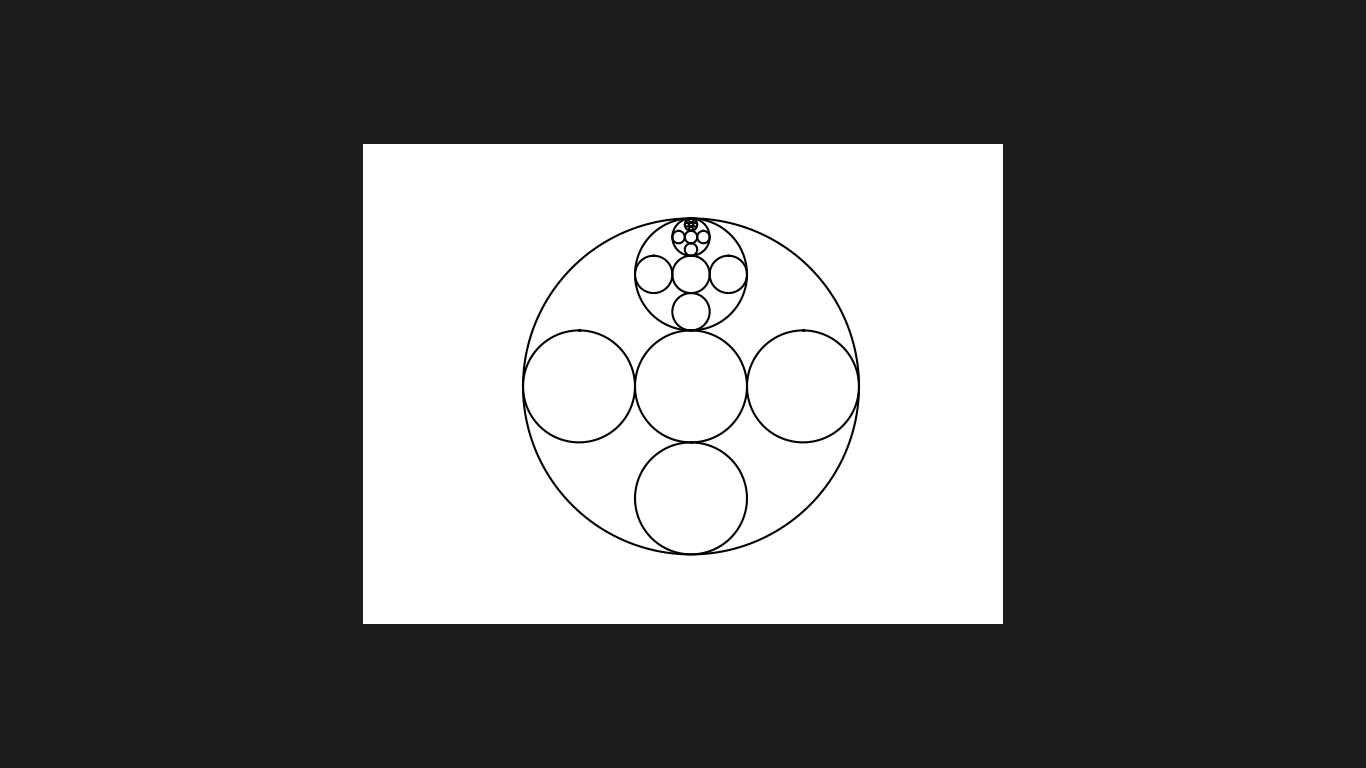
Solomon Davis Lab 1 Report

CS 2302 - MW 1:30 Spring 2019

The problem that I trying to solve is how to draw figures using recursive calls. The figures I attempted to draw on python were circles within 5 circles, squares rotating across each corner of the previous square, a tree structure drawing more trees on each branch, and drawing circles with insides 1 big circle.

