

National University of Singapore  
School of Computing  
CS1010X: Programming Methodology  
Semester II, 2019/2020  
**Mission 5 - Side Quest**  
**Kochize**

Release date: 22 February 2020

**Due: 15 March 2020, 23:59**

## Required Files

- sidequest05.2-template.py
- hi\_graph\_connect\_ends.py

## Background:

Nature hides many beauties, and one of these, as you have seen, are fractals. Grandmaster Ben now invites you to use your new-found skills to conjure up an image that is not quite so simple: the snowflake fractal, more commonly known as the Koch snowflake.

Upon successful completion of this side quest, you will earn the “Snowflake Catcher” achievement.

## Information:

The Python source file has been renamed to `hi_graph_connect_ends.py` and modified to include the function `connect_ends` which should have been previously coded. You may now use the function directly from this source file.

For your convenience, the template file `sidequest05.2-template.py` contains a line to load the Python source file `hi_graph_connect_ends.py`. Use the template file to answer the questions.

This side quest consists of **two** tasks.

## Task 1: (4 marks)

We begin by drawing only part of the snowflake.

Figure 1 shows the first few approximations to the Koch curve, where we stop after a certain number of levels: a level-0 curve is simply a straight line; a level-1 curve consists of four level-0 curves; a level-2 curve consists of four level-1 curves, and so on. The figure also illustrates a recursive strategy for making the next level of approximation: a level- $n$  curve consist of three parts. The first part is a level- $(n-1)$  curve. The second part

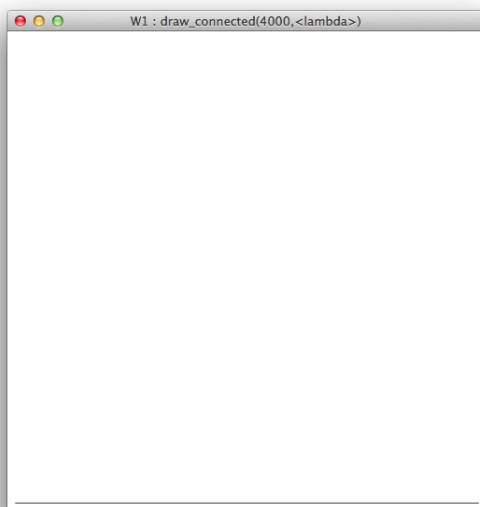
is two level- $(n - 1)$  curves, one is rotated by  $\pi/3$  and the other is rotated by  $-\pi/3$ . The last part is another level- $(n - 1)$  curve.

Define a function `kochize` such that the evaluation of

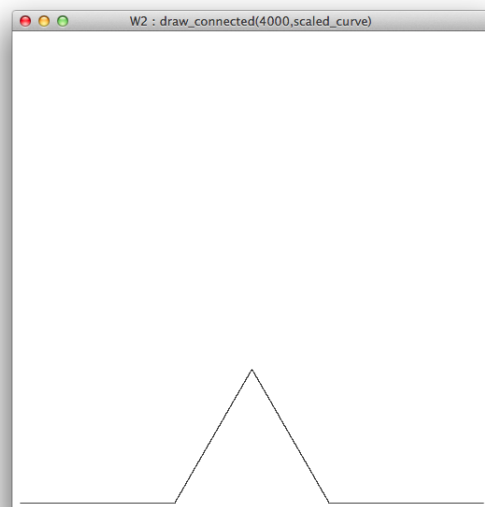
```
show_connected_koch(level, num_points)
```

will show `num_points` connected points of the level `level` koch curve in window starting with a unit\_line, where `show_connected_koch` is defined as:

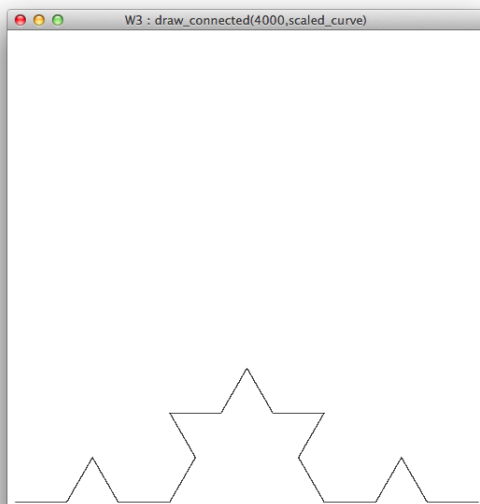
```
def show_connected_koch(level, num_points):
    draw_connected(num_points, kochize(level))
```



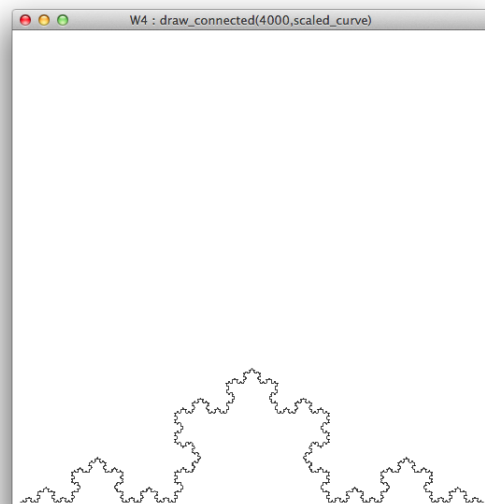
`show_connected_koch(0, 1000)`



`show_connected_koch(1, 1000)`



`show_connected_koch(2, 4000)`



`show_connected_koch(5, 4000)`

Figure 1: Samples for Koch Curve.

**Task 2: (4 marks)**

By completing **Task 1**, you have one third of your snowflake in hand. Now construct your complete snowflakes using a level-5 Koch curve and tools you have learned in previous missions. Precisely, define `snowflake` such that `draw_connected_scaled(10000, snowflake())` displays your snowflake within the viewport.

[Note: `draw_connected_scaled` scales proportionally and translates your curve to the center.]

An example is shown in Figure 2.

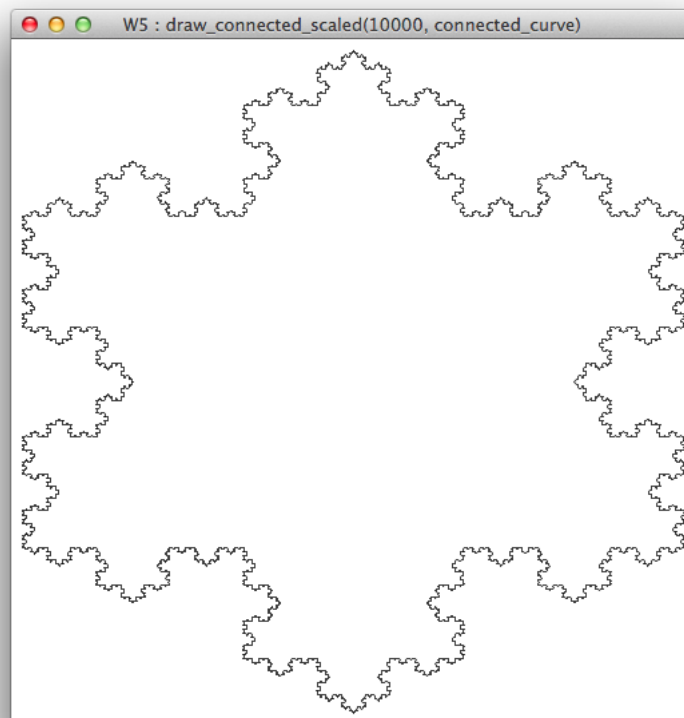


Figure 2: `draw_connected_scaled(10000, snowflake())` displays the snowflake above.