1001

在做这题之前，我们需要知道 **Anti_Nim**：即在经典 Nim 游戏中，把获胜的条件改成 **最后一个将石子取完的人失败**。我们有以下结论：

1. 当每堆石子的数量都为 1 时，异或和为 0 时先手获胜；
2. 当有至少一堆石子的个数大于 1 时，异或和不为 0 时先手获胜。

由于全 1 的情况比较特殊，我们先假设不存在有全 1 的堆。此时可以发现，当异或和不为 0 的时候，在 Nim 和 Anti_Nim 中都是必胜的。在本题中，当异或和不为 0 的时候，先手就可以决定下一个房间谁先手。因此，当第一个房间的异或和不为 0 的时候先手必胜，反之先手必败。

现在考虑加入全 1 的房间。我们发现如果是偶数个 1，对本题没有任何影响；如果是奇数个 1，如果不放在最开头的情况下也没有任何影响。而如果放在开头，也只是改变了 Alice 和 Bob 的先手顺序。

因此最后的结论为：

1. 当开头有奇数个 1 时，第一个非全 1 的房间的异或和不为 0 时 Alice 必胜；
2. 当开头有偶数个 1 时，第一个非全 1 的房间的异或和为 0 时 Alice 必胜；

计算方案数的时候，我们先不考虑偶数个全 1 的房间。然后我们枚举有多少个奇数个全 1 的房间放在最前面。

时间复杂度 $O(\sum k_i)$。

1002

因为操作 4，显然我们需要将 $1 \sim n$ 断成多个区间进行查询。

考虑线段树进行维护，每个区间维护三个信息：
1：已有的贡献；
2：剩余的金币数量；
3：被索要但未交付的金币数量；

合并左右两个区间时，左区间的 2 和右区间的 3 可以产生新的贡献。

操作 $1,2,3$ 会形成一个树形结构，例如第 5 天发生了修改事件，则可以看作节点 4 向节点 5 连一条边，边的信息就是修改操作，递归的时候进行修改，回溯的时候撤销此次修改，第 7 天回溯到第 5 天，就从节点 5 向节点 7 连一条边，不进行操作，最后在以节点 0 为根的树上进行递归和回溯即可。

时间复杂度：$\mathcal{O}(n + m + k \log n)$

1003

首先，奇数长度的字符串一定优秀，若所有长度为 $n(n \geq 2)$ 且 n 是偶数的字符串优秀，则所有长度为 $m(m \geq n)$ 且 m 是偶数的字符串优秀。所以我们只需要检查一个 len 即可，若 len 是奇数，则加 1 变成偶数。

以 17 位二进制表示一种方案，初始所有方案都合法，若某位为 1，则表示此方案不对该字母幻视。

对所有长度为 len 的串进行判断，若其不优秀，我们就会得到一个非法方案，比如对于 $abcd$，非法方案就是同时不选 a, b。

对于一个 17 位二进制，如果它有子集是不合法的，则它也必定是不合法的。因为如果不选 a, b 不合法，不选 a, b, c 当然也不合法。

通过 $SOSdp$ 处理出所有合法情况，取最小即可。

时间复杂度：$\mathcal{O}(T * 17 * 2^{17})$

1004

由于每个串中取出的字符串需要相等，我们不难想到 SA 可以实现这个过程。字符串中的长度可以用 height 来更新。

我们用单调栈来维护长度，合并时暴力更新 n 个串中的最大左端点 L_i。正向反向各做一遍即可。时间复杂度 $O(n \times \sum |s|)$。

1005

考虑建分层图，对于每个协会都单独建一个图，表示连续使用同一个协会的传送门，边的权值均为 0，对于每个遗迹再建一个汇点，每个协会子图对应点向该点连边，权值为 1，同时该点也向子图对应点连边，权值为 0，表示转换不同协会的传送门。由于协会的数量很多，对于每一个协会的子图，不需要把所有点都建出来，只用建有该协会传送门的点即可，可以用 map 处理。最后再建一个超级源点 s和超级汇点t ，s 连接 1 号遗迹所有子图的点，权值为 1，t 连接 n 号遗迹所有子图的点，权值为 0。 建完图后跑最短路, 求出 s 到 t 的最短距离即为答案。