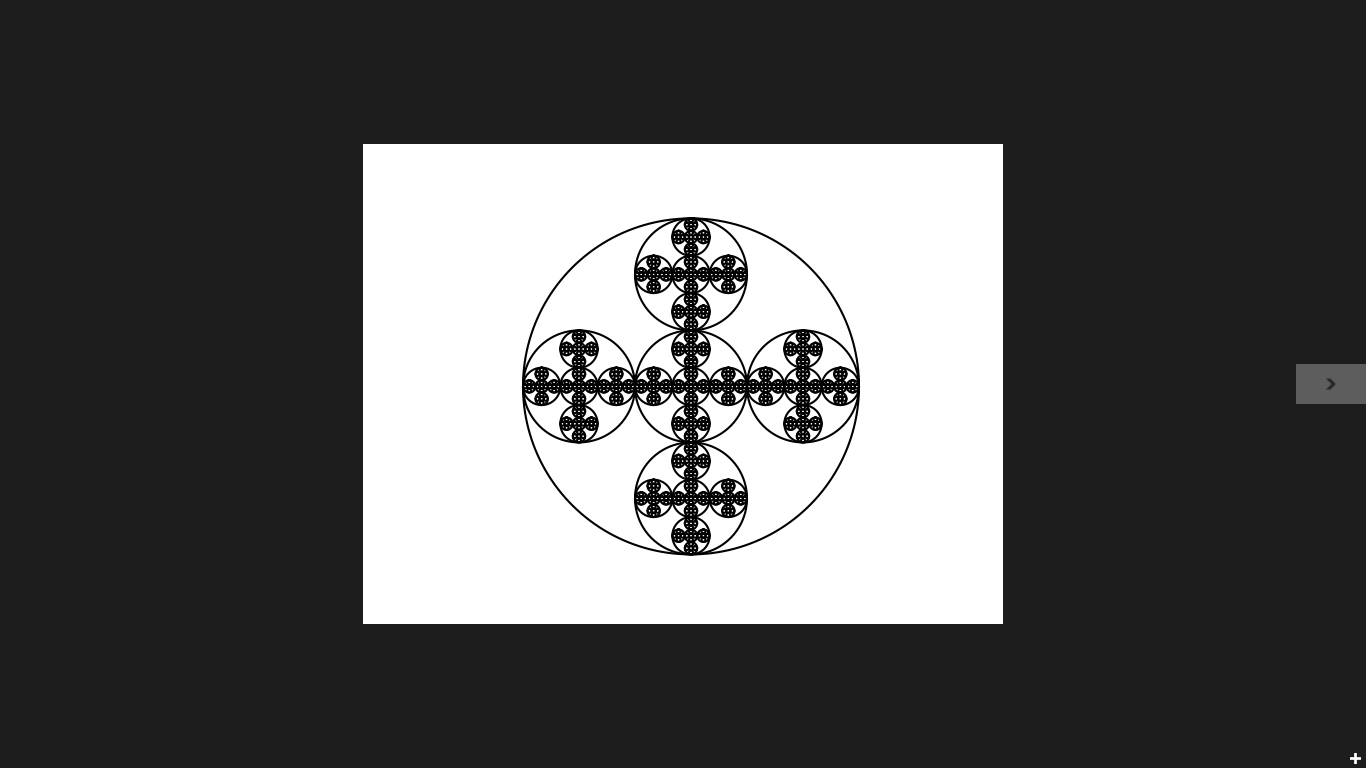
For my solution I wanted to recursively call the specific function however many times I needed to draw the shape. So, for example if I needed to draw 4 circles for the base figure the recursive code was called 4 times. I worked on the first recursive call for each figure because I knew it I figured out the correct code for one recursive call then the rest of the recursive calls would work.

My first thought was to draw the number of recursive calls for the number of shapes. So, for figure 4 I thought I needed 4 different recursive calls. When I did this for the circles. I got the figure below where the circles wouldn’t rotate. This is pictured below.

