Problem N.1

Collatz Conjecture

Due Date: 3/12/2019
Folder: `NumberTheory`
File Name: `N1_Collatz_Name.py`
Points: 2 points

Learning Objectives

- Review basic programming skills
- "Semi prove" a conjecture
- Find sub-theorems that can be proven from a conjecture

Problem Background

The Collatz conjecture is one of those quintessential number theory problems that is easy to state but incredibly difficult to prove. In fact, this particular problem is so difficult that it has not yet been proven, despite that fact that the conjecture was stated by Lothar Collatz in 1937. Paul Erdős said of this problem that "Mathematics may not be ready for such problems".

The problem involves a sequence of numbers a_n which are defined iteratively. We start with some positive integer n and make that the first term of the sequence, that is we set $a_0 = n$. Each term is then defined by considering the term before it, using the following relation,

$$a_{i+1} = \begin{cases} a_i/2 & \text{if } a_i \text{ is even} \\ 3a_i + 1 & \text{if } a_i \text{ is odd} \end{cases}$$

For instance, if we set the first term to be $a_0 = 6$, then the next term would be $a_1 = a_0/2 = 3$, since 6 is even. The term $a_2 = 3a_1 + 1 = 10$ since 3 is odd. Continuing like this, the sequence would look like

$$6 \to 3 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$$

Note that after we reach 1, the sequence will repeat forever between $1 \to 4 \to 2 \to 1$. The Collatz conjecture can be stated as: *The sequence of terms will eventually reach 1, no matter what initial positive integer is chosen for* a_0.

The Collatz has been shown to be true for all integers less than $87 \cdot 2^{60}$. While this is a very large number, it does mean the conjecture is true. There may still exists an integer for which the sequence either diverges or loops periodically in a way that does not contain 1. It will only take **one** starting value for this sequence that diverges or does not ever reach 1 to prove the conjecture is false. To prove it is true for all positive integers is a problem that has eluded mathematicians for almost a century.

Programming Reminders

- Create function: `def func_name(params):`
- Loops syntax: `for ii in range(N)`
- Create empty list: `my_list = []`
- Append to list: `my_list.append(new_thing)`

Program Criteria

Write a program that does the following:

- Your program will have two input variables
  - `a0`: The initial term in the sequence.

    - ⋆ `N`: The total number of terms to compute in the sequence.

- Compute the first `N` terms in the sequence, starting with the value `a0`, and store the sequence in a list or numpy array. Stop computing terms early if your terms reach 1.
- Print out many terms it took to reach 1, with appropriate descriptive text. If you never reached 1, print out a message that you did not reach 1 after `N` terms.

Deliverables

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

- A Python file `N1_Collatz_Name.py` that satisfies the program criteria.