Ice Cubes from Heaven

November 26, 2016

- *LIKELY RELATIONSHIP TO FINAL EXAM:* Memoization and dynamic programming are imporant for the exam, but the open "Collatz Conjecture" related to the hailstone problem is not something we plan to ask about.

The next step in the "hailstone" sequence is defined for integers $n \geq 1$ by:

$$h(n) = \begin{cases} n/2 & \text{if } n \text{ is even} \\ 3n+1 & \text{otherwise} \end{cases}$$

We consider the sequence to end when $n = 1$.

For example, here are the sequences beginning at 1–5, each labeled with the number of steps taken to reach 1.

n	sequence	# of steps
1	1	0
2	2, 1	1
3	3, 10, 5, 16, 8, 4, 2, 1	7
4	4, 2, 1	2
5	5, 16, 8, 4, 2, 1	5

In this problem, we'll explore methods to find the number of steps in the sequences starting at each initial number $1 \leq i \leq n$ given an input n. Note: it is unknown whether the sequence does indeed reach 1 for every starting point.

1. Identify and explain a feature of this problem that makes memoization promising for its solution. (Note: this is way more blank space than you need but too little to include the next problem on this page.)

2. Give a memoized pseudocode algorithm that takes a number n and computes the number of steps in the sequence required to reach 1 for every value of i from 1 up to (at least) n.

   You may assume that your pseudocode language allows "automatically resizing sparse arrays". That is, you can define an array without specifying how big it is and then reasonably efficiently store a value at any index 1 or more (or 0 or more if you prefer 0-based indexing) without explicit resizing.

3. The obvious subproblem ordering for a dynamic programming variant is $1, 2, 3, 4, \ldots$. Explain why this subproblem ordering is **not** correct here.