水题选讲

彭博

广州大学附属中学

2022 年 2 月

AGC040F Two Pieces

有两个不可区分的棋子放在数轴上。初始都在 0 点。

可以进行以下两种操作:

- 选择一个棋子，向右移动一格。
- 把坐标较小的那个棋子移动到坐标较大的棋子的位置。如果两个棋子在相同位置那么仍然可以这样操作。

做 n 次操作，使得一个棋子的位置在 A ，另一个在 B 。求出移动棋子的方案数。

由于棋子是不可区分的，所以两种方案不同当且仅当存在一个时刻 i ，使得棋子的位置集合不同。

$n \leq 10^7$

想象有初始为空的两行。把操作 1 看做给某一行加一个球，把操作二看做加一根棍子把之后的操作和之前的操作割隔开。

因为题目的判重条件，可以认为每一段中第一行的球数都大于等于第二行的球数。那么第一行会有 B 个球，第二行不确定。但是最后一段中第二行会比第一行少 $B-A$ 个球，而我们并不关心前面的段中球数具体怎么分布，因为不管怎么分布，最后一根棍子都会消除所有的差异。

然后转换到坐标系上，用 (x, y) 表示两个棋子的坐标分别是 x, y 。那么操作有三种：

- $x := x + 1$ ；
- $y := y + 1$ ，只能在 $y < x - 1$ 的时候使用；
- $y := x$ 。

操作一用了 B 次，不妨枚举操作二用了 k 次。特判掉 $k + B = n$ 的情况。

假装我们确定了操作一和操作二之间的顺序，现在要加入 $n - k - B$ 个操作三进去。

“只能在 $y < x - 1$ 的时候使用”，相当于除了刚进行完操作三的时候和刚开始的时候，不能碰到 $y = x$ 这条直线。

根据前面所说，最后一次操作三需要保证 $x - y = A - k$ 。而因为以后不会再碰到 $y = x$ 这条直线，所以在只考虑操作一二的时候，只有最后一次碰到 $x - y = A - k$ 时才能使用这最后一次操作三。

同理，只有在最后一次碰到 $x - y = A - k - 1, A - k - 2, \cdots, 0$ 的时候才能使用操作三。并且这些位置是肯定会被碰到过的。而且每次使用的次数是任意的。

那么就把操作三的插入和操作一二给分离了，只需要用卡特兰数算出 (A, k) 的方案数即可。

有 n 个盒子摆成一排，第 i 个盒子的权值是 a_i 。

两人轮流操作，每次操作的方法如下：

- 设上一个人选择的盒子是 i 。如果 i 存在，并且 $i-1, i+1$ 中至少有一个还没被选择过的盒子，那么就在 $i-1, i+1$ 中选一个未被选过的盒子，取走其中的权值。
- 否则，可以任选一个未被选过的盒子，取走其中的权值。
- 如果每个盒子都被选过了则结束游戏。

两人都希望最大化自己拿到的权值。求两人最终分别能拿到多少权值。

$n \le 3 \times 10^5$

首先，先手肯定有一个策略是拿边上的球，然后结果就固定了。称这个策略为基本策略。然后按照盒子个数的奇偶性分类讨论。

奇数

如果先手拿了偶数位置的球，那么不管后手往哪边走，子问题中先手还是先手；否则后手可以使得先手变成后手。注意到先手一定不会使得后面那种情况发生，因为在这种情况下后手可以强制先手只获得基本策略的收益。

所以先手一定会先拿几次偶数位置的球，最后在某个区间中采取基本策略结束游戏。可以设先手的基础收益是所有偶数位置的球，那么最后还要加上区间中奇数位置减偶数位置的球。所以可以二分额外收益，然后以确定能走到有额外收益的区间为胜利。

那么先手的策略可以被写成一棵类似线段树结构的二叉树，每次从一个偶数位置劈开，两边的区间都是必胜态。能造出这么一棵树当且仅当存在一组必胜区间能恰好覆盖所有奇数位置，这个可以随便 DP 。

