2019-3-6

Lab 07 - Due at end of class

As usual, make a new branch in your assignments repo called ‘lab07’ and do your work in there. commit your changes to github and make a pull request assigned to me.

1. 50 pts

Do all the questions from the Lotka-Volterra Models Discrete Time and Continuous time simulations lab. (included in the class materials)

I added the plotting in the discrete time script, if you find it as useful as I did, I will leave it to you to duplicate it in the second script to better visualize what happens in the simulations.

1. 50 pts

Memiozation and recursive algorithms

Fibbonacci is the classic algorithm to show multiple recursion. There is another very similar algorithm called the Lucas series (predictably named after the mathematician François Édouard Anatole Lucas (1842–91)) where the numbers are generated in the same algorithm as the Fibbonacci, but with different initial starting conditions. That sequence is generated as follows:

L(n) = {

2: if n == 0

1: if n == 1

L(n-1) + L(n-2) : if n > 1

}

If you look at the page at <https://www.python-course.eu/python3_recursive_functions.php>

Immediately after the call graph for fibbonacci(5), there is an example implementation of a simple memiozed version of the fibbnacci function.

The basic premise is to create a dictionary of known answers that can be returned immediately. For answers that are not in the dictionary, the answer must be computed but then stored in the dictionary. For both of these sequences, the initial dictionary would cover the inputs for 0 and 1.

Your task is to implement functions to generate both the Fibbonacci and Lucas sequences, one set using the standard method, the other set using memoization. So 4 total functions.

Measure how much time it takes to print the Fibbonacci and Lucas sequences from 1 to 40 using both methods.