1006

贪心。最终目的是让所有区域可达的最小代价。游客只能从高处走向低处，且区域高度两两不同，意味着所有局部极高点（峰点）是游客无法自然走到的。具体而言，若某个区域四周所有相邻区域的高度都低于该区域，则该点为峰点，只能通过传送器到达，故必须在该点建传送器。因此，第一步是遍历整个区域，找出所有峰点，这些点必须设置传送器。并且只会有这些传送器，因为新建传送器的价格大于任意一条传送器路径费用。

接下来考虑最小费用，建完所有必要的传送器后，为了让所有区域互通，只需要确保这些传送器之间连通即可，即求方格中 n 个点对应的最小生成树。由于传送器路径价格可得，区域高度涉及的权值参数远大于横轴数轴坐标权值，因为 x 和 y 的范围一定在 [1, 100] 有 919810 > (11400 + 514100) 。贪心可得所有点从低到高依次连接即最小生成树。

证明：使用Prim算法，初始化选定起始顶点为最低点，每次选取连接生成树与剩余顶点的最小边，必定是剩余顶点中高度最小的点和已有连接树中高度最大的点的边，以此类推，可得即将所有点从低到高连接为最小生成树。

1007

给出一种高维前缀和的做法。我们把 x, y 看成质因子相乘的形式。关于 LCM 有两个重要结论：

1. LCM(x, y) 的过程实际上是对 x, y 的每个质因子个数取 $\max$ 的过程。
2. 如果 LCM(x, y) $\neq$ x，那么如果仅对 y 进行 LCM 操作是无法变成 x 的。

而 $x \leq 10^7$ 的时候，x 的不同的质因子个数不超过 7 个。我们设 x 的质因子个数为 k 个。我们把每个节点的值 a_i 状压成二进制数 b_i。如果 a_i 的某个质因子个数与 x 相等，就把这一位置 1。而如果 a_i 不满足结论 2，我们就将 b_i 置 -1。

接下来我们用 dfs 来遍历这棵树。在枚举到节点 u 时，$f_{u,i}$ 表示在 u 的子树中，经过 u 路径的 LCM 的状压值为 i 的路径数量。两条路径合并时，$i_u | i_v = 2^k - 1$ 的路径的数量就是 LCM 为 x 的数量。这一步可以用高维前缀和来解决。

时间复杂度 $O(n \times k \times 2^k)$。瓶颈在高维前缀和的计算。

1008

由于答案需要对 2 取模，因此我们不考虑对答案为偶数的贡献。假设 b 中的数字分别是 $d_1, d_2, \ldots, d_m$，数量分别是 $c_1, c_2, \ldots, c_m$，且 $\sum c_i = k$。我们先考虑 b 的性质。b 的个数为

$\binom{k}{c_1} \cdot \binom{k-c_1}{c_2} \cdot \binom{k-c_1-c_2}{c_3} \cdot \ldots \cdot \binom{k-c_1-c_2-\ldots-c_{m-1}}{c_m}$

如果这个值是奇数，那么每一项都必须是奇数。根据 $Lucas$ 定理可知，$\binom{n}{m} \mod 2 = 1$ 的充要条件为 $n|m = n$。当 $\binom{k}{c_1} \mod 2 = 1$ 时，$k - c_1$ 只是在 k 中拿掉一些二进制位剩下的位。

注意到 $k \leq 10^{18}$，因此 k 的位数不超过 60 位。，而每一个 d_i 都至少会拿掉 1 位。当 $odd(a[L,R])$ 的大小超过 60 时，答案一定为 0。而 $odd(a[L,R])$ 的大小不超过 60 时，我们利用异或哈希在主席树上二分，依次找出 a 中的元素。可以维护方案数 $dp_{i,j,k}$ 表示在二进制下 1 的总和为 i 个，前 j 个数中选了 k 个位的方案数。这一步可以预处理。注意当 k 为奇数的时候需要特殊处理。

时间复杂度 $O(q \times log_2 n \times log_2 k)$。

1009

注意到是一个排列，而且是两端最大，我们不妨从大到小来做。

首先记录一下每个数所在的位置，将最大的两个数的位置记为 l, r （$l \leq r$）。

不难看出此时的答案是 $r - l + 1$，接着加入第三大的数，并将其位置和 l, r 分别取 $\min$ 和 $\max$，此时分为两种情况。

若 l, r 不变，那么更优的答案不会出现。

若 l, r 中某个发生了改变，说明新的子序列必定有一端是新加进来的数，另一端不变，可以统计答案为$(r - l + 1) - (n - i - 1)$ （i表示第几大的数）