偶数

现在先手的基本策略的收益是 $\max(odd, even)$ 。由于先手不管走哪里，在子问题中都会被后手强制变为后手，所以最优解就是直接取基本策略。

有 n 个红色石头和 m 个蓝色石头，放在坐标系上。你可以任意次移动蓝色石头，代价是起点到终点的哈密顿距离。你需要满足任意一个红色石头的右上方都有至少 K 个蓝色石头。

$n, m \leq 10^5, K \leq 10$

显然我们只需要考虑外层轮廓线上的红色石头，于是把二维转成类似一维的情况。

先考虑 $K=1$ 的情况。

因为红色石头的位置是一维的，可以用路径来描述一种方案：新建虚点 $n+1$ ，如果一个蓝色石头 i 覆盖了区间 $[l,r)$ ，那么就是走了 $l\to i\to r$ 的路径。代价可以被写在边上。

当然如果这么做那边数就起飞了。

注意到代价可以被写成 $\max(0, RY_l - BY_i), \max(0, RX_{r-1} - BX_i)$ ，那么可以把它写在坐标轴上。

在 y 轴上，向下走需要 1 的代价，向上走不用代价；在 x 轴上向右走需要代价而向左走不用。原点不会连接两个轴。

于是路径就变成 $(0, RY_l) \to (0, BY_i) \to (BX_i, 0) \to (RX_{r-1,0})$ 。然后再把 $(RX_{r-1}, 0) \to (0, RY_r)$ 这条边补上就完美了。

检查一下，不会出现奇怪路径的情况，所以 $K = 1$ 的时候就转化为最短路。

根据经典结论，任何一个 $K>1$ 的合法方案都可以被拆成 K 个 $K=1$ 的方案，所以直接变成费用流即可。

ARC119F AtCoder Express 3

有 $n+1$ 个车站，编号为 0 到 n 。

对于编号在 $[1, n-1]$ 的车站，它们要么是 A 类车站，要么是 B 类车站。而 0 号和 n 号车站则两个都是。

你从 0 号车站出发，每天可以向前或向后走到下一个车站，或是向前或向后走到下一个类型与自己相同的车站。特别地，0 和 n 号车站被视为与任何一个车站类型相同。

给定一个长度为 $n-1$ 的字符串，表示 $[1, n-1]$ 的车站的类型。如果对应位置为 ? 则表示类型未定。再给一个 K ，求有多少种把 ? 替换为 a 或 b 的方式，使得从 0 走到 n 的最少天数不超过 K 。

$n \le 4000$

观察一下，发现如果自己身前有一大段的相同字符，那么用另一种字符跳过去一定是最优选择。

那么说明任意时刻要考虑的字符数量应该不会太多，考虑建一个自动机来求出任意字符串对应的最短步数。

最后发现有用的状态只有 13 个：

- A,AA,AAA,BA...A,AB ，人站在第一个位置。其中 A,AA,AAA 前面一格还会有个 B 。
- 上面 5 种状态取反得到另外 5 种。
- 0A,0B,0 ，表示现在既可以站在 A 上也可以站在 B 上。

手玩一下即可建出自动机，然后暴力做。复杂度 $O(13n^2)$ 。

ARC117F Gateau

有一个长度为 $2n$ 的环形蛋糕，现在要往上面放草莓。

对于每个 i ，都有限制 $i, i+1, \cdots, i+n-1$ 位置上的草莓总数至少是 a_i 。(注意蛋糕是环形的。)

问至少要放几个草莓。

$n \leq 1.5 \times 10^5$

显然可以二分答案。不过二分答案有什么用？

注意到区间长度是 n ，恰好为环长一半，所以取反之后仍然是环长一半。那么在二分答案之后可以把一半的限制转化到另一半。

现在的限制是对于 $i \in [0, n-1]$ ，$[i, i+n-1]$ 中的个数要在 $[a_i, b_i]$ 之中。

一颗草莓可以让任意一个前缀或后缀加一，但是并不知道有多少个是前缀，有多少个是后缀，有点不爽。

如果只有后缀操作呢？容易发现合法当且仅当左边的下界小于等于右边的上界，并且草莓个数大于等于所有下界。

假装我们枚举了给前缀使用的草莓个数。那么维护 (x, mn) 表示有 x 个操作可以超过这里，和右边经过操作之后的上界至少要为 mn 。当前位置能加就加，直到 x 全部用完（即可以通过到下一个位置），或是这里的上界变为 mn 。

