DTU Compute

Lyngby 14-11-2023
Michael R. Hansen

An exercise on sequences: Collatz conjecture

Consider a sequence `collatz` n generated from a positive natural number n as follows way:

$$a_0\, a_1\, a_2\, a_3\, \cdots\, a_i \cdots$$

where $a_0 = n$ and

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

for $i > 0$.

For example, the first 8 elements of `collatz` n, for $n = 1, 2, 3, 4$, are:

$n = 1:$	1	4	2	1	4	2	1	4
$n = 2:$	2	1	4	2	1	4	2	1
$n = 3:$	3	10	5	16	8	4	2	1
$n = 4:$	4	2	1	4	2	1	4	2

1. Declare a function `collatz: int -> seq<int>` that for a given $n > 0$ denotes the Collatz sequence starting with n.
2. Declare a sequence `collatzSequences: seq<seq<int>>` with the elements:

   `collatz`(1)   `collatz`(2)   `collatz`(3)   `collatz`(4) $\cdots$

Collatz' conjecture is: Every sequence `collatz` n, for $n > 0$, will always reach 1.

This is an unsolved mathematical problem. An abundance of sequences have been generated and no counter example has so far been found, see e.g. `https://en.wikipedia.org/wiki/Collatz_conjecture`.

The *stopping time* of `collatz` n is the index in the sequence at which 1 first appears.

3. Declare a sequence `stoppingTime: seq<int>`. The element t_i in this sequence is the stopping time of `collatz` $(i + 1)$, for $i \geq 0$. Hence, the first 4 elements of this sequence are 0,1, 7 and 2. Hint: You may use `Seq.findIndex`.

Let $t_i, i \geq 0$, denote the elements of `stoppingTime`. Consider the sequence `maxStoppingTimes`:

$$m_0\, m_1\, m_2\, m_3\, \cdots\, m_i \cdots$$

where m_0 is t_0, m_1 is `max` $m_0\, t_1$ and m_i is `max` $m_{i-1}\, t_i$, for $i > 0$. Hence, m_i is the maximal stopping time found for the sequences `collatz` k, for $k = 1, 2, ..., i+1$. The first four elements of the sequence are 0, 1, 7 and 7.

4. Make a declaration of the sequence `maxStoppingTimes`. You may consider making two declarations, where one is based on a recursive function and the other on the library function `Seq.scan`. Let sq be a sequence with elements $a_0\ a_1\ a_2\ a_3\ \cdots$, then `Seq.scan` $f\ x_0\ sq$ gives the sequence

   $$x_0\ x_1\ x_2\ x_3\ x_4\ \cdots$$

   where $x_1 = f\, x_0\, a_0,\ x_2 = f\, x_1\, a_1$ and $x_{i+1} = f\, x_i\, a_i$.