Running Time: 0.18750262260437012 seconds

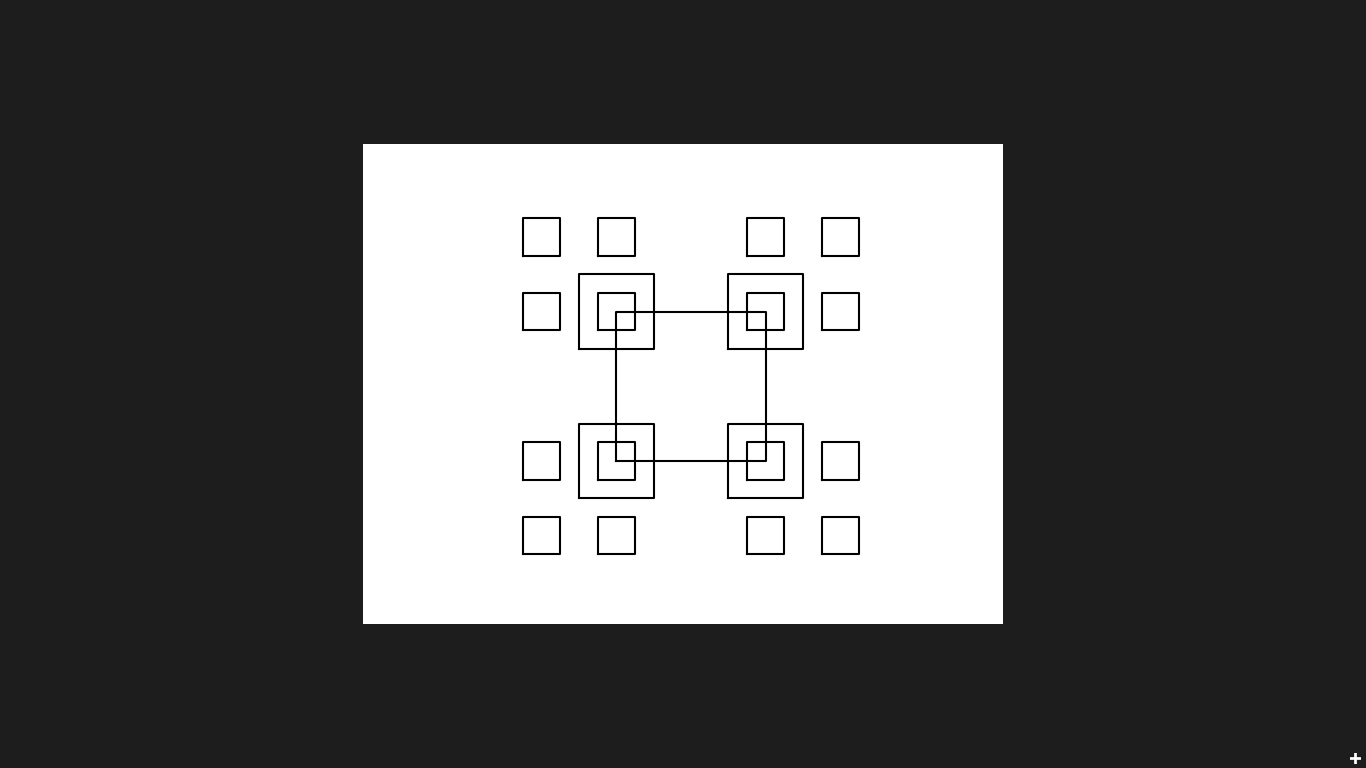
After changing the code so each recursively call is the same and not different I received this.

