22C:16 (CS:1210) Homework 1

Due via ICON on Friday, Feb 8th, 4:59 pm

What to submit: Your submission for this homework will consist of four files. One of them will be a pdf file called `homework1.pdf`. This will contain partial answers to Problems 2 and 3. This pdf file should start with your name, section number, and student ID. The remaining files should be called `hopto1.py`, `hopto2.py`, and `hopto3.py`. These should contain Python programs for Problems 1, 2, and 3 respectively. These files should also start with your name, section number, and student ID appearing at the top of the file as Python comments. You will get no credit for this homework if your files are named differently, have a different format (e.g., docx), and if your files are missing your information.

The Collatz Conjecture. The *Collatz conjecture*, named after someone called Luther Collatz is about a simple mathematical process. Here is a straightforward explanation copied from Wikipedia's entry on the Collatz conjecture.

> Take any natural number n. If n is even, divide it by 2 to get $n/2$. If n is odd, multiply it by 3 and add 1 to obtain $3n + 1$. Repeat the process indefinitely. The conjecture is that no matter what number you start with, you will always eventually reach 1.

This process has also been called the "Half Or Triple Plus One" or HOTPO process. For example, the HOPTO process starting from $n = 6$ results in the following sequence 6, 3, 10, 5, 16, 8, 4, 2, 1. In this homework you will write a series of programs that explore the Collatz conjecture.

1. Write a program that takes as input a natural number n and outputs the number of steps it takes the HOPTO process to reach 1, if it is started at n. For example, if your input is $n = 6$, then the output should be 8 because the number of steps it takes the HOPTO process to get to 1 starting at 6 is 8.

2. Define the *HOPTO length* of a natural number n as the number of steps it takes the HOPTO process to reach 1 starting at n. Thus the HOPTO length of 6 is 8. Write a program that reads a natural number N and finds a natural number $n < N$ with maximum HOPTO length among all natural numbers in the range 1 through $N - 1$. Your program should output both n and its HOPTO length. This program can be obtained by modifying the program you wrote for Problem 1.

   We would like you to use your program to verify a claim on Wikipedia's Collatz conjecture page that among all natural numbers less than 100 million, 63,728,127 has the maximum HOPTO length, which is 949. Wikipedia has claims for larger values of N as well: among numbers less than 1 billion, 670,617,279 has maximum HOPTO length, which is 986, and among numbers less than 10 billion, 9,780,657,630 has maximum HOPTO length, which is 1132. Can you verify these claims?

   We would like you to respond to these questions by writing a few (5-8) sentences describing your *experiments* and the results you got. It is possible that your program takes too much time for some of the Wikipedia claims. This is okay and if so you can say something like "I ran my program with N equals 10 billion overnight and my program still did not complete and so I tried it for N equals 2 billion and what I see is that..." On the other hand, you might be able to verify all of Wikipedia's claims and you should say so, while mentioning how much time (roughly) each verification took. Clearly, there is no single correct answer to this question and what we would like to see is a clear written description of your experiments and results, independent of the results.

3. We suspect that most natural numbers have a (relatively) small HOPTO length. One way to explore this is to write a program that reads a natural number N and outputs the number of natural numbers less than N that have HOPTO length at most 250, the number that have HOPTO length in the range $(250, 500]$, the number that have HOPTO length in the range $(500, 750]$, and the number that have HOPTO length more than 750. Thus your program needs to print 4 numbers, some of which could be 0. This program can also be obtained by modifying the program you wrote for Problem 1.

   Report the output your program produces when run for N equals 100 million.
