Analysis of Algorithms: Bonus project

Due date: April 8 (Thursday), 11am

This project is optional; it is worth 6 bonus points toward you final grade for the course. Note that the deadline is hard; you cannot submit your results after the due time.

Suppose that we fix some positive integer, *big-num*, and consider the following procedure:

```
LOOPER(n)   ▷ n is a positive integer
while 1 < n < big-num
    do  if n is even
            then n ← n/2
            else n ← 3n + 1
if n = 1
    then  return TRUE
    else  return FALSE
```

This procedure may return TRUE or FALSE, or it may go into an infinite loop. If n is much smaller than *big-num*, then the algorithm returns TRUE in almost all cases. Finding values of n that result in returning FALSE or infinite looping is a difficult problem.

Now suppose that we fix another positive integer, *small-num*, which is smaller than *big-num*. Your goal is to design an efficient algorithm for finding *all* values of n, no greater than *small-num*, that cause LOOPER to return FALSE or run forever. The algorithm must input *big-num* and *small-num*, and return the following information:

1. The total number of values of n, no greater than *small-num*, that cause LOOPER to return FALSE. If your algorithm finds at least one such value (that is, their total number is not zero), then it must return the smallest of such values.

2. The number of values of n, no greater than *small-num*, that cause LOOPER to run forever. If your algorithm finds at least one such value, then it must return the smallest of them.

(a) Algorithm

Give a pseudocode for the designed algorithm and estimate its asymptotic time complexity, in terms of *small-num* and *big-num*. Note that your algorithm must terminate even if LOOPER goes into an infinite loop. Also note that calling LOOPER for every number from 1 to *small-num*, and monitoring its execution, is *not* an efficient solution. For the full bonus, you need to develop a faster algorithm.

Hint: Do not try to save space; you may need a lot of memory for storing intermediate results.

(b) Programming

Implement your algorithm (in any machine language), and run it with *big-num* $= 1,000,000,000$ (one billion) and *small-num* $= 100,000$. Give the results of its execution: How many values of n cause LOOPER to output FALSE? How many values lead to an infinite execution? What is the minimal value in each category? Enclose a print-out of your program; make sure that it is well-commented.

(c) Mathematics

Choose any two of the following five questions and write your thoughts about them. If you cannot get precise mathematical results, give your hypothesis and some intuition supporting it.

1. What mathematical properties of n determine whether LOOPER terminates or runs forever?
2. Similarly, what properties of n determine whether LOOPER returns TRUE or FALSE?

For the next three questions, assume that *big-num* is $+\infty$; i.e. LOOPER terminates only if n becomes 1.

3. Are there any values larger than 100,000 that make LOOPER run forever?
4. Is the number of such values finite or infinite?
5. Are there any initial values of n that result in an infinite growth of n,
   rather than its decrease to 1 or a fixed cycle of values?