# IMO 1993/6 Evan Chen

TWITCH SOLVES ISL

Episode 45

### Problem

Let n > 1 be an integer. In a circular arrangement of n lamps  $L_0, \ldots, L_{n-1}$  each of which can either be ON or OFF and with indices taken modulo n, we start with the situation where all lamps are ON, and then carry out a sequence of steps. On the jth step (for  $j = 0, 1, \ldots$ ) if  $L_{j-1}$  is ON then  $L_j$  is toggled, whereas if  $L_{j-1}$  is OFF then nothing happens. Show that:

- (i) There is a positive integer M(n) such that after M(n) steps all lamps are ON again,
- (ii) If n has the form  $2^k$  then all the lamps are ON after  $n^2 1$  steps,
- (iii) If n has the form  $2^k + 1$  then all lamps are ON after  $n^2 n + 1$  steps.

## Video

https://youtu.be/OvHiDzshu7M

#### **External Link**

https://aops.com/community/p372299

#### Solution

For part (i), the sequence of states (and time indices modulo n) is finite, so it is certainly eventually periodic; however, because the operation is invertible, it is totally periodic.

Throughout the problem we always work modulo 2. By a *round* we mean a sequence of n consecutive operations. Note that a round has the following effect:

$$(a_1, \dots, a_{n-1}, a_0) \mapsto (a_2 + a_3 + \dots + a_0, a_1 + a_2, a_1 + a_2 + a_3, \dots, a_1 + a_2 + a_3 + \dots + a_0).$$

(We prefer to list  $L_1$  first, rather than  $L_0$ .)

For (ii), we give a description of the process for  $n = 2^{k+1}$  in terms of the process for  $n = 2^k$ . We write our states as binary strings of length n, with the *i*th bit corresponding to  $L_i$  (the last bit corresponds to  $L_n = L_0$ ). The description goes as follows:

- We split the string into  $L_1L_2 \ldots L_{n/2} \mid L_{n/2+1} \ldots L_n$ .
- There are two copies of the process for  $n = 2^k$  side by side until we reach the string  $100 \cdots \mid 1000$ .
- After the next round, we get  $11 \dots 11 \mid 00 \dots 00$ .
- Then, the left half plays another copy of the  $n = 2^k$  process, with the right half being all zero, until we reach  $1000 \mid 0000$
- Then n-1 turns later we get 111...111 as needed.

For concreteness, if n = 4 is given by

$$1111 \xrightarrow{4 \text{ moves}} 1010 \xrightarrow{4 \text{ moves}} 1100 \xrightarrow{4 \text{ moves}} 1000 \xrightarrow{3 \text{ moves}} 1111$$

in terms of the rounds, then n = 8 is given by

and seven moves later we return to all 1's. One can now check by a straightforward induction that the number of moves is  $n^2 - 1$  exactly.

For (iii), we give a description for  $n = 2^k + 1$  in terms of that for  $n = 2^k$ . All that needs to be observed is that the situation after the *i*th round of  $n = 2^k + 1$  is the same as the *i*th round of  $n = 2^k$ , except with last bit moved to the first, and one extra 0 appended, up until we reach the special state 0010000.... For example, for n = 9 we have

$$111111111 \xrightarrow{9 \text{ moves}} 001010101 \xrightarrow{9 \text{ moves}} 001100110$$

$$\xrightarrow{9 \text{ moves}} 001000100 \xrightarrow{9 \text{ moves}} 001111000$$

$$\xrightarrow{9 \text{ moves}} 001010000 \xrightarrow{9 \text{ moves}} 001100000 \xrightarrow{9 \text{ moves}} 001000000$$

and then after an additional n + 1 moves we return to all 1's. The total move count can be computed to be the desired  $n^2 - n + 1$ .