

数学与程序一道游戏题目的快速解法 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/253/2021\\_2022\\_\\_E6\\_95\\_B0\\_E5\\_AD\\_A6\\_E4\\_B8\\_8E\\_E7\\_c97\\_253818.htm](https://www.100test.com/kao_ti2020/253/2021_2022__E6_95_B0_E5_AD_A6_E4_B8_8E_E7_c97_253818.htm) 题目：有十个开关等间距排成一线，每个开关对应其上方的一盏灯(十盏灯也排成一线)。每按动一下开关，可以使对应的灯改变状态(原来亮着的将熄灭，原来熄灭的将被点亮)。但是，由于开关之间的距离很小，每次按动开关时，相邻的一个开关也将被按动。例如：按动第5个开关，则实际上第4、5、6个开关都被按动。而按动靠边的第1个开关时，第1、2个开关都被按动。并且，无法只按动最靠边的一个开关。现在给出十盏灯的初始的状态和目标状态，要求计算：从初始状态改变到目标状态所需要的最少操作次数。函数接口：int MinChange(const int Start[],const int End[])。其中：Start表示了初始状态，End表示了目标状态。表示状态的数组(Start和End)中，若某元素为0表示对应的灯亮着，否则表示对应的灯没有亮。调用函数时保证Start和End数组长度均为10，并保证有解。看了很多人的解法都是用循环遍历来判断是否达到最后要求，但是如果和线性代数结合的话，就有一种很快速的解法。约定：以下所用的‘^’号都是‘异或’的运算。先简化一下，假设有四个灯，初始状态s0~s3，目标状态是e0~e3，转换一次状态就是和1进行异或运算一次，所以状态转移矩阵为： $(s_0, s_1, s_2, s_3) \begin{matrix} k_0^* & (1, 1, 0, 0) & k_1^* & (1, 1, 1, 0) & k_2^* & (0, 1, 1, 1) & k_3^* & (0, 0, 1, 1) \end{matrix} = (e_0, e_1, e_2, e_3)$ 。其中k(n)表示第n个开关所翻动的次数。并且，注意异或运算中 $a \oplus b = b \oplus a$ ，所以，某个开关翻动偶数次的效果相当于没有翻动，翻动奇数次的效果相当于翻动一次。又由于异或运算满足

交换律，所以翻动的顺序没有影响。综上每个开关翻动的次数只有1次或0次就足够了。设 $m(n)=s(n) e(n)$ ，注意异或运算中的-也就是，所以解线性方程组： $k_0 k_1 = m_1$ .  $k_0 k_1 k_2 = m_2$ .  $k_1 k_2 k_3 = m_3$ .  $k_2 k_3 = m_4$ . 假设解存在，就可以算出通解 $(k_0, k_1, k_2, k_3)$ ，再统计出通解中1的个数，就是所需要翻动的次数了。并且还可以知道哪些开关需要拨动，比如算出解是 $(1, 0, 1, 0)$ 就是第0和2个开关需要拨动一次。因此针对本题目的10个灯泡，本人已算出这10元线性方程组的通解： $k_0 = m_0 m_2 m_3 m_5 m_6 m_8 m_9$ .  $k_1 = m_2 m_3 m_5 m_6 m_8 m_9$ .  $k_2 = m_0 m_1$ .  $k_3 = m_3 m_0 m_1 m_5 m_6 m_8 m_9$ .  $k_4 = m_5 m_6 m_8 m_9$ .  $k_5 = m_4 m_3 m_0 m_1$ .  $k_6 = m_6 m_4 m_3 m_0 m_1 m_8 m_9$ .  $k_7 = m_8 m_9$ .  $k_8 = m_7 m_6 m_4 m_3 m_0 m_1$ .  $k_9 = m_9 m_7 m_6 m_4 m_3 m_0 m_1$ . 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)