不合法情况有两种：一是做的过程中出现某个位置的上界已经小于 mn ，二是做完之后发现剩下的草莓不足以搞定所有下界。

但是注意到，即使给前缀的草莓加一，最多也只能使得剩下的下界减一，所以第二种情况只能代表左边的草莓个数过多。

同样，如果某个草莓个数是合法的，那么给左边多加一些也不会使得第一种情况发生，所以第一种情况只能代表左边的草莓个数过少。

那么就可以二分给前缀多少个，然后 $O(n)$ 判断。总复杂度 $O(n\log^2 A)$ 。

有 n 个朋友住在一条环形的街道上，他们和他们的房子按顺时针标号为 0 到 $n-1$ 。

一开始第 i 个人有 a_i 块石头。他们想让他们之间石头分配得完美均衡：每个人都应拥有相同数量的石头。

改变石头分布的唯一途径是举行会议。在一次会议中，连续 k 个房子的人（记住这条街道是环形的）聚集在同一个地方并带上他们的石头。所有带来的石头可能会在参与会议的人中任意重新分配。会议结束后，每个人回到自己的房子。

找到一种方案使得结束时每个人的石头个数都一样且举行尽量少的会议。

输出举行会议的次数及每次会议的描述。

$2 \le k < n \le 10^5, 0 \le a_i \le 10^4$

CF1237G Balanced Distribution

对于一个环，一个容易想到的构造方法是选择一个点开始，每次把下面 $k-1$ 个位置都摆好，然后以最后那个位置为开头接着做。

怎样分析它的优秀之处呢？这种做法可能会使得某些位置被覆盖两次，但是脑洞一下可以发现，如果在相邻两个人之间插板，那么每块板只会被插一次。

所以现在考虑所有没有被插过的板，从这里切开。对于每个线段，如果长度为 l ，那么一个显然的下界是 $(l-1)/(k-1)$ 上取整。手玩一下发现可以取到。

因为是上取整，所以只要 $(l-1) \neq 0 \pmod{(k-1)}$ 那么就可以把这段和相邻的某一段合并，答案不会更劣。

另外一个限制是每段之和都等于 $l \cdot avg$ ，相当于每个位置都减去 avg 之后和都为 0 。

现在已经可以随便倍增了。

交互题。

有一个竞赛图，每条边的颜色是红色或绿色。红色边及它们的方向会告诉你。

进行不超过 $2n$ 次询问，每次询问一条绿边的方向。找到一个点 S ，使得对于任意一个点 x ，都存在一条边的颜色相同的路径 $S \to x$ 。

$n, m \le 10^5$

CF1142E Pink Floyd

CF1142E Pink Floyd

考虑没有红边的时候怎么做。

维护一个 root ，和一个暂时不能从 root 走到的点集 S 。

每次从 S 里面拿出一个点 x ，询问 $(root, x)$ 。如果 $root \to x$ 就把 x 删掉，否则令 $root = x$ ，然后把 x 删掉。

有红边的时候，两点之间不再一定有绿边，所以需要想办法利用红边。

CF1142E Pink Floyd

考虑把红边缩点，每个强联通分量只用一个代表点表示。维护所有入度为 0 的强联通分量的代表点组成的集合 S 。

任意选一个代表点作为 root ，并从 S 里面删掉。如果此时 S 为空那么 root 就是答案。

每次仍然询问 $(root, x)$ ，但是此时不能简单地删掉 x 所在的一整个强连通块，因为 root 经过绿边走到 x 之后就不能再往 x 所在的强联通分量走了。

