# IMO 1993/6 <br> 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 $j$ th 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:

$$
\begin{aligned}
\left(a_{1}, \ldots, a_{n-1}, a_{0}\right) \mapsto & \left(a_{2}+a_{3}+\cdots+a_{0},\right. \\
& \left.a_{1}+a_{2}, a_{1}+a_{2}+a_{3}, \ldots, a_{1}+a_{2}+a_{3}+\cdots+a_{0}\right) .
\end{aligned}
$$

(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 $L_{1} L_{2} \ldots L_{n / 2} \mid L_{n / 2+1} \ldots L_{n}$.
- 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 $11 \ldots 11 \mid 00 \ldots 00$.
- 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 $111 \ldots 111$ as needed.

For concreteness, if $n=4$ is given by

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

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

$$
\begin{aligned}
1111 \mid 1111 & \xrightarrow{8 \text { moves }} 1010|1010 \xrightarrow{8 \text { moves }} 1100| 1100 \\
& \xrightarrow{8 \text { moves }} 1000|1000 \xrightarrow{8 \text { moves }} 1111| 0000 \\
& \xrightarrow{8 \text { moves }} 1010|0000 \xrightarrow{8 \text { moves }} 1100| 0000 \xrightarrow{8 \text { moves }} 1000 \mid 0000 .
\end{aligned}
$$

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 $0010000 \ldots$. For example, for $n=9$ we have

$$
\begin{aligned}
111111111 & \xrightarrow{9 \text { moves }} 001010101 \xrightarrow{9 \text { moves }} 001100110 \\
& \xrightarrow{9 \text { moves }} 001000100 \\
& \xrightarrow{9 \text { moves }} 001111000 \\
\xrightarrow{9 \text { moves }} 001010000 & \xrightarrow{9 \text { moves }} 001100000 \xrightarrow{9 \text { moves }} 001000000
\end{aligned}
$$

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 $n^{2}-n+1$.