Running Time: 2.7661030292510986 seconds



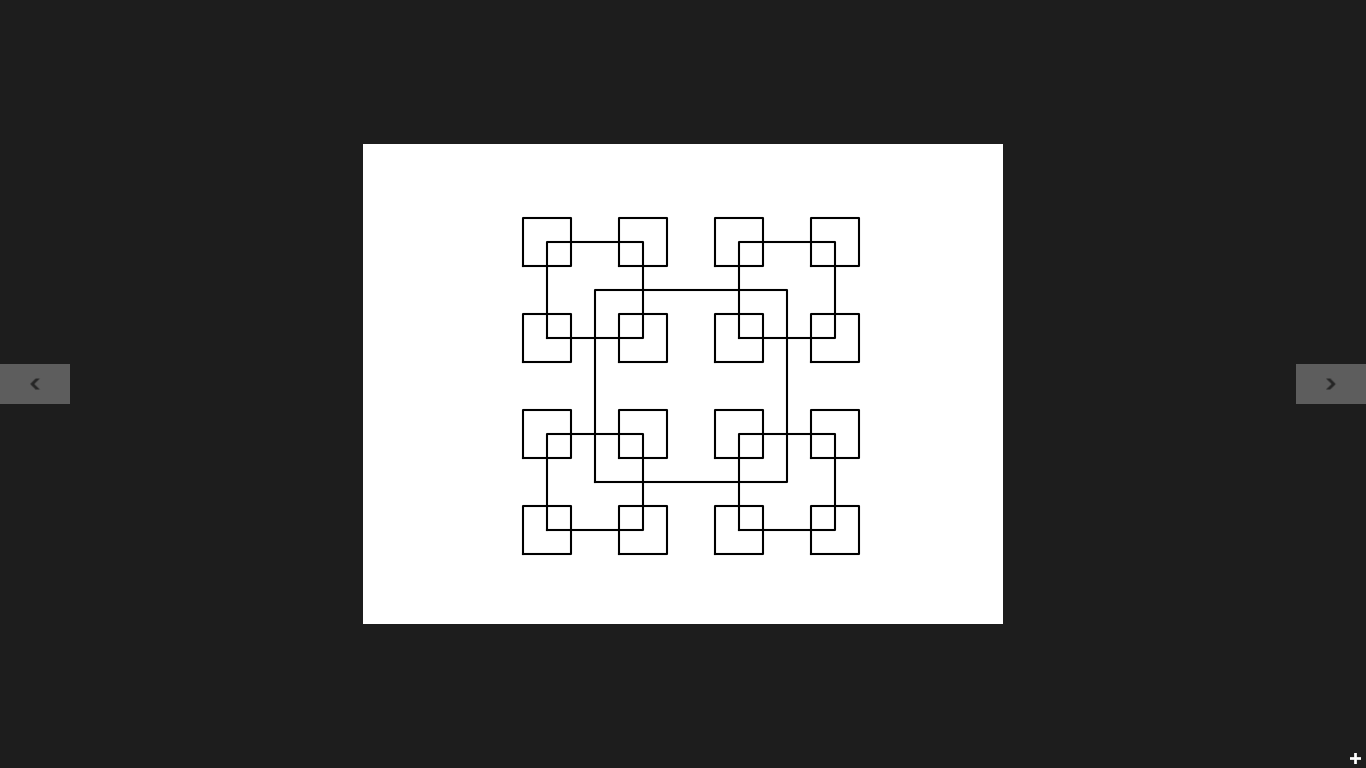
To get this figure I had drawn the five main circles in certain positions. For the circle in the middle the center is the same as the base circle. For the circle on the left I had to change the x coordinate to the x coordinate of the base circle minus the 2/3 radius of the base circle, while the y coordinate remained the same as the base circle. For the circle on the right of the middle circle I changed the x coordinate to the x coordinate of the base circle plus the 2/3 radius of the base circle while the y coordinate remained the same as the base circle. For the top circle I changed the y coordinate to the y coordinate of the base circle plus the 2/3 radius of the base circle, while the x coordinate remained the same as the base circle. For the bottom circle I changed the y coordinate to the y coordinate of the base circle minus the 2/3 radius of the base circle, while the x coordinate remained the same as the base circle. Once I figured out how to draw figure 4 I attempted to draw the square figure. One problem I had was getting the squares to be drawn correctly after a certain number of squares. For example, I could get the program to draw correctly when n = 2 but when n was anything greater than 2 I would get an oddly drawn figure. An example of this is pictured below. This is where n is equal is to 3.