可以把 x 从图中删去，然后重新跑一遍 tarjan ，加入新的代表点，但是这样很麻烦，而且复杂度也爆炸了。

更好的解决办法是保留这个强联通分量，仅仅删除 x ，然后从里面随便再找一个点作为代表点。

如果这个强联通分量被删完了那么就把新的入度为 0 的强联通分量的代表点加入 S 中。

由于每次询问一定会删掉一个点，所以总询问次数最多是 $n-1$ 。

给定 n, K 和一个长度为 n 的数列 b 。求有多少个值域在 $[0, n]$ 的长度为 n 的数列 a ，使得

$$\forall 1 \le i \le n, |\mathsf{MEX}(a_1, \cdots, a_i) - b_i| \le K$$

模 998244353 。

$n \le 2000, K \le 50$

CF1608F MEX counting

考虑 $K=0$ 怎么做。

$\text{MEX}(a_1,\cdots,a_i)=b_i$ 蕴含两个限制，一个是 b_i 没有出现，另一个是 $[0,b_i)$ 都出现过。前者容易处理，而后者只需要容斥一下。

所以可以容斥哪些数在该出现的时候还没出现，然后就可以 DP 计数了。

$K\neq 0$ 几乎完全一样，只是每次要额外枚举前缀的真实 MEX 是多少。

转移细节留给读者思考。

UOJ84. 【UR 7】水题走四方

给定一棵有根外向树（所以人只能向深度更深的点走）。

你可以控制两个人。每个时刻可以让某个人走一步，也可以让两个人同时走一步。

另外，你也可以让其中一个人瞬移到另外一个人的位置。瞬移不花费时间。

求最少要多少时间，能让每个点都被走过至少一次。结束时两人可以在任意位置。

$n \leq 5 \times 10^6$ 。

两个人同时站在根节点，现在能有什么操作呢？

如果还剩余多于两棵子树，那么显然只能派一个人下去走。如果剩两棵子树，但都不是链，那么也还是只能有一个人下去走。否则可以一个人去走链，另外一个人走另一棵子树，然后把走链的那个人传送过来。

设总是被传来传去的那个人是 A ，只需要往下走的那个人是 B 。

我们希望能够用“dp_x 表示此时两人都在 x ，且 x 子树完全没动，的最小时间”来递推，所以考虑这是否可行。

发现这确实是可行的，因为 B 往下走了一段路之后再把 A 接过来，此时子树内必然没有走过，否则肯定不优。

于是获得了 $O(n^2)$ 的暴力做法。

我们还发现 A 走完链的时候必须立即传送过去，否则也是不优的。所以最后走的这条链肯定是除 B 所在子树外最深的叶子。

可是这样从下往上转移还是不好处理，所以转换思路，设 f_x 表示从根转移到 dp_x 期间用的最小花费。这样就是从祖先转移了。

如果要从 anc 转移到 x ，那么需要存在一个叶子 l ，它在 anc 子树内，却不在 x 子树内，且深度大于等于 x 。

发现如果有多个合法的 anc ，选择最深的那个一定不亏。于是问题转化为对每个点求最深的合法 anc 。

自底向上合并，维护子树中哪些点还没有找到自己的 anc 即可。

UOJ177. 新年的腮雷

给定可重集合 A, B ，其中 $|A| = n, |B| = m$ 。每次可以在 A 中选出 m 个元素 $x_1, \cdots, x_m$ ，合并成一个元素 $\max\{x_i + b_i\}$ ，放回 A 中。

保证 $n = 1 \pmod{m - 1}$ ，所以最终一定可以合并成一个元素。求最终这个元素的最小值。

$n, m \leq 5 \times 10^4$ 。

UOJ177. 新年的腮雷

先把 b 从小到大排序，把 a 从大到小排序。

正着做太难了，完全不知道策略应该长啥样。

不妨二分答案之后倒着做，维护一棵 m 叉树，每次把一个叶子拆成 m 份。最后需要满足每个 a 都可以匹配一个比自己大的叶子。

问题又出现了：现在应该拆哪个叶子呢？

UOJ177. 新年的腮雷

考虑最大的那个叶子 x ，如果不拆它，那么它就要匹配掉目前最大的 a 。直觉告诉我们如果把它拆掉之后最大的 a 仍然能被匹配，那么似乎拆掉会更优。

当然，策略不会这么简单。

经过一番冷静分析，我们考虑 $x - b_1$ ，它是以后能拆出的最大值了。考虑所有 $a_i > x - b_1$ ，如果它们在拆掉之后无法找到能匹配的叶子，那么这个 x 肯定是不能拆的。

