Vode

我们知道一类问题：两人分别取数(由上一人取的数决定区间)，最终被迫取到m的人败。显然，这类问题是此题的简化版(因为不存在连续取数的情况)。对于这类题目，我们数学上的解决方法一般是从结果往前推，推出每一个**必胜点**(若得到这个点，则可以得到下一个，以此类推，最终得到$m-1$而使敌方必败)。

有了这类问题的基本解决思路，我们来解决本题中同一种族可以连续取数的情况。我们当前考虑第i头羊取j这个数的情况。
若$[j+1,j+k]$中有必胜点且下一头羊是**同种族的羊**，则j点也是必胜点。这是因为$[j+1,j+k]$是下一头羊的势力范围，它拿到了必胜点，那么j根据我们上文的分析，也是必胜点。我们只需要查找$[j+1,j+k]$中有无必胜点即可。普通的暴力无法胜任，我们可以通过后缀和(当然也可以前缀，只不过要将答案的更新从从后往前变为从前往后)发现区间内有无必胜点(无必胜点则区间和为0)。这样我们通过DP形式(伪DP)推出答案

设$f[i][j]$为轮到第i个人，还剩j颗石子的胜负态，暴力转移$O(n^3)$，考虑前缀和优化。

设l为取最多石子能到达的数量，我们可以求得$[l,j-1]$的胜负态之和，现在讨论先后手的队伍：

- 如果先后手同属一队，那么和必须包含至少一个必胜态当前状态必胜，否则必败。
- 如果属于不同对，那么和只要包含至少一个必败态当前状态必胜，否则必败。

在dp时顺便维护一下前缀和即可，时间复杂度$O(n^2)$

Pictionary

首先我们可以考虑如何把这棵树建出来。
每一个时刻，**显然**可以直接考虑第$m-i+1$座城市和所有$m-i+1$的倍数连边。
这两张图是等价的。（在只需要判断联通性的情况下
那么就可以直接这样暴力枚举，然后用一个并查集记录两个点是否在同一个集合里。
暴力枚举的复杂度是$\frac{N}{1}+\frac{N}{2}+...+\frac{N}{N}$的，所以复杂度是$O(N\times log_N)$的

这棵树建出来了，我们可以给这棵树附上权值。
每条边的边权就是当前的时刻，从小到大加边。
然后直接查两个点之间的路径上的瓶颈就好了。
这个贪心**显然**是对的，笔者不证。

然后，这棵树**显然**是一棵无根数。
那么如何转化成一个有根数呢?
方法很简单，直接钦定1为根就好了。

然后这棵树建出来之后，接下来也没有修改操作。
我就很想不通为什么楼上楼下都要用树剖做。
这不是很**显然**一个lca的问题吗，真是令人谔谔。

每一个时刻，题目要求我们把 $\gcd(a,b)=m-i+1$ 的 a 与 b 之间连边，我们可以想到把 $m-i+1$ 与它的倍数连边。不难发现，在只考虑连通性的情况下两张图等价。

给第 i 天连的边赋权值 i，题目转换成查询两点 x,y，找到图上一条两点之间的路径使得路径上的最大值最小。

稍微一想，这不就是 P1967 [NOIP2013 提高组] 货车运输 吗？（很好奇为什么题解里没有人提出这一点）一个小区别就是这两个题目一个是最大值最小，一个是最小值最大。

原题做法是用 kruskal 先求出最大（这题里是最小）生成树，然后再树上倍增类似 LCA 地维护最值。在本题中同样适用也可以通过。注意到本题连边的过程已经排好序，所以省去了 kruskal 的排序过程。

Programiranje

这题就是让我们判断字符串里的两个字串所包含的每个字母的个数是否相同。

因为子串是连续的，所以我们不难想到前缀和，利用前缀和来快速求出一个子串所包含每个字母的个数。

设 $f[i][j]$为字母j在1到i的子串出现次数,那么$f[i][j]=f[i-1][j]+(j==c[i]);$

求出前缀和后，求a到b子串出现字符j的次数为 $f[b][j]-f[a][j-1]$

最后判断询问的两个子串所含每个字母的个数是否相同即可。

Portal

我们考虑本题中人物的行走方式。容易发现，如果人物在(x,y)，那么人物就只有以下两种移动方式：

1. 移动至(x,y)上下左右的格子，花费1。
2. 移动到(x,y)四个方向的墙壁旁的格子，由于必须移动到一面墙壁旁边才可以使用传送门，所以花费为(x,y)最近的墙壁+1。因为走进传送门也需要1的花费。

这显然是一道最短路的题目，~~显然出题人没法用菊花图来卡spfa~~，所以就可以考虑用$spfa$来解决。

对于移动方式一，我们只需要从任意一点向四周连边即可。

对于移动方式二，我们可以$O(nm)$先暴力求出每一个点四个方向的墙壁位置。以上方的墙壁为例：

- 如果$(x-1,y)$为墙壁，则能传送到的位置为(x,y)。
- 如果$(x-1,y)$不为墙壁，那么$(x-1,y)$能传送到的位置即为(x,y)能传送到的位置，直接用$(x-1,y)$的答案赋值即可。

然后要求出每一个点到最近的墙壁的距离。这个只需要从所有的墙壁开始bfs就可以$O(nm)$求出。

然后跑一遍最短路就可以了。

向周围四个格子建立传送带，**实质上可以转为一种操作**：

设从一个格子 (x,y) 向四周直着走，碰到墙的最短步数为 s，走到尽头可以走到 (x_0,y_0)、(x_1,y_1)、(x_2,y_2)、(x_3,y_3)，那么可以看做可以从 (x,y) 可以走到 (x_0,y_0)、(x_1,y_1)、(x_2,y_2)、(x_3,y_3) 耗费 s 的时间。

那么有了这一个，就变成了一个非常普通的最短路了。用 Dijkstra 实现即可。

那么怎么求出 (x_0,y_0) 这些点呢？事实上可以暴力从每一个点开始向四个方向循环，时间复杂度 $O(n^3)$（这里视为 n,m 同阶），这样子是可以过的。

怎么更高效呢？考虑到一个可走格子 (x,y) 左边的可走格子 $(x,y-1)$ 和 (x,y) 一直往左走走到的格子是一样的。所以可以用记忆化搜索来实现。