light：

首先注意到连续段数等于开着的灯数减相邻两个都开着的灯对数。

令 k=sqrt(n)，对于出现次数>k 的颜色，我们维护与这种颜色相邻的开着的灯数。每次我们改变>k 的颜色时，直接用这个算答案。改变<=k 的颜色就暴力枚举每个位置。

时间复杂度 O(nsqrt(n))。

crossing：

我们设 x_i 表示第 i 次观察的时间(对周期取模)，那么如果有一个灯在两次观察中都是红灯，我们就可以得到一个形如 $x_i-x_j=k$ 的方程。将这看做是 i 指向 j 权值为 k 的有向边，那么一个环的长度显然是周期的倍数。不难证明最小环就是周期，可以用 floyd 求出，时间复杂度为 $O(m^3+nm^2)$

同理我们可以设 y_i 表示第 i 个灯由红变绿的时间，可以用一样的方法求出，时间复杂度为 $O(n^3+mn^2)$。结合两种方法可以做到 O(nmsqrt(nm))。

如果我们建图时将权值排序后只连接相邻权值之间的边，那么可以证明两种方法建出来的图中，任意一个简单环的长度都是周期，那么可以 dfs 求出。

时间复杂度 O(nmlogn)

escape：

注意到人的操作是可逆的，那么我们考虑将所有可能出现的状态整合成一些“标准状态”。我们不停合并能全部走到一起的人，那么会形成这样的情况：

(1) 有若干个段，每段之间是不连通的；
(2) 每段中有若干个小段，每小段中至少会有若干人，其余的人则可以在整个段中任意移动。

那么我们发现能到达某个房间的人数是固定的(与其它人在哪里无关)。

我们设 f[i][j]表示能到达第 i 个房间的人数恰好为 j 时，前 i 个房间最多有多少人。转移如下：

(1) j<a[i]时，转移到 f[i+1][j+b[i]]或 f[i+1][0~b[i]-1]；
(2) a[i]<=j<a[i]+b[i]时，转移到 f[i+1][j-a[i]]；
(3) a[i]+b[i]<=j 时，转移到 f[i+1][j]。

转移时出现新的人时加上对应的人数。

时间复杂度 O(n*max{m,ai,bi})