否则是不是一定能拆呢？假设没有拆它，而是拆了更小的 y ，发现总是可以存在一种对应方法，使得拆了 x 之后剩下的叶子对应比拆 y 之后的叶子更大。

然后拿一棵线段树维护即可。

给定一棵 n 个点的树（边带权），你要在选择恰好 K 个互不相同的点 $p_1, \cdots, p_K$ ，最大化

$$\sum_{i=1}^{K} \mathsf{dis}(p_i, p_{i \bmod K+1})$$

$n \le 2 \times 10^5$

容易想到，只要枚举了这 K 个点的重心，就很容易用贪心求出最优解。每次选择距离重心最远的点，而且强制不存在一个子树超过一半即可。

另外，只要在贪心的过程中从来没有出现过“一个子树的点数超过一半”的问题，就可以确定这就是任何一个重心的最优解。

这时候就已经可以猜想，一旦出现了“一个子树的点数超过一半”，那么最优重心就必然在那个方向。

这在大部分情况下都是正确的，但是 K 是奇数时会出现一些问题。不过这个不会有太大问题，判一下就好了。

因此，只需要对原树进行点分治，就可以 $O(n\log^2 n)$ 解决问题。

给定 c, d ，定义 n 阶矩阵 A ，其中 $A(i,j) = \gcd(i,j)^c \operatorname{\mathsf{lcm}}(i,j)^d$ 。

q 次询问，每次给定向量 b ，求解方程 $Ax = b$ 。如有多解输出任意一解，也要判断无解。

$nq \le 3 \times 10^5$ 。

把 lcm 拆开，变成 $A(i,j)=\gcd(i,j)^c i^d j^d$ 。j^d 可以和 x_j 合并在一起，而 i^d 可以和 b_i 合并在一起。经过转化之后就只有 $A(i,j)=\gcd(i,j)^c$ 了。

直接做似乎无从下手，考虑对于确定的 x 如何计算得到的 b 。

推式子。

$$
\begin{aligned}
b_i &= \sum_j \gcd(i,j)^c x_j \\
&= \sum_{t|i} t^c \sum_{j=1}^{n/t} x_{jt} \sum_{k|j,k|(i/t)} \mu(k) \\
&= \sum_{t|i} t^c \sum_{k|(i/t)} \mu(k) \underbrace{\sum_{j=1}^{n/kt} x^{jtk}}_{y_{tk}} \\
&= \sum_{T|i} y_T \sum_{t|T} t^c \mu(T/t)
\end{aligned}
$$

设后面的东西是 $f(T)$ ，就有

$$(y \cdot f) * I = b$$

就可以解出 y ，然后进一步解出 x 了。

有 n 堆石子，第 i 堆有 a_i 个。两个人玩游戏，设这两个人分别是 A 和 B 。A 先手。

当 A 要操作时，由 B 选取一个非空的石子堆，然后 A 从中拿走任意多个石子。同理，当 B 要操作时，由 A 选取一个非空石子堆，然后 B 从中拿走任意多个。

如果轮到某人操作时没有石子了那么这个人输。问谁赢。

多组数据，$T \leq 200, n \leq 10^3, a_i \leq 10^9$ 。

CC DESTRUCT

不妨枚举 B 第一次选哪个，然后游戏规则就变成

- 从当前石子堆拿走若干个，然后选定下一次的石子堆。
- 如果拿完之后没有石子了那么获胜。

考虑拿完当前这一堆。如果拿完之后是必胜的那么就赢了。否则，可以拿到只剩一个，然后强迫对方还是拿这一堆，那么就还是赢了。

所以只要当前这堆不是只剩一个就赢了。

所以判一下 1 的个数的奇偶性，以及是否存在非 1 的堆即可。

CF1268D Invertation in Tournament

给一个竞赛图，你可以选择若干个点组成点集 S ，那么一条边的方向会被反转当且仅当它的一个端点属于 S 而另一个端点不属于 S 。求为了让这个图强连通所需的最小的 $|S|$ 。

