原文：http://www.cnblogs.com/Java-tp/p/3873557.html

这一题大意如下：

      一个冰箱上有4\*4共16个开关，改变任意一个开关的状态（即开变成关，关变成开）时，此开关的同一行、同一列所有的开关都会自动改变状态。要想打开冰箱，要所有开关全部打开才行。

     输入：一个4×4的矩阵，+表示关闭，-表示打开；

　  输出：使冰箱打开所需要执行的最少操作次数，以及所操作的开关坐标。

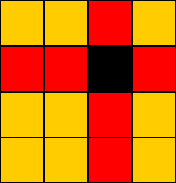
这一题很多人使用BFS的方法，其实还有一种更为精妙的方法，值得大家思考。

这种方法网上很多人都提到了，但是说的都不太清晰，我想了很久也没想明白，后来我用程序追踪一下运行的过程，才豁然开朗。

首先要明白最基本的原理：对一个开关进行操作n次，如果n为偶数，那么这个开关以及同行、同列的开关状态都不发生改变，等价于没有操作；如果n为奇数，那么这个开关以及同行同列的开关状态全都发生改变，等价于只操作了一次。

要想使所有开关状态全部打开（全部是-），就要把所有+变成-，所有-不改变。我们要做的就是找到一种“公式”，策略，使得不改变已经打开的开关状态的情况下，把关闭的开关打开。这点很类似于魔方（PS：玩过魔方的都知道，魔方所谓的公式，其实就是在不改变已经拼好的部分的情况下，把其他部分一点一点添加到已拼好的部分）。

我们找到的策略就是：把开关本身以及其同一行同一列的开关（总共7个）都进行一次操作，结果是，开关本身状态改变了7次，开关同一行、同一列的开关状态改变了4次，其他开关状态改变了2次。如下图所示。



假如开关坐标为第二行第三列的（2，3），那么按照上述策略（把开关本身以及其同一行同一列的开关都进行一次操作），结果分析如下：

对于黄色部分的开关，只有与此黄色开关同一行和同一列的两个红色开关操作时，此黄色开关的状态才会发生改变，因此所有黄色部分状态改变次数为2，相当于0次

对于红色部分的开关，只有与此红色开关同一列或同一列的开关操作时，此红色开关状态才会发生改变，一行或者一列有4个开关，因此红色部分开关状态改变次数为4，相当于0次

对于最原始的那个黑色开关，所有红色开关操作时，它的状态改变一次，然后黑色开关自己操作一次，因此黑色开关状态改变7次，相当于改变1次。

总结上述分析可以得出结论，把开关本身以及其同一行同一列的开关都进行一次操作，最终结果是只有开关本身状态发生变化，其他所有开关状态都不变。

策略找到之后，那我们就想，如果对于所有关闭着的开关都进行一次上述策略，那么肯定是能把冰箱打开的，下面我们要做的就是把一些无用的，重复的操作去掉即可。

用一个4\*4的数组记录每个开关操作的次数，初始化为0，开关操作一次，记录就+1，以样例（http://poj.org/problem?id=2965）为例：

-+--  
----  
----  
-+--

1 0 0 0  
0 0 0 0  
0 0 0 0  
0 0 0 0

1 1 0 0  
0 0 0 0  
0 0 0 0  
0 0 0 0

1 1 1 0  
0 0 0 0  
0 0 0 0  
0 0 0 0

1 1 1 1  
0 0 0 0  
0 0 0 0  
0 0 0 0

1 1 1 1  
0 1 0 0  
0 0 0 0  
0 0 0 0

1 1 1 1  
0 1 0 0  
0 1 0 0  
0 0 0 0

1 1 1 1  
0 1 0 0  
0 1 0 0  
0 1 0 0

1 1 1 1  
0 1 0 0  
0 1 0 0  
1 1 0 0

1 1 1 1  
0 1 0 0  
0 1 0 0  
1 2 0 0

1 1 1 1  
0 1 0 0  
0 1 0 0  
1 2 1 0

1 1 1 1  
0 1 0 0  
0 1 0 0  
1 2 1 1

1 2 1 1  
0 1 0 0  
0 1 0 0  
1 2 1 1

1 2 1 1  
0 2 0 0  
0 1 0 0  
1 2 1 1

1 2 1 1  
0 2 0 0  
0 2 0 0  
1 2 1 1

对于样例中的每一个+开关，进行一次策略，记录数组所记录的每一个开关操作的次数变化如上所示。那么在最终得到的数组中可以看出，有些开关操作了偶数次，有些操作了奇数次。操作了偶数次的开关就是上面所说的无用的，重复的操作，直接去掉，留下奇数次的就最终的答案。

代码写的不好，仅供参考