时间复杂度 $O(n)$

1010

注意到是一个排列，考虑第 i 个数 $p[i]$ 作为中位数的情况。

如果 $p[i]$ 是中位数，那么其他数跟它就只有大小关系有用，我们不妨设小于 $p[i]$ 的数为 -1 ,大于 $p[i]$ 的数为 1 。

然后对这个 $-1/1$ 的数组进行一个前缀求和，假设求出来是 sum 数组。

显然如果满足 $sum[r] = sum[l-1]$,那么从 l 到 r 的中位数就是 $p[i]$。

根据这个性质，我们用一个计数数组就可以统计答案了。

时间复杂度 $O(Tn^2)$。

1011

先简化一下题意，给定一个颜色序列，要求支持区间覆盖为一种颜色，一种颜色全部修改为另一种颜色，以及给定区间 $[l,r]$ 求 $\sum_{i=1}^{n} \binom{k}{c_i}$,其中 $c_i = \sum_{j=l}^{r} (a_j == i)$ 。

求这种与区间每种颜色个数都有关的量有一个比较经典的套路，以 $n^{\frac{2}{3}}$ 大小分块，维护任意两块之间的答案以及前 i 个块颜色 j 的个数，对于散块暴力计算每个 a_i 加入后对答案的贡献。对于单点修改，我们枚举任意两块并计算修改对这两块之间答案造成的贡献。为了方便描述，我们将 n, q 视为同阶，则这样处理的复杂度是 $n^{\frac{5}{3}}$ 。

我们考虑拓展单点修改，对于一个颜色相同的连续段的修改，容易发现与单点修改并没有太多区别，计算与目前枚举两个块之间区间的交长度是容易计算贡献的。对于区间覆盖，我们可以暴力枚举该区间的每一个颜色段，对其做一次整体修改操作。容易发现我们一次覆盖如果操作了 $x(x > 1)$ 个连续段，那至少会让序列的连续段减少 $x - 2$ 个。如果正好覆盖了某个连续段的中间，那么会让连续段个数增加 1 。而把一种颜色修改为另一种颜色的操作不会增加连续段的个数，因此可以通过势能分析得到这样暴力修改的操作次数依然是 $O(n)$ 级别的。

但是这样会产生一个问题，就是散块如何 $O(1)$ 查询每个位置的实际值，维护的方法应该有很多种，我的做法是用 set 维护每种颜色的连续段以及序列的所有连续段。set 的遍历可以近似看成 $O(1)$ 从而保证了时间复杂度。

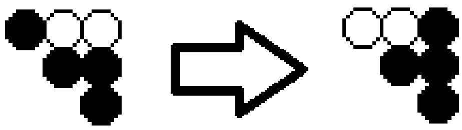
最后考虑一种颜色修改为另一种颜色怎么做，很容易想到可以启发式合并，把连续段较少的颜色修改为连续段较多的颜色。通过之前的分析，如果暴力将每一段按之前的方法修改为另一种颜色的话，我们可以得到一个 $O(n^{\frac{5}{3}}log_2n)$ 的做法，显然不够优秀。但是因为启发式合并极其小的常数，就算刻意卡满也不会慢太多，进行一些卡常操作后有通过的可能。考虑进一步优化，容易发现每次合并我们会连续进行多次把一段的颜色 x 修改为 y 的操作，每一次修改都是 $n^{\frac{2}{3}}$ 的。我们不妨考虑先去除所有颜色 x 与 y 的贡献，全部修改完成后再把合并完的贡献加回来。去除一种颜色的贡献可以 $n^{\frac{2}{3}}$ 完成，之后的每次修改就只需要修改前 i 块颜色 j 的数量，这样单次是 $n^{\frac{1}{3}}$ 的，所以总复杂度变为 $O(n^{\frac{5}{3}}+n^{\frac{4}{3}}log_2n)$ 。