$n \leq 2000$

CF1268D Invertation in Tournament

由于竞赛图的特殊性质，容易感性理解得到大部分情况下都只需要操作一个点。不过具体操作需要自己手玩来得到，所以这里直接给出定理和证明。

关于竞赛图有三个性质：

1. 竞赛图缩点之后是一条" 链"，满足前面的点都向后面的点有连边。
2. $n > 3$ 个点的强联通竞赛图存在大小为 $n - 1$ 的强联通子图。
3. 一个竞赛图强联通当且仅当把点按出度排序后不存在 $k < n$ 使得前 k 个点的出度之和是 $\binom{k}{2}$ 。

性质一的证明：随便手玩一下。

性质二的证明：归纳证明 n 个点的强联通竞赛图存在 $[3,n]$ 的哈密顿回路。

性质三的证明：考虑缩点之后的最后一个块，里面的点的出度比其他点的出度都更小。

获得这么些性质之后可以发现:

1. 缩点之后有超过 3 个块，那么设开头为 H，结尾为 T，反转中间任意一个点 x，对于其他任意一个点 y，都有路径 $H \to y \to T \to x \to H$，所以最小操作 1 次。
2. 如果一个块 X 至少 4 个点，那么可以选一个点 x 出来使得反转之后它还是个块，然后有路径 $x \to (X - x) \to y \to x$ 或 $x \to y \to (X - x) \to x$。

只要满足这两条中的任意一条，就都可以枚举反转哪个点，然后用性质三判断。

否则，一定有 $n < 7$，直接暴力即可。

Alice 和 *Bob* 在玩一个游戏. 游戏中有 n 堆筹码, 其中第 i 堆中有 v_i 个筹码.

游戏开始前， *Alice* 先从区间 $[1, m]$ 中选择一个正整数 a, 随后 *Bob* 同样地选择一个正整数 b.

游戏开始. 在 *Alice* 的回合中, 她可以选择任意至少包含 a 个筹码的一堆, 然后从中取出 a 个筹码. 同理, 在 *Bob* 的回合中, 他也可以选择任意至少包含 b 个筹码的一堆, 然后从中取出 b 个筹码. 直到某回合不能继续操作的一方判负.

假设双方每次都会选择最优策略, 游戏结果将只由 a, b 和 决定. 考虑一对选定的 (a, b) , 则有四类可能的局面: *Alice* 必胜、*Bob* 必胜、先手必胜、后手必胜。

在所有合法的 (a, b) 中 (即所有满足 $1 \leq a, b \leq m$ 的 (a, b) 中), 请问每类局面各出现多少次.

$n \leq 100, m \leq 10^5$

CF1033G Chip Game

注意到当一个人取一堆石子时，另一个人可能可以选择跟着取这堆石子，使得这一堆石子减掉 $(a+b)$ 。

那么我们先把所有石子都模掉 $(a+b)$ ，然后看此时的赢家能不能保证在原游戏胜利。

发现是可以的：必胜者该干啥干啥；必败者如果不按套路出牌那么必胜者可以在这堆石子跟一步。

那么枚举 $a+b$ ，然后大力分类讨论。

令 $N=\prod_{i=1}^{n}p_i^{a_i}$，$M=\prod_{i=1}^{n}p_i^{b_i}$，其中 p_i 是两两不同的素数。
求将 N 表示成 k 个正整数的乘积的方案数，也就是 $N=\prod_{i=1}^{k}x_i$ 的解的个数，答案对 10^9+21 取模。