Running Time: 0.2343759536743164 seconds

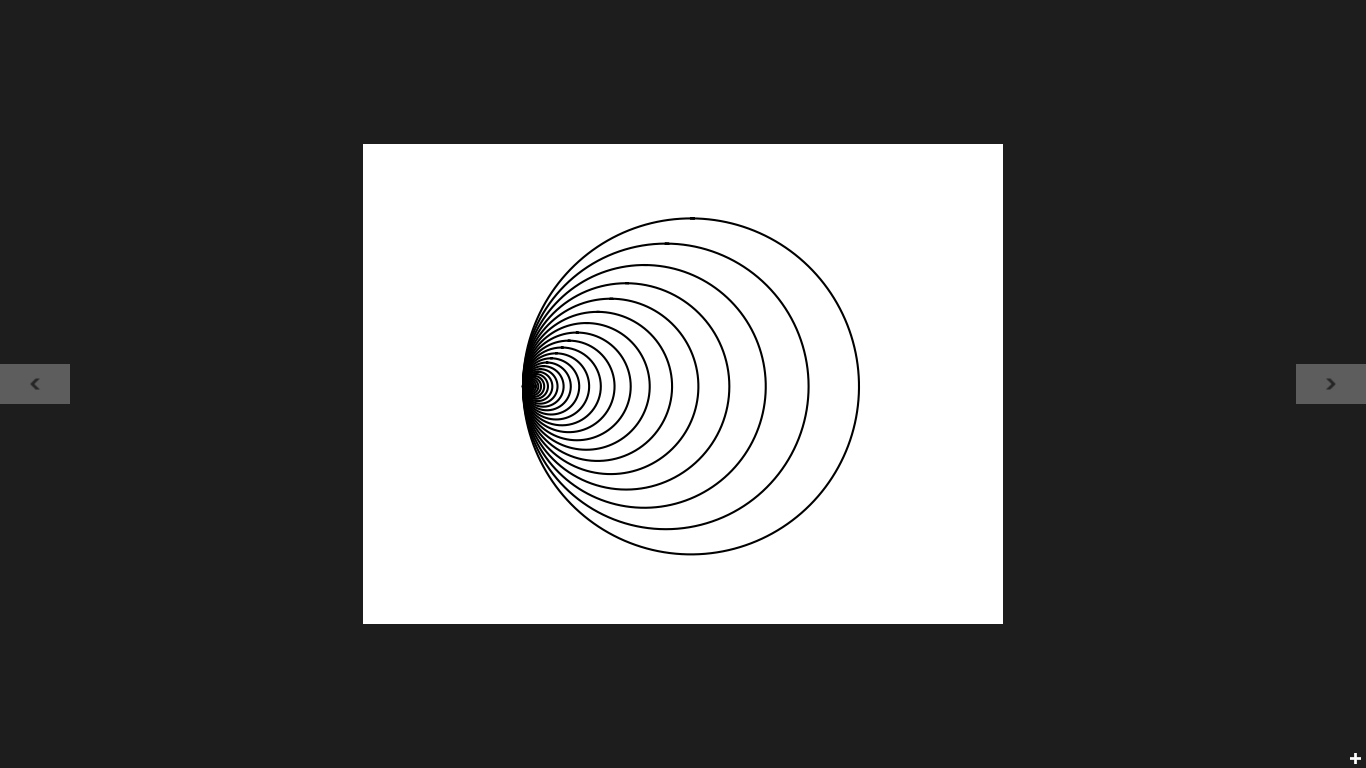


To get the squares working correctly I divided the coordinates of the previous square in half, so the squares are a fourth of the size of the previous squares and subtracted that value from a fourth of the second and forth coordinate in the variable p to correctly place to the squares in the correct positions. After doing this I received this figure. N=2 for this pic

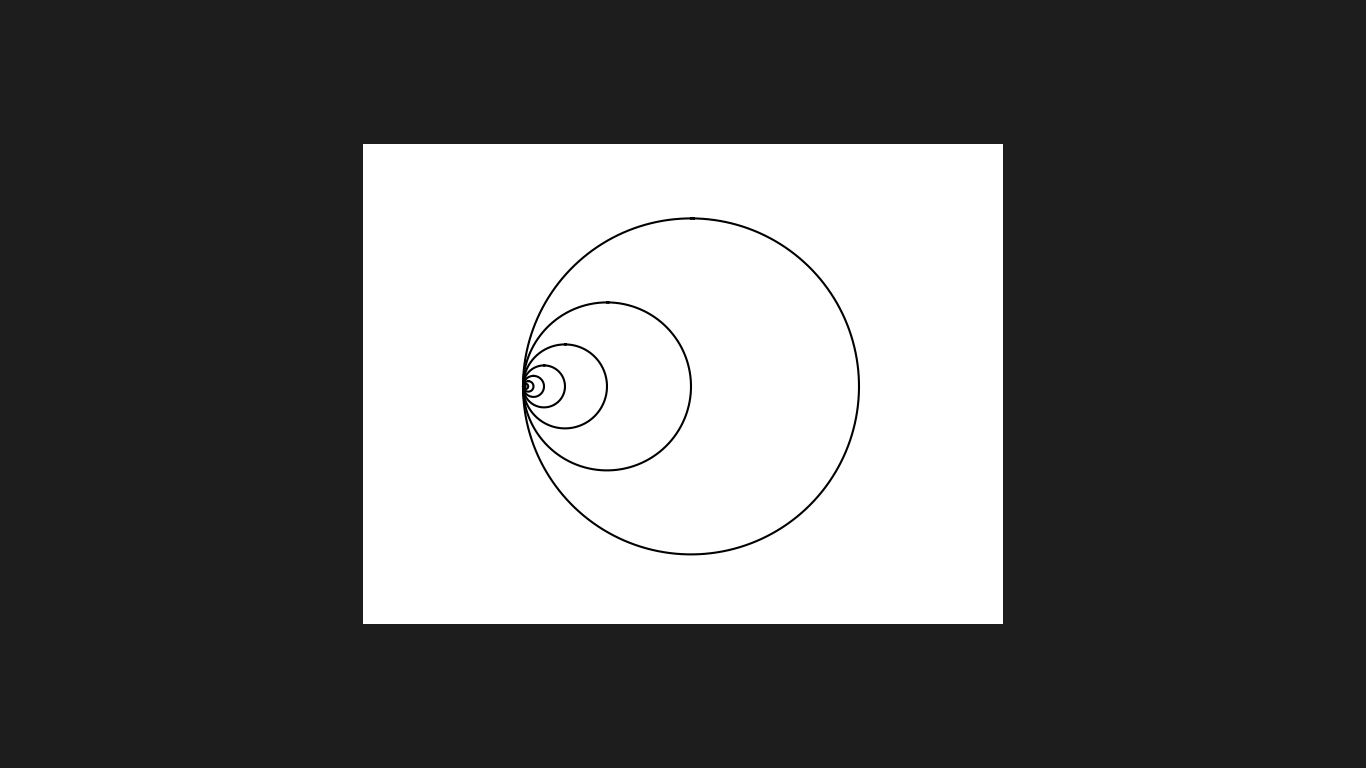
Running Time: 0.5000030994415283 seconds



