Assignment #8

Math 1800: Mathematical Programming in Python

Instructions: Choose 3 of the following problems to work on. Submit your responses as Python text files, with the extension `.py`. Each file should include your name and the problem number. This problem set is due Thursday, March 23, by midnight.

- *Problem 8.0:* The "Lollatz transformation" is defined by

$$L(n) = \begin{cases} \frac{2n}{3} & \text{if} \quad \text{mod } (n,3) = 0 \\ \frac{4n-1}{3} & \text{if} \quad \text{mod } (n,3) = 1 \\ \frac{4n+1}{3} & \text{if} \quad \text{mod } (n,3) = 2 \end{cases}$$

  Write a Python program `lollatz_sequence(n)` which computes the sequence of values generated by repeatedly applying the Lollatz transformation. This transformation is known to have some short cycles. It may also have some very long cycles, or even some infinite sequences. See if you can find a starting point n which can be transformed 100 times without returning to the starting value.

- *Problem 8.1:* The Lollatz transformation is actually a permutation, so for every positive integer n, there is a "preimage", that is, an integer m, such that $L(m) = n$. We can therefore define the inverse Lollatz transformation, so that $L^{-1}(n) = m$. For example, since L(6) = 4, we have $L^{-1}(4) = 6$. Write the definition of this function

$$L^{-1}(n) = \begin{cases} \frac{3n}{2} & \text{if} \quad \text{mod } (n,2) = 0 \\ \frac{?}{?} & \text{if} \quad \text{mod } (n,?) = ? \\ \frac{?}{?} & \text{if} \quad \text{mod } (n,?) = ? \end{cases}$$

  Implement your inverse Lollatz transformation as a Python code.

- *Problem 8.2:* The "Mollatz transformation" is defined by

$$M(n) = \begin{cases} \frac{n}{2} & \text{if } n \text{ is even} \\ 3n-1 & \text{if } n \text{ is odd} \end{cases}$$

  If the sequence reaches 1, it stops. Write a Python program `mollatz_sequence(n)` which computes the sequence of values generated by repeatedly applying the Mollatz transformation. Search for a starting value n which generates a sequence of at least 100 steps.

- *Problem 8.3:* For the Mollatz transformation starting at n, define $mlen(n)$ to be the length of the trajectory, that is, the number of times the transformation must be applied before the sequence reaches 1. We aren't sure that the trajectory always reaches 1, so if a sequence has taken 1,000 steps, set $mlen(n) = 1000$ and stop there.

  Write a Python program which evaluates $mlen(n)$, and use it to make a table of trajectory lengths for starting values $2 \leq n \leq 30$.

- *Problem 8.4:* The "Nollatz transformation" is defined by

$$N(n) = \begin{cases} \frac{n}{2} & \text{if } n \text{ is even} \\ n+1 & \text{if } n \text{ is odd} \end{cases}$$

  If the sequence reaches 1, it stops. Write a Python program `nollatz_sequence(n)` which computes the sequence of values generated by repeatedly applying the Nollatz transformation until you reach 1. Try your code on the starting values $2 \leq n \leq 30$.

- *Problem 8.5:* For the Nollatz transformation starting at n, define $nlen(n)$ to be the length of the trajectory, that is, the number of times the transformation must be applied before the sequence reaches 1. The Nollatz trajectory will always reach 1, so don't worry about an infinite sequence.

  Write a Python program which evaluates $nlen(n)$, and use it to make a table of trajectory lengths for starting values $2 \leq n \leq 30$.

- *Problem 8.6:* The Pollatz transformation is defined by

$$P(n) = \begin{cases} \frac{n}{2} & \text{if } n \text{ is even} \\ 5n - 1 & \text{if } n \text{ is odd} \end{cases}$$

  If the sequence reaches 1, it stops. Write a Python program `pollatz_sequence(n)` which computes the sequence of values generated by repeatedly applying the Pollatz transformation. Use this code to compute and print the Pollatz sequence for starting values $2 \leq n \leq 9$. Warning: if a sequence takes more than 100 steps, assume it's not going to reach 1, and just terminate it!

- *Problem 8.7:* For the Pollatz transformation starting at n, define $plen(n)$ to be the length of the sequence starting at n until reaches 1. If a sequence keeps iterating past 100 steps, simply stop, and set $plen(n) = 100$. Write a Python program which evaluates $plen(n)$, and make a table of $plen(n)$ for starting values $2 \leq n \leq 30$.

- *Problem 8.8:* For the Pollatz transformation starting at n, define $pmax(n)$ to be the largest value in the sequence before it reaches 1 or a cycle. Write a Python program which evaluates $pmax(n)$, and use it to make a table of trajectory maximums for starting values $1 \leq n \leq 50$.

- *Problem 8.9:* Suppose that, for the Collatz transformation, we allow the starting value n to be negative. Then all the entries in the sequence will be negative. So we also change the rules, so that the sequence will stop if it reaches -1 or +1. It is known that if we allow negative starting values, the Collatz sequence can fall into a cycle. Write a code that computes this modified Collatz sequence for any starting point, positive or negative. Search for a cycle. If you use `np.append()` to keep the sequence `C` as you go, you can actually detect if the new value *cn* closes a cycle, using a command like

```
if ( cn in C ):
  print ( 'Found a cycle!' )
  break
C = np.append ( C, cn )
```