

Stream Sample Questions

Background: Collatz Conjecture

Given a number n , consider the 2 basic operations

$$f(n) = \begin{cases} n/2 & \text{if } n \bmod 2 = 0 \\ 3n + 1 & \text{otherwise} \end{cases}$$

Imagine we repeatedly apply f on some n , i.e. $f(f(f(\dots f(n),$

Collatz conjecture suggests that: This process will eventually reach 1, regardless of what value n was initially.

If the conjecture is false, then there exists a number n such that repeatedly applying f will enter a loop that excludes 1, or increase without a bound. However, till now no such sequence has been found.

Create a stream of numbers where it corresponds to the sequence of Collatz Numbers. In other words, index i (assuming 1 based indexing) of the stream would be the result of applying f on n repeatedly i times, $s[i] = f^{i-1}(n)$, where s refers to the stream.

Question 1 Hailstone Sequence

Create a stream of numbers where it corresponds to the sequence of after applying f multiple times, this is the Hailstone sequence. In other words, index i (assuming 1 based indexing) of the stream would be the result of applying f on n repeatedly i times, $s[i] = f^{i-1}(n)$, where s refers to the stream.

Write a static method `Stream<Integer> hailstoneNumber(int n, int m)` where n is the initial value that we will apply f on, and m is the size of the stream. Write in a single stream pipeline.

For example `hailstoneNumber(2, 2)` will evaluate to $[2] \rightarrow [1]$

and `hailstoneNumber(3, 9)` will evaluate to $[3] \rightarrow [10] \rightarrow [5] \rightarrow [16] \rightarrow [8] \rightarrow [4] \rightarrow [2] \rightarrow [1] \rightarrow [4]$

Question 2 Hailstone Sequence length

Write a static method `int hailstoneLength(int n)` where n is the initial value that we will apply f on, which returns the number of applications of f to reach 1. Use a single stream pipeline to calculate this.

For example `hailstoneLength(2)` will return 1

and `hailstoneLength(3)` will return 7.

Question 3 Stream of Hailstone lengths

Write a static method `Stream<Integer> streamHailstoneLength(int n)` where it is a stream of n elements. The i_{th} element (assuming 1 based indexing) is the collatz step for the i_{th} number. Use a single stream pipeline to calculate this.

For example `streamHailstoneLength(7)` will return $[0] \rightarrow [1] \rightarrow [7] \rightarrow [2] \rightarrow [5] \rightarrow [8] \rightarrow [16]$

Question 4 Powerset

Using the provided `ImmutableSet` class, write a static method which takes in a Set of numbers and returns a `Stream of ImmutableSet`. The stream should correspond to the power set of the input set. Order of the elements does not matter.

```
Stream<ImmutableSet<Integer>> powerSetStream(Set<Integer> set)
```

For example `powerSetStream(Set.of(1, 2, 3))` will return

$\{\} \rightarrow \{1\} \rightarrow \{2\} \rightarrow \{1, 2\} \rightarrow \{3\} \rightarrow \{1, 3\} \rightarrow \{2, 3\} \rightarrow \{1, 2, 3\}$