Oware

(with integers)

On Tue, Sep 11, 2012 at 7:20 AM, Eric Angelini wrote:

> Hello SeqFans,
> The sowing technique performed below could recall the Oware game.
> (...)
> In French:
>
http://www.cetteadressecomportecinquantesignes.com/Awale.htm
>
> Maximilian Hasler is already working on this -- I hope this will
> be of interest for the rest of the list.
> Best,
> É.

Let's start with the integer 541, for example - which will be seen as three bowls containing respectively 5, 4 and 1 seed (there are as many bowls as figures in the original integer). At the start of a "game" one takes all the seeds of the leftmost bowl, which are then sowed one by one to the right (the last bowl on the right is followed by the first bowl on the left). The bowl where the sowing ends is also the one by which the next sowing begins. And so on. Here is how the 541 game starts (and ends); the yellow color marks the starting bowl and the one where the sowing ends:

5 4 1
1 6 3
2 7 1
3 7 0
1 8 1
0 9 1
3 3 4
4 1 5
4 0 6
6 2 2
7 3 0
8 1 1
8 0 2
9 1 0
9 0 1
10 0 0
3 4 3
4 1 5

6 3 1
7 1 2
7 0 3
8 1 1
9 1 0
3 4 3
1 5 4
0 6 4
2 2 6
3 0 7
1 1 8
0 2 8
1 0 9
0 1 9
0 0 10
4 3 3
1 5 4
3 1 6
1 2 7
0 3 7
1 1 8
1 0 9
4 3 3
5 4 1
6 4 0
2 6 2
0 7 3
1 8 1
2 8 0
0 9 1
1 9 0
0 10 0
3 3 4
5 4 1 <-- identical state of the bowls as at the start of the “game” – color included

We will then say that 541 loops in 51 steps.

How do the other naturals behave (from a “looping” point of view)?

“1” loops in a single step, of course. As do the integers from 2 to 9. Let’s see what happens with 10, 11, 12, 13... :

10
01
10 <-- loops in 2 steps

11
02
11

```
20
11  <-- loops in 4 steps

          12
          03
          21
          12  <-- loops in 3 steps

                    13
                    04
                    22
                    31
                    40
                    22
                    13  <-- loops in 6 steps
```

...

Is there a way to find the number of steps required by "n" to loop, only by looking at "n"?

The sequence **S** of "looping values" starts thus like this:

$\mathbf{S}_{looping(n)}$ = 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 4, 3, 6, ...

_____

September 12th upgrade

[**David Scambler**]:

> Hi Eric, nice idea!
>
> For 2-digit numbers maximum loops is 36, for number 99.
> There are 13 distinct loop lengths.
> For 3-digit numbers maximum loops is 657, for number 999.
> There are 60 distinct loop lengths.
> For 4-digit numbers maximum loops is 8700, for 64
> different numbers, lowest 2999, highest 9992. There are
> 223 distinct loop lengths.
>
> It is not clear to me that there will always be a
> solution. Maybe a loop could be formed before the
> original number is reached again.
>
> e.g.
>
> abc
> ...
> ...
> def
> ...

...
...
def <- loop perhaps, depending on which "digit" is "active"

I have not found such an example - maybe it can be proved that there is always a solution.

[**Maximilian Hasler**] :

Dear Eric & SeqFans,

(...) unfortunately I don't have the time, not even to think whether (or why) there cannot be some smaller "loop" in the orbit of some numbers such that the initial position would never be reproduced. But since you asked me, I looked at it and found this an interesting idea. I quickly wrote a PARI 1-liner (so cryptic that it segfaults on PARI versions prior to 2.4.4 due to a Vecsmall()++ bug) which calculates the "looping values":

```
b(n)={my(o=n=Vecsmall(Str(n)),c,p=Mod(0,#n));until(!p &
o==n,c++;for(i=1,n[lift(p)+1]-
n[lift(p)+1]=48,n[lift(p++)+1]++));c}
```

It gives:

Slooping(n) = [1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 4, 3, 6, 10, 12, 4, 8, 18, 6, 4, 3, 6, 10, 12, 4, 8, 18, 6, 11, 3, 2, 10, 12, 4, 8, 18, 3, 11, 20, 6, 10, 12, 4, 8, 18, 6, 11, 20, 18, 10, 12, 2, 8, 18, 6, 11, 4, 18, 28, 12, 4, 8, 18, 3, 11, 20, 6, 28, 5, 4, 8, 18, 2, 11, 20, 18, 28, 5, 10, 8, 18, 6, 11, 20, 18, 28, 5, 10, 12, 18, ...]

which seems to coincide with the **S** on your page (...)

[**David Scambler**]:

Maximilian wrote:

> "whether (or why) there cannot be some smaller "loop" in the orbit of some numbers such that the initial position would never be reproduced"

Indeed. I have unsuccessfully searched for short orbits in integers up to 1 million.
I am not sure when to stop searching and conjecture instead that there are none.

> The longest loop so far is 223200 for the integer 98999.
> Perhaps someone can check this assertion.

[**Lars Blomberg**]:

> For numbers below 100,000: 98999 has the longest loop = 223200
> For numbers below 1,000,000: 799989 has the longest loop = 1534716
> /LBg

Many thanks for the great job, folks. This is now in the OEIS, here.

_____

September 25th upgrade

[**Lars Blomberg**]:

> Hi Eric,
> Here are summaries of numbers containing 2 to 7 digits.
> David Scambler has already supplied 2-4 digits.
> It seems to me that the interest in your problems peaks fast and recedes quickly. This is a pity since it takes some time to investigate, not to mention computer time (7 digits took 25 hrs).
> (...)
>
> 2-digit: maximum loop 36 for **99**
> -- most common loop length 18 occurs 13 times
> -- 13 distinct loop lengths.
>
> 3-digit: maximum loop 657 for **999**
> -- most common loop length 48 occurs 64 times
> -- 60 distinct loop lengths.
>
> 4-digit: maximum loop 8700 for 64 numbers: **2999, 3998, 4799, 4889, 4979, 4988, 4997, 5789, 5969, 6599, 6698, 6779, 6788, 6797, 6869, 6968, 6995, 7499, 7589, 7688, 7697, 7769, 7787, 7949, 7958, 7967, 7985, 8498, 8588, 8597, 8768, 8777, 8786, 8795, 8867, 8885, 8948, 8966, 8975, 8993, 9299, 9389, 9479, 9488, 9497, 9569, 9587, 9668, 9677, 9686, 9695, 9749, 9758, 9794, 9839, 9857, 9884, 9929, 9938, 9947, 9956, 9974, 9983, 9992**
> -- most common loop length 164 occurs 261 times
> -- 223 distinct loop lengths.
>
> 5-digit: maximum loop 223200 for **98999**
> -- most common loop length 17395 occurs 1231 times

-- 789 distinct loop lengths.

6-digit: maximum loop 1534716 for 5 numbers: **799989, 879999, 899988, 899997, 969999**
-- most common loop length 88644 occurs 5641 times
-- 3171 distinct loop lengths.

7-digit: max loop 39079320 for 8 numbers: **7979989, 8889988, 9797998, 9879799, 9896899, 9899599, 9968998, 9995998, 9998977, 9998986**
-- most common loop length 925799 occurs 35056 times
-- 10472 distinct loop lengths

Many thanks again, **Lars**!
Best,
É.