Cached Collatz P33673_en

For any positive integer number n, let $C(n)$ be $n/2$ if n is even, and $3n + 1$ if n is odd. The Collatz Conjecture states that the sequence $n, C(n), C(C(n)), \ldots$ leads to 1 for any n.

For example, let $n = 17$. We get $17 \to 52 \to 26 \to 13 \to 40 \to 20 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$, for a total of 12 steps. After that, the cycle 4, 2, 1 would keep repeating.

You want to make some computer analysis about the number of steps to get to 1 on several starting numbers, and leave the computer running overnight. To speed things up and avoid recalculation, you plan on storing all already computed numbers in a set. However, you realize that this would quickly exceed the memory constraints of your computer.

To avoid that, you discover LRU (Least Recently Used) caches, which can be helpful. Here's how they work:

- LRU caches have a fixed capacity K. They map at most K keys to their values.
- Whenever a key is checked in the cache, its value will be returned if the key is present. Otherwise, the value will need to be computed, and then stored in the cache.
- To store a new (key, value) when the cache is already full, the least recently used pair is evicted. We consider that we "use" a pair whenever we either store or check it.

With the help of this cache, your program will compute $S(n)$ for some initial values n in the following way:

- $S(n)$ will be computed recursively via the $C(n)$ transformation. However, if n is a cache hit (the key n exists in the cache), $S(n)$ is returned immediately. Otherwise, it's a cache miss, and $(n, S(n))$ is stored in the cache once $S(n)$ is computed.
- We will never store 1 as a key in the cache, and we can assume $S(1) = 0$.

For instance, this is the state of the cache for the first case of the sample input. It shows the (key, value) pairs, ordered from least to most recently used:

- After the 1st step: [] (The cache is empty.)
- After the 2nd step: [(2, 1)]
- After the 3nd step: [(2, 1)]
- After the 4th step: [(2, 1), (4, 2), (8, 3)] (Note that we can conclude $S(8) = 3$ only after we know $S(4) = 2$.)
- After the 5th step: [(16, 4), (32, 5), (64, 6)]
- After the 6th step: [(2, 1), (4, 2), (8, 3)]
- After the 7th step: [(4, 2), (8, 3), (2, 1)]
- After the 8th step: [(2, 1), (8, 3), (16, 4)]

Given K and some starting values n, please print $S(n)$ and the number of cache misses to compute $S(n)$ with the previous procedure.

Input

Input consists of several cases. Each case starts with K, followed by a non-empty sequence $n_1, n_2, \ldots$ of positive integers ending with a 0. Assume $1 \leq K \leq 10^4$, and that the Collatz sequence of each given n_i will never overflow with the usual 32-bit signed integers.

Output

For each n_i, print $S(n_i)$, and the number of cache misses found while computing it. Suppose that the cache is empty at the beginning of each case. Print a line with 10 dashes at the end of every case.

Sample input 1

```
3  1 2 2 8 64 8 2 16 0
50  15 7 85 18 92 0
```

Sample output 1

```
1: 0 0
2: 1 1
2: 1 0
8: 3 2
64: 6 3
8: 3 3
2: 1 0
16: 4 1
----------
15: 17 17
7: 16 8
85: 9 5
18: 20 4
92: 17 1
----------
```

Problem information

Author: Víctor Chabrera

Generation: 2026-01-25T10:18:50.182Z