I didn’t have too many problems out of figure 2 I simply moved the center of the previous circle by keeping the original y coordinate of the base circle and having the x coordinate being the radius multiplied by w, which was the percentage of the size of the previous circle thus causing the circles to shrink in size. Here are some pictures of my output.

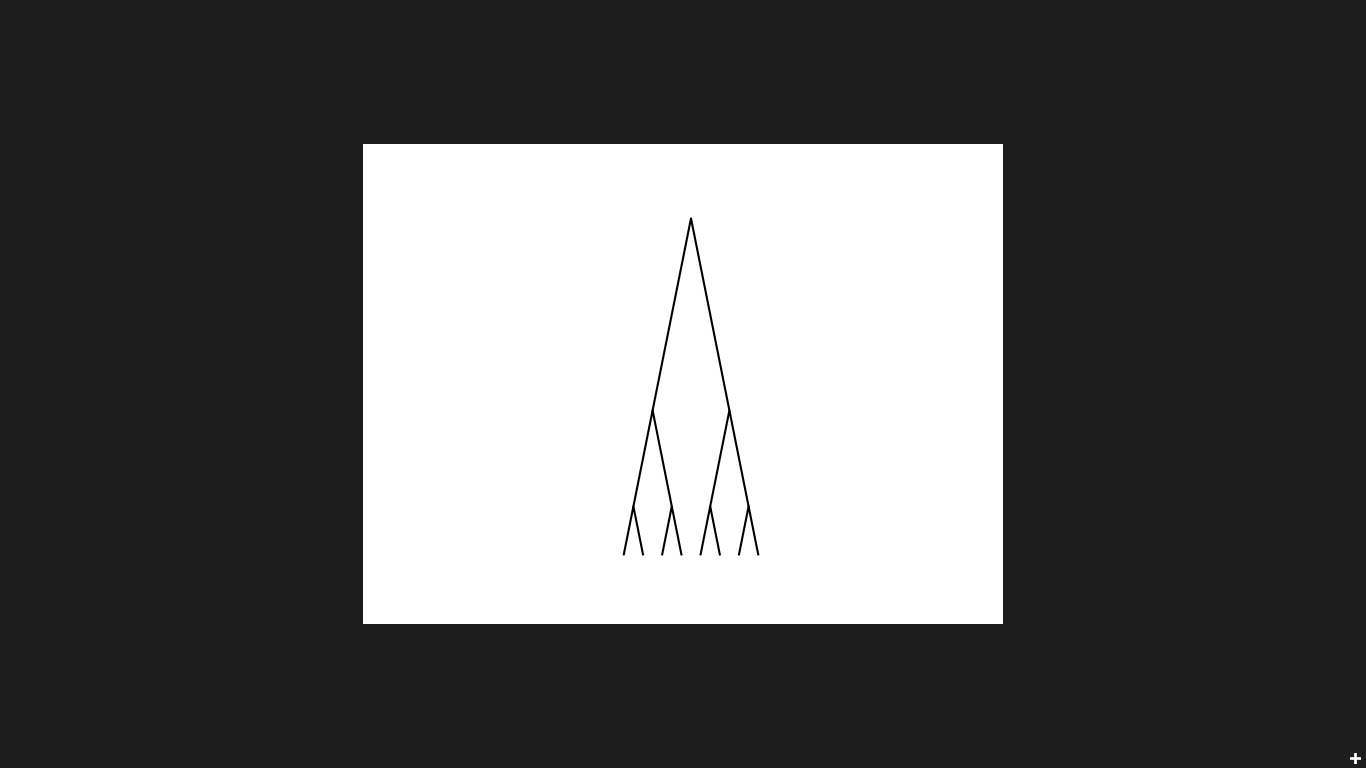
Running Time: 0.20341038703918457 seconds

