A 无限农场

大模拟。

因为坐标极限范围很大，但是牛的数量很小，我们需要一些有技巧的模拟。

我们可以记录所有奶牛的的位置和方向，以及奶牛是否停止，通过一些计算我们可以确定下一次出现奶牛被逼停的时刻，然后我们就标记被逼停的奶牛的状态，然后让所有奶牛都移动到当前时间点的状态，然后重复上述过程。每一步都需要$O(n^2)$去枚举寻找下一个时间点，每找到一个时间点有一只奶牛被逼停，所以整个时间复杂度为$O(n^3)$，核心问题就是直接跳到逼停的时间点，而不是1个单位时间的跳。

B 传染

有一棵树，根上有一只病牛，每一天可以选择将一个节点上的病牛复制或者让一头病牛转移, 求使所有节点都有病牛的最小天数

直接从最开始入手是比较难的，但是我们可以从树的最底下入手，我们可以发现我们的最优策略应该就是自己一次性传染够再一个个发到下一层（也就是自己的所有儿子）， 这样是最好的，因为如果先转移再让下面的复制会浪费时间。所以答案就应该是 $\left(\sum_{i=1}^{n}\lceil\log son_i\rceil\right)+n-1$, 建图遍历即可。

C 传染病

思路一：

先对数据进行离散化和排序, 然后枚举矩形的最左点 i 和最右点 j 。发现矩形的上边最小为 $u=\max\left(y_i, y_j\right)$, 下边最大为 $d=\min\left(y_i, y_j\right)$ 。子集数量则为满足 $x_i<x<x_j, y>u$ 的 (x, y) 点数量, 与 $x_i<x<x_j, y<d$ 的 (x, y) 数量相乘。 如何统计这些点的个数呢? 我们可以二维前缀和预处理, 时间复杂度为 $O(n^2)$, 可以通过本题。

思路二：

显然，每行、每列都至多只有一个点。对行坐标排序，滤掉没有牛的行（这实际上相当于离散化）。设 $a[i]$ 代表第 i 行的那头牛所在的列坐标。 考虑下侧木板在第 i 行, 上侧木板在第 j 行的情况。为了避免相同情况重复计算（即围栏形状不同，但圈 住的牛相同）, 第 i 行的牛和第 j 行的牛必须被圈住。

- 如果第 j 行的牛在第 i 行的牛左边（即 $a[i]>a[j]$）:
- 任何一头在 第 i 行与第 j 行之间，在第 j 行的牛左边 的牛, 即满足 $i<k<j, a[k]<a[j]$ 的 k ，都可以作为左侧围栏所在位置的选项（如果不满足此条件, 则第 j 行的牛无法被圈住) ; 当然, 第 j 行的牛本身 $(a[j])$ 也可以作为左侧围栏所在位置的选项。
- 任何一头在 第 i 行与第 j 行之间，在第 i 行的牛右边 的牛, 即满足 $i<k<j, a[k]>a[i]$ 的 k ，都可以作为右侧围栏所在位置的选项（如果不满足此条件, 则第 i 行的牛无法被圈住) ; 当然, 第 i 行的牛本身 $(a[i])$ 也可以作为右侧围栏所在位置的选项。
- 第 j 行的牛在第 i 行的牛右边（即 $a[i]<a[j]$) 的情况同理。 所以, 下侧木板在第 i 行, 上侧木板在第 j 行的所有可能, 即为 左侧围栏选项总数 $\times$ 右侧围栏选项总 数。 我们只需在枚举 i, j 的过程中, 维护:

对于第 i 行的牛, 在第 i 行与第 j 行之间, 有多少头牛在它的左边/右边;

对于第 j 行的牛, 有第 i 行与第 j 行之间, 有多少头牛在它的左边/右边;

顺带累加答案, 即可轻松完成此题。

时间复杂度为 $O\left(N^2\right)$ 。

阻碍

第一题的进阶版，求的东西不一样。

思路一：

将奶牛根据行进方向的不同分成两类

因为所有的奶牛的 x, y 坐标都不相同。

意味着向北走的奶牛只会被向东走的奶牛阻挡，向东走的奶牛只会被向北走的奶牛阻挡。

那么如何判断每头奶牛阻挡的个数呢? 观察数据范围发现：只可能从奶牛的行进方向入手（因为 x, y 的取值过大）。所以我们可以将每一头奶牛的 路径"画出来"。但是这样还是不能避免涉及过多坐标的问题。所以还需要精简。 考虑奶牛什么时候会停下：当且仅当方向不同且已行进路程的两头奶牛相遇时，才会有一头奶牛停下。所以我们可以

维护交点

所以现在重点就是如何记录交点以及交点需要存储哪些信息。首先我们要考虑哪些奶牛可以产生交点。 产生交点的条件:

1. 两头奶牛的方向不同
2. 假设向东走的奶牛为 c_e,向北走的奶牛为 c_n, 则奶牛要相遇的条件为 $c_e . x < c_n . x$ 且 $c_e \cdot y > c_n . y$

这个结论很好推, 因为对于 c_n, 它的 x 坐标是不变的, 所以如果一开始 c_e 的 x 坐标就比 c_n 大，那么很显然不能交上, y 坐标同理 现在我们已经有了一张偌大的图，图上有若干个交点，那么我们应该维护哪些信息呢? 假设现在有一个交点 P, 由两头牛 N, E 相交得到, 那么我们现在要判断哪一头牛会被挡住, **只需判断两头牛起始点距交点的距离即可**（很显然距离大的会被挡住)。 所以交点维护的信息已经一目了然： 交点坐标, 由哪两条线相交而来（存储线的编号）。 但现在又有一个问题了：有的牛会被挡住, 它实际上只走了一个线段, 但是我们存储的仍然是一条射线。 这个问题其实很好解决, 我们可以设置一个删除数组，维护每一头牛有没有被挡住, 那么在交点判断时, 如果有牛被挡住了，很显然这个交点是不存在的。那么对于没有被挡住的牛，他挡住的牛的个数应该是

他本身挡住的牛的个数 $+$ 他挡住的另一头牛挡住的牛的个数 $+1$ (另一头牛本身)

最后一个问题：判断交点顺序

相交距离越小的点越先被判断。那么哪些点的距离比较小呢? 很显然是左下角的点（因为牛是往右或者上走的）。所以我们要对交点进行排序，以 x 坐标和 y 坐标的任意一个作为第一关键字排序,另一个作为第二关键字排 序（从小到大），然后判断交点即可。

思路二：

阻碍关系建图跑拓扑排序，将 a 阻碍 b 的关系看成一条有向边 (a, b), 那么这些关系会转化为若干棵树, 即森林。建反图并拓扑排序 即可。 有关森林的说明：由于一头奶牛至多会被阻碍一次, 且阻碍奶牛存在先后关系, 不可能出现 a_1 阻碍 a_2, a_2 阻碍 $a_3, \cdots$, 且 a_k 阻碍 a_1 的情况。