# 问题1：奶牛拍照（leftout）

## 题意

给定一个 `01` 矩阵，每次可以对一行或者一列翻转，多次操作后可以实现只有一个位置与其他所有位置不同，求这个位置。

## 思路

对于任意局面，将每行都再翻转一次，就可以变成对称的局面，所以首先可以假定最终能够让所有位置均为 `0` ，只有一个位置为 `1` 。

考虑枚举这个不同的位置，再进行检查，只不过这样做时间复杂度较高，但是不难发现，不论枚举的位置在哪，检查的步骤总是相似的。

可以先假设位置不在第一行和第一列，那么为了将第一行和第一列都变成 `0` ，应该根据第一列的每个元素决定哪些行要操作，再根据第一行的每个元素决定哪些列要操作，最后再找到矩阵中 `1` 的位置。

如果位置在第一行或者第一列，那么按照上述规则操作后会导致两种可能的局面：

1. 如果位置在第一列的非第一行，那么会导致这一行的非第一列均变成 `1` 。
2. 如果位置在第一行的非第一列，那么会导致这一列的非第一行均变成 `1` 。
3. 如果位置在第一行第一列，那么会导致非第一行且非第一列的所有元素都变成 `1` 。

四种局面之间是没有交集的，通过判断是哪种局面就能找到 `1` 的位置，可以统计出每行、每列和全部 `1` 的数量进行判断。

## 复杂度

### 时间

模拟操作并统计 $O(N^2)$ 。

判断位置 $O(1)$ 。

总时间复杂度为 $O(N^2)$ 。

### 空间

记录矩阵 $O(N^2)$ 。

---

# 问题2：假期安排（vacation）

## 题意

给定 $N$ 个点的有向图，其中有 $K$ 个关键点。进行 $Q$ 次询问，每次求 $A$ 到 $B$ 至少经过一个关键点的最短路径。

# 思路

因为点数很少，所以可以直接用 Floyd 算法求出任意两点之间的最短路长度。

对于每个询问，可以枚举要经过的关键点，利用已求出的最短路长度计算经过关键点的最短路径长度，并找到其中最短的即可。

# 复杂度

## 时间

Floyd 算法 $O(N^3)$ 。

询问数量 $O(Q)$ ，枚举关键点 $O(N)$ ，总共 $O(NQ)$ 。

总时间复杂度为 $O(N^3+NQ)$ 。

## 空间

记录所有点对最短路 $O(N^2)$ 。

---

# 问题3：爬山（climb）

# 题意

给定 $N$ 只奶牛上山和下山需要的时间，可以按照任意顺序上山和下山，但同时只有一只奶牛在上山，只有一只奶牛在下山，求所有奶牛上下山花费的最少时间。

# 思路

首先，虽然奶牛可以按照任意的顺序上下山，但是不难发现，下山的顺序是没必要调整的，如果同时有多只奶牛在山顶，她们按照任意顺序下山，都不影响答案，所以我们只需要考虑奶牛行动的顺序即可。

令当前最后一只奶牛上山的时间为 $x$ ，下山时间为 $y$ ，加上下一只上下山时间分别为 $u$ 和 $d$ 的奶牛，更新时间的方式为：

1. $x$ 加上 $u$ 。
2. $y$ 和 $x$ 取较大值。
3. $y$ 加上 $d$ 。

初始时 $x$ 和 $y$ 均为 $0$ ，最后的答案就是 $y$ 。

通过更新的方式可知，我们应该尽量避免 $y$ 被 $x$ 更新，由此可以贪心的安排顺序：

1. $u<d$ 的奶牛应该比 $u\geq d$ 的奶牛先走。
2. 在所有 $u<d$ 的奶牛中， $u$ 小的奶牛应该先走，此时第一只奶牛上山的时间也最短。
3. 在所有 $u\geq d$ 的奶牛中， $d$ 大的奶牛应该先走，此时最后一只奶牛下山的时间也最短。

# 复杂度

## 时间

排序 $O(N\ log\ N)$ 。

计算答案 $O(N)$ 。

总时间复杂度为 $O(N\ log\ N)$ 。

## 空间

记录时间 $O(N)$ 。