对于子问题 1，要求对于所有整数 i 满足 $1\le i<k$，都有 $x_i<x_{i+1}$。对于子问题 2，要求对于所有整数 i 满足 $1\le i<k$，都有 $x_i\le x_{i+1}$。

对于两个子问题都要求对于所有整数 i 满足 $1\le i\le k$ 都有 $x_i\nmid M$，即 x_i 不是 M 的约数。

$n\le 50, k\le 25, a,b\le 10^{18}$ 。可以发现 p 的值和答案无关。

用斯特林反演，发现子问题 1 和子问题 2 没啥区别，都是对于所有 K 的整数拆分求方案数。

那么考虑一组拆好了的等价类，如何计算方案数。

使用生成函数来描述。对于一个大小为 t 的等价类，它对应的生成函数是

$$(\prod \frac{1}{1-x_i^t})(1-\prod(1-x_i^{t(b_i+1)}))$$

而我们希望把所有等价类的生成函数乘起来，求 $[\prod x_i^{a_i}]$ 这一项的值。

UOJ99. 【集训队互测 2015】普罗达科特

自然的想法是希望不同的 a 之间独立。那么可以容斥哪些等价类不满足条件，相当于直接把 $1-\prod(1-x_i^{t(b_i+1)})$ 拆成 1 和 $\prod$ 。然后把每个 a 的方案数乘起来即可。

计算一个 a 的时候暴力把分子拆开成为关于 x^{b+1} 的多项式，每一项和分母组合起来分别做。另外还可以注意到把分子看做 x^{b+1} 的多项式之后每个 a 的分子是相同的，于是暴力拆开的复杂度不必和 n 乘在一起。

小结：

经过斯特林容斥和普通容斥，任务转化为对于每个 K 的整数拆分 $p_1,\cdots,p_m$ ，和 $1\le i\le n, 1\le j\le K$ ，求

$$[x^{a_i-j(b_i+1)}]\prod_{k=1}^{m}\frac{1}{1-x^{p_k}}$$

而转化的复杂度是什么？在普通容斥的过程中可以加一点优化：相同大小的等价类只需要枚举有几个被容斥掉了。那么每种拆分的复杂度之和设为 Q_K ，在 $K=25$ 时为 129512 。

然后还需要爆拆多项式，以及对于每个 a_i 分别计算，所以转化的复杂度是 $(nK+K^2)Q_K$ 。

那么现在要计算

$$[x^{a_i-j(b_i+1)}]\prod_{k=1}^{m}\frac{1}{1-x^{p_k}}$$

无脑做法是直接乘在一起，然后线性递推。然而过不了。

事实上，要求这个多项式的第 a 项，在 $a \bmod \mathrm{lcm}(p_k)$ 固定的情况下，是一个关于 a 的 $m-1$ 次多项式。只需要把前 $m\ \mathrm{lcm}(p_k)$ 项求出来，就可以在 $O(m)$ 的时间内插出任意一个 a 。

有一个 n 阶排列 p 。你和老王轮流操作这个排列，老王先手。

在第 i 步，老王会交换 p_{x_i}, p_{y_i} 。然后你可以任意交换两个位置（可以不操作）。

你的目标是把 p 排序，并且用的步数尽可能少。如果第 $i+1$ 步开始前 p 已经有序了那么就是 i 步。

你已经把老王看透了，所以你知道所有的 x_i, y_i 。

$n \le 2 \times 10^5$ ，保证最优方案可以在 n 步之内完成。

注意到答案显然有可二分性。所以先二分答案。

注意到两次交换 $(a,b),(c,d)$ 的顺序是可以交换的，得到 $(c,d),(a',b')$ 。并且通过合理选择 (a,b) 就可以任意选择 (a',b') 。

因此可以直接把老王的操作全部拉到前面，然后专心干自己的。答案很容易就出来了。

构造方案也不难。

XX Open Cup GP of Zhejiang – J

你需要在 $\{1,2,\cdots,n\}$ 里面选出一个大小至少为 $\lfloor\sqrt{0.5n}\rfloor$ 的子集 S ，使得每一对 $(a,b), a<b, a,b\in S$ 得到的 $a\oplus b$ 都互不相同。