出题人并未进行任何卡常操作也可以在 $5s$ 内通过该题，如果有复杂度正确的做法被卡或者复杂度错误的做法通过，欢迎来找出题人反馈，给大家磕头了。

1012

结论：这个游戏是经典的保加利亚单人纸牌游戏（Bulgarian Solitaire）。若n是三角形数（可以表示为$1+2+\cdots+p$的形式），则只有当$k=1$时有解，答案是长度为1的序列n；否则无解，输出-1。若n不为三角形数，我们设p是满足$1+2+\cdots+p>n$最小的p，令$s=1+2+\cdots+(p-1), r=n-s$，则有解的$k$必须满足条件$k|p,\frac{p}{k}|r$，若$k$满足条件，则答案是项数为$\frac{p}{k}$，公差为$k$，和为$n$的等差数列；否则无解，输出$-1$。$p$可以直接暴力求解，时间复杂度是$O(\sqrt{n})$，同时可以发现输出的序列长度也是$O(\sqrt{n})$级别的，所以总的复杂度是$O(\sqrt{n})$。

证明：首先，我们对游戏进行一个变换：以局面1,1,2和1,3为例，将球堆从左往右按如图方式排列，每一个竖着的列表示一堆，每轮从每堆一个球组成新堆相当于把最左侧对角线上的球移动到最右边一列，原来每堆球数减一相当于原来每堆都向左移动一列。所以游戏规则就相当于每轮每个小球在当前层向左循环移位一格，然后调整一下最右边一列（若小球下方有空隙，让小球向下移动）。

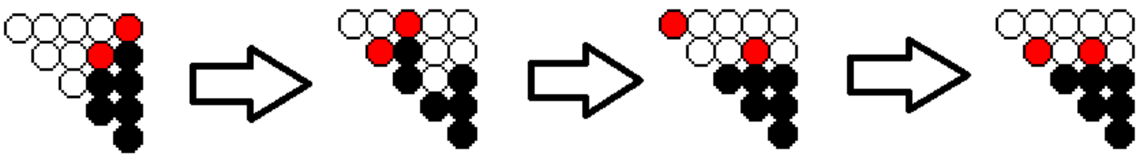


由此，若第i层有一个空隙且第$i+1$层还有小球，只关注这两层，由于两层循环移位的周期不同，第i层的这个空隙一定会被填补；由此，若第i层a个空隙而第$i+1$层及上面还有至少a个小球，那么若干轮以后，第i层的空隙一定会被填满，这也就是三角形数最后一定会稳定在一个局面的原因。而当n不是三角形数的时候，设p是满足$1+2+\cdots+p>n$最小的p，当游戏进入循环后一定是底部$p-1$层被填满，第p层有$r=n-(1+2+\cdots+(p-1))$个小球在循环移位。我们记第p层有球的位置为1，没有球的位置为0，可以得到一个有r个1的01串。不难发现，循环的周期即为这个01串的最小循环节长度。故有解的k必须满足条件$k|p,\frac{p}{k}|r$，表示可能进入每个循环节长度为k，共有$\frac{p}{k}$个循环节，每个循环节中有$r\div(\frac{p}{k})=\frac{rk}{p}$个小球的局面。

接下来，我们来分析一下为什么当n不是三角形数时，对于有解的k，结论中构造的初始局面（长度为$\frac{p}{k}$，公差为k，和为n的等差数列，方便起见，下文称这个初始局面为$A=\{a_1,a_2,\cdots,a_{\frac{p}{k}}|a_1=\frac{nk}{p}-\frac{p-k}{2}=\frac{(k-1)p}{2}+\frac{rk}{p},a_i=a_{i-1}+k,2<=i<=\frac{p}{k}\}$，$A_{i,j}$表示$A$中从左往右第$i$堆球从上往下第$j$个球）为什么一定能满足游戏最终循环长度为$k$。我们先来证明一个引理。

引理：最终第p层的r个球一定是A中每堆顶部的$\frac{rk}{p}$个球，即$\{A_{i,j}|1<=i<=\frac{p}{k},1<=j<=\frac{rk}{p}\}$组成的。