Running Time: 0.2680675983428955 seconds

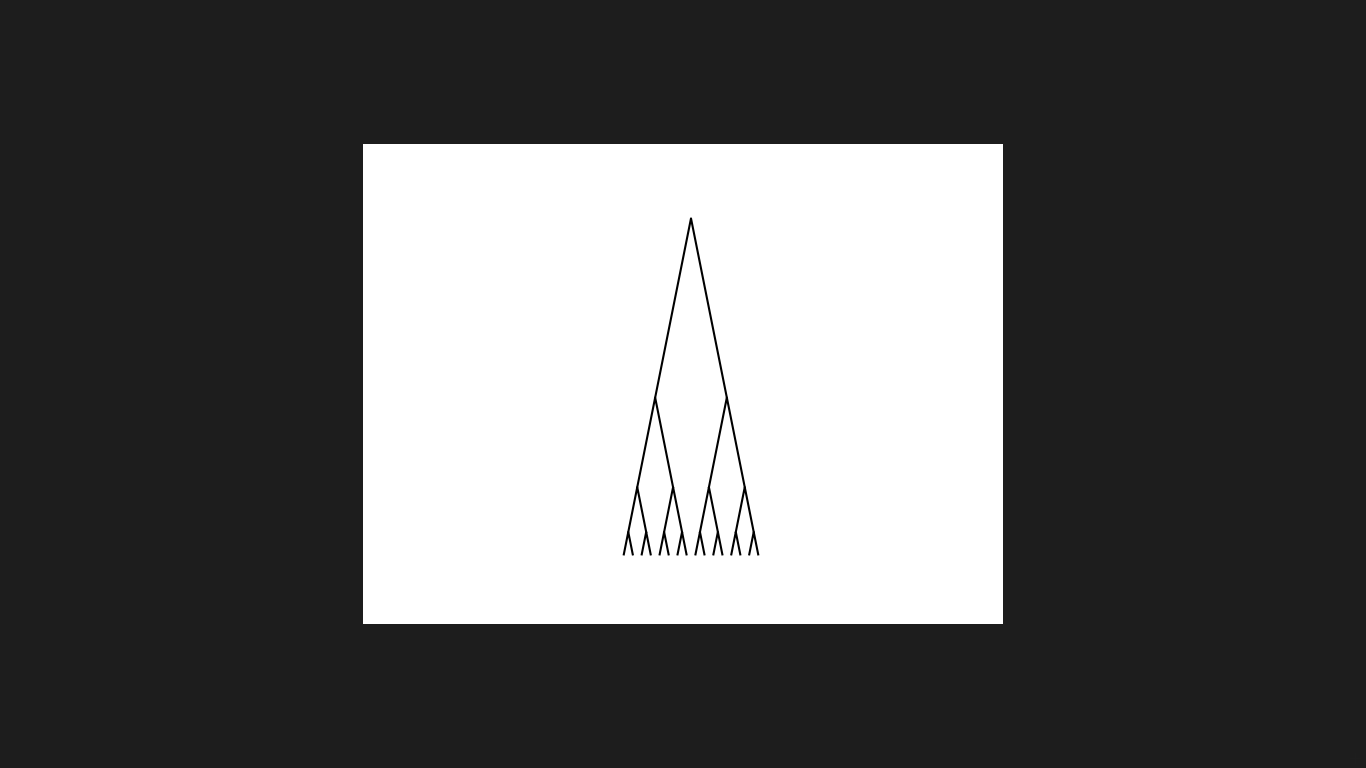


For figure 3 I didn’t have too much trouble. To solve this, I pictured drawing a triangle that doesn’t connect between the first and last point. On each branch I have another tree figure that forms that is half the size of the previous tree. The figures are shown below.

