Analysis of LAMPS.

This task has basically 3 solutions with different complexity.

1. The first solution is to *simply compute all the intermediate patterns*. In this case, M different states for each of the N lamps must be computed, yielding an O(NM) solution.

2. A standard approach to improve the solution (for smaller values of M) is to *hash the pattern with some function.* Then it can be easily checked (probably with O(1) complexity), whether the same pattern has ever occurred before.

If the same pattern occurs at different moments t_1 and t_2, then the patterns repeat with cycle of length $t_2 - t_1$. Knowing that, the pattern at the moment M is equal to the pattern at the moment $t_1 + (M - t_1)$ **mod** $(t_2 - t_1)$.

The most intuitive hash function to use is to simply view the representation of the pattern as a binary number. Since the number of different patterns of length N is 2^N, the number of hash values is exponential to N. This method has a running time complexity $O(N \times 2^N)$, which does not depend on M. But it also requires $O(2^N)$ of memory.

However, since for most numbers N, the maximum length of the cycle of patterns consisting of N lamps is only a small fraction of 2^N, hash functions with less hash values can be used. For example, one could use only the states of k<N fixed lamps, rather than the states of all lamps, when hashing the pattern. We believe that the most suitable values for k are 20..24, as the array of 2^k elements should fit into the memory. However, for small values of N (N<100), this seems to be enough. Although for most (if not all) numbers N, the running time complexity is asymptotically less, than $O(N \times 2^N)$, we do not know any better bound.

3. The third solution is based on the fact that *we do not need to calculate all the M-1 intermediate patterns*, to get the M–th one.

First of all, we need to prove the following lemma:

Lemma 1. For every prime p and positive integers k, l, which satisfy $1 < l < p^k$, the binomial coefficient $C(p^k, l)$ is divisible by p.

Proof. Skipped, but not difficult.

Choosing $p = 2$, $C(2^k, l)$ is thus an even number.

Let L (l, t) denote the state of lamp l at moment t.

Let XOR($a_1 \times b_1$, $a_2 \times b_2$, ...,$a_i \times b_i$) denote the XOR of binary digits

$$\underbrace{a_1, a_1, ..., a_1,}_{b_1 \text{ times}} \quad \underbrace{a_2, a_2, ..., a_2,}_{b_2 \text{ times}} \; ..., \quad \underbrace{a_i, a_i, ..., a_i}_{b_i \text{ times}}$$

As the function XOR is both commutative and associative, we can write:

$L(n, n) = \mathrm{XOR}(L(n–1, n–1)\times 1, L(n, n–1)\times 1) =$

$= \mathrm{XOR}(\mathrm{XOR}(L(n–2, n–2)\times 1, L(n–1, n–2)\times 1), \mathrm{XOR}(L(n–1, n–2)\times 1, L(n, n–2)\times 1)) =$

$= \mathrm{XOR}(L(n–2, n–2)\times 1, L(n–1, n–2)\times 2, L(n, n–2)\times 1) =$

$= \mathrm{XOR}(L(n–3, n–3)\times 1, L(n–2, n–3)\times 3, L(n–1, n–3)\times 3, L(n, n–3)\times 1) =$

$= \mathrm{XOR}(L(n–j, n–j)\times C(j, j), L(n–j+1, n–j)\times C(j, j–1), L(n–j+2, n–j)\times C(j, j–2), \ldots$
$L(n–1, n–j)\times C(j, 1), L(n, n–j)\times C(j, 0))$

Choosing $j = 2^k$, and bearing in mind that $C(2^k, l)$ is even and $\mathrm{XOR}(a, a) = 0$, we get

$L(n, n) = \mathrm{XOR}(L(n–2^k, n–2^k)\times C(2^k, 2^k), L(n–2^k+1, n–2^k)\times C(2^k, 2^k–1),$
$L(n–2^k+2, n–2^k)\times C(2^k, 2^k–2), \ldots L(n–1, n–2^k)\times C(2^k, 1),$
$L(n, n–2^k)\times C(2^k, 0)) =$
$\mathrm{XOR}(L(n -2^k, n–2^k)\times 1, L(n, n–2^k)\times 1),$

thus the state of the lamp is determined by the states of 2 lamps 2^k seconds before, and can be computed with a single XOR.

Hence we only need to compute *lg*(M) intermediate patterns to get the answer, therefore the time complexity is O(N×*lg*(M)).