

Problem

The following iterative sequence is defined for the set of positive integers:

$$n \rightarrow \frac{n}{2} \text{ (} n \text{ is even)}$$

$$n \rightarrow 3 \cdot n + 1 \text{ (} n \text{ is odd)}$$

Using the rule above and starting with 13, we generate the following sequence:

$$13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$$

It can be seen that this sequence (starting at 13 and finishing at 1) contains 10 terms. Although it has not been proved yet (Collatz Problem), it is thought that all starting numbers finish at 1.

Which starting number, under one million, produces the longest chain?

NOTE: Once the chain starts the terms are allowed to go above one million

Solution

Solution: 837799

Time: 1.55472

Algorithm

If we compute chain for every number from scratch, our program will be recomputing many chains. Moreover, we are only interested in length of chain for every number below 1,000,000. Therefore, it is more time efficient to store length of chain for every number.

The number and its number of terms in sequence can be placed into structure. Hash table is an ideal data structure for storing the structures. Since there can be more structures in every bucket of the hash table (Bucket 3 can store structures for number 3 and 1,000,003.), we will use hash table of linked lists.

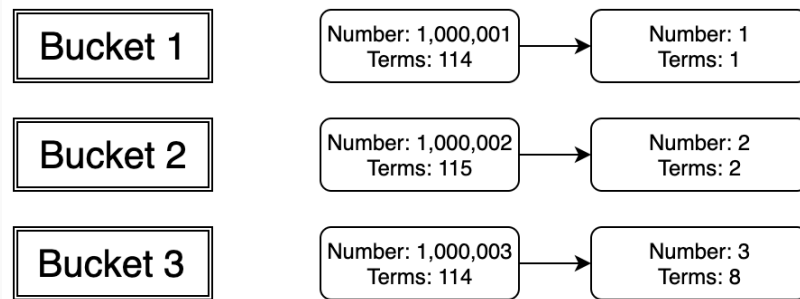


Figure 1: Hash Table of Linked Lists

Algorithm 1 Find Number With The Longest Chain

```
1: function MAIN
2:   create hash table hashTable of linked lists of structures  $\{number, terms\}$  with
      1,000,000 buckets
3:   insert base case  $\{1, 1\}$  to hashTable
4:   create vector numbersNotInTable of integers
5:   for  $n = 2$  to 1,000,000 do
6:      $number = n$ 
7:     empty numbersNotInTable
8:     while  $number$  is not in hashTable do
9:       push  $number$  to the back of numbersNotInTable
10:       $number = \text{NEXTCOLLATZNUMBER}(number)$ 
11:      if numbersNotInTable is not empty then
12:         $\text{INSERTNUMBERS TOTABLE}(hashTable, numbersNotInTable, number)$ 
13:      find number with the most terms in hashTable

14: function NEXTCOLLATZNUMBER( $number$ )
15:   if  $number$  is even then
16:     return  $\frac{number}{2}$ 
17:   else
18:     return  $3 \cdot number + 1$ 

19: function INSERTNUMBERS TOTABLE( $hashTable, numbersNotInTable, number$ )
20:   search hash table for number of terms of  $number$ 
21:   set  $terms$  to number of terms of  $number$ 
22:   set  $size$  to size of numbersNotInTable
23:   for  $i = 0$  to  $size$  do
24:     insert number at index  $size - 1 - i$  in numbersNotInTable and number of
      terms  $terms + i + 1$  to hashTable
```