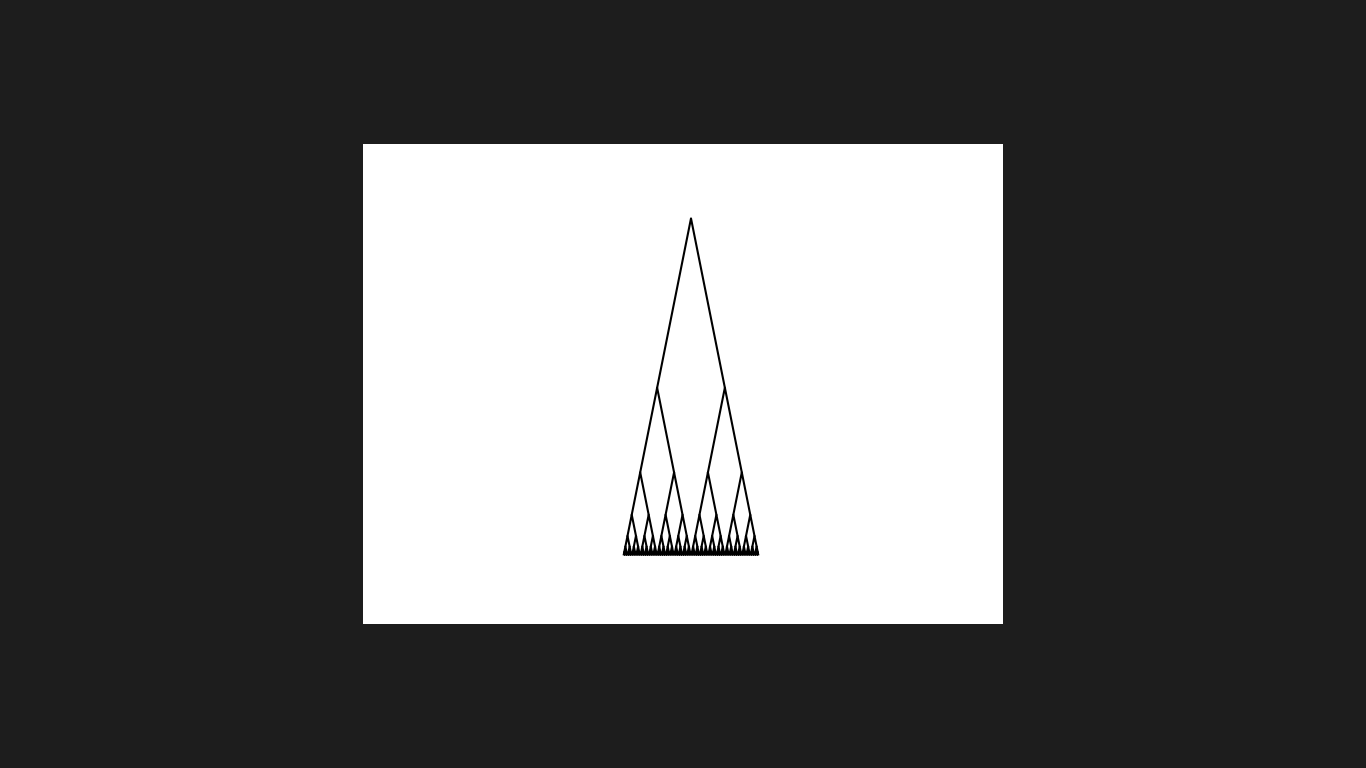
Running Time: 0.12508773803710938 seconds



Running Time: 0.12655162811279297 seconds



Running Time: 0.4843721389770508 seconds

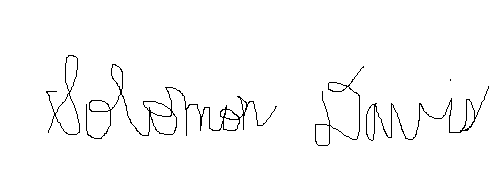


In this lab I learned how to draw figures in python. I also learned how to use recursion to draw figures. In conclusion this lab has helped me grasp recursive concepts better and made a better programmer in python.

Source Code: Posted on GitHub

Academic Service Certificate:

I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class.



Solomon Davis