以$n=8,k=2$为例，得到下图。



证明：设f_x为从第x层作为未被填满的最低一层到第x层被填满的轮数，设g_x为从游戏开始到第x层被填满的轮数（只考虑新生成的堆，初始局面下的几堆球不算）。易得$f_x = \begin{cases} 1 & ,1 \le x \le \frac{p}{k} \\ x - \sum_{i=1}^{\frac{p}{k}-1} f_{x-i} & ,\frac{p}{k} < x \le p \end{cases}$，$g_x = \sum_{i=1}^{x} f_i$，在前$a_1$轮总是成立，化简得$f_x = \lceil \frac{xk}{p} \rceil$。发现从开始到第$p - \frac{p}{k}$层被填满的轮数$g_{p-\frac{p}{k}} = \frac{k(k-1)}{2}\frac{p}{k} = \frac{(k-1)p}{2} = a_1 - \frac{rk}{p}$，正是$A_{1,\frac{rk}{p}}$移动到最左侧对角线上的一轮。由于$a_2 - a_1 = k$，从此轮起到$A_{2,\frac{rk}{p}}$移动到最左侧对角线上的轮数为$k$轮，在这$k$轮中，前$\frac{rk}{p}$轮，$A$中第一堆球数仍大于$0$，总堆数为$\frac{p}{k} + p - \frac{p}{k} = p$，并且由于$A$中第一堆小球处在最左侧，所以在进入右侧是最高层的，故A中第一堆顶部$\frac{rk}{p}$个小球全部进入第p层；后$k - \frac{rk}{p}$轮由于A中第一堆球数变为0，堆数-1，总堆数为$p - 1$，故没有球进入第p层。在这k轮中，一直有小球进入第$p - \frac{p}{k} + 1$层，故$f_{p-\frac{p}{k}+1} = \lceil (p - \frac{p}{k} + 1)\frac{k}{p} \rceil = k$依旧成立，这全部$k$轮结束后，第$p - \frac{p}{k} + 1$层被填满，堆数又+1，总堆数为$p$。我们发现，后面每$k$轮都与这$k$轮同理，并且，在这个构造下，$f_x = \lceil \frac{xk}{p} \rceil$将一直成立。如此循环，直至$A$中初始球堆全部为空。由此可得，最终第$p$层的$r$个球一定是$A$中每堆顶部的$\frac{rk}{p}$个球组成的，引理成立。

证明了引理，我们来看看这$\frac{p}{k}\frac{rk}{p} = r$个球在第$p$层的分布。从上面的证明中可以看出，从A中第一堆顶部$\frac{rk}{p}$个球进入第p层开始，每k轮一周期，前$\frac{rk}{p}$轮会连续有小球进入第p层，后$k - \frac{rk}{p}$轮连续没有小球进入第p层。记有球为1，无球为0，由此，第p层会形成$\frac{p}{k}$个形如$\underbrace{11\cdots1}_{\text{rk/p个1}}\underbrace{00\cdots0}_{\text{k-rk/p个0}}$重复形成的01串，显然这个01串的最小循环节长度为k，故游戏最终循环长度为k。

最后，我们来证明为什么A是所有满足条件（使得游戏最终循环长度为k）的初始局面中堆数最小的，且在堆数最小的同时字典序最小的。由于局面和球堆顺序并无关系，下面我们描述的局面都以升序形式排列，以保证其在字典序上的最小性。下面再来证明一个引理。

引理：所有堆数为$\frac{p}{k}$的初始局面中，A是唯一一个能进入长度为 k 的循环的。

证明：假设存在堆数为$\frac{p}{k}$的初始局面$B(B \neq A)$也满足条件。B恰好有$\frac{p}{k}$堆球，若B中存在相邻两堆球数之差不为k，将会导致最终第p层的$\frac{p}{k}$个长度为k的01串中存在相邻两段不相同，导致循环长度不为k。又因为$B \neq A$，所以B必然存在相邻两堆球数之差不为k，所以B不满足条件，引理成立。

我们发现游戏每轮球堆数量最多+1，由于引理成立，所以若有堆数小于$\frac{p}{k}$的初始局面满足条件，则它一定能转移到局面A。我们假设局面C包含$\frac{p}{k} - 1$堆并且其下一个局面为A，由于C共有$\frac{p}{k} - 1$堆球，所以其下一个局面A中必然包含一个球数为$\frac{p}{k} - 1$的球堆。我们有

$a_1 = \frac{(k-1)p}{2} + \frac{rk}{p}, \because k \ge 2, \therefore \frac{(k-1)p}{2} \ge \frac{p}{2}, a_{\frac{p}{k}} > a_{\frac{p}{k}-1} > \cdots > a_1 = \frac{(k-1)p}{2} + \frac{rk}{p} >= \frac{p}{2} + \frac{rk}{p} > \frac{p}{2} > \frac{p}{k} - 1$，可知$A$中每堆的球数都大于$\frac{p}{k} - 1$，这与$A$中必然包含一个球数为$\frac{p}{k} - 1$的球堆矛盾，假设不成立，所以不存在堆数更少的初始局面可以转移到A。由此可知，A是满足条件的堆数最小的初始局面。

综上所述，A是满足条件的最优解。