$n\le 10^7$

取一个最小的 q 使得 $2^q \geq |S|$ 。显然 $2^q < 2|S|$ 。

随机一个以异或为加法的域 F_{2^q} ，然后对于每一个 $0 \leq x < |S|$ ，设 $f(x)$ 是在 F^{2^q} 里面的 x^3 ，把 $2^q x + f(x)$ 丢进 S 。

显然 $2^q x + f(x) \leq n$ ，而且数量也刚好是 $|S|$ 个，所以只需要证明不存在 $a \oplus b = c \oplus d$ 即可。

我们在 F_{2^q} 的意义下考虑。接下来用 + 来代替 $\oplus$ 。

假设有一组撞上了，就说明存在 $0 \le i, j, k, l < |S|$ 使得 $i + j = k + l$ ，且 $i^3 + j^3 = k^3 + l^3$ 。后者又可以推出 $(i + j)(i^2 + ij + j^2) = (k + l)(k^2 + kl + l^2)$ ，进而 $i^2 + ij + j^2 = k^2 + kl + l + 2$ 。

然后 $(i + j)^2 = i^2 + 2ij + j^2 = i^2 + j^2$ ，所以 $ij = kl$ 。

但是 $i + j = k + l, ij = kl$ 在一个域中已经可以直接推出 $(i, j) = (k, l)$ ，矛盾。

因此这个构造是合法的。

这是一个交互题。

有一个“圆方树”，所有叶子都是圆点，所有非叶子都是方点，且方点度数至少为 3 。一共有 n 个圆点，方点数量未知。边带权。

你可以询问两个圆点的距离。

你需要确定

- 树的半径的长度。定义 $r(C)$ 表示离点 C 距离最远的圆点的距离，那么半径定义为 $r(C)$ 的最小值。不是直径除以二。
- 是否存在一个 $r(C)$ 最小的 C 使得 C 是带权重心。带权重心表示每个子树都有不超过 $\lfloor n/2 \rfloor$ 个圆点。

你可以询问 $\lceil 7n/2 \rceil$ 次。$6 \leq n \leq 110$ 。

先用 $2n$ 次询问得到直径 (a, b) 以及 a, c 到每个点的距离，其中 c 一定更靠近 b 。

现在已经可以确定每个点在直径上的位置了，所以半径也很容易求。

容易发现两个直径中点和一个直径中点区别并不大，下面只考虑一个直径中点的情况。现在只需要判断它是不是带权重心。

使用经典的” $O(n)$ 在线判断是否存在严格众数 “的算法，容易用 $2n$ 次操作来判断是不是带权重心。不过现在刚好超了 $0.5n$ ，考虑怎么砍下去一点。

后面 n 次操作是用来数求出来的那个子树是不是严格众数的，但是这样没有利用到前面 n 次操作的信息，非常浪费。

注意到在前面只要发现两个点是同一个子树的，就可以少一次操作。容易发现至少有 $n/2$ 次这样的发现，所以次数变成 $1.5n$ 了。总次数 $3.5n$ 。

CF1458D Flip and Reverse

给定一个 01 串，每次可以选择一个 01 个数相等的区间，把它翻转并反转（$001101 \to 010011$ ），求能得到的拓扑序最小的串。

$|s| \le 5 \times 10^5$

CF1458D Flip and Reverse

考虑建图，在数轴上从 0 开始走路，见到 0 就往右走一步，否则往左走一步。走路的时候连有向边。最终走出一条欧拉路径。

考虑一次操作在干什么。区间 01 数量相同，即走出了一个环。一次操作就是把环上的边全部反向，走的顺序也反向。

然后大胆猜想，如果把图变成无向图，那么任意一个欧拉路径都是合法的。

证明可以考虑两条欧拉路径的第一个分叉点 x ，然后搞一下，就可以把分叉点的时间往后推。

字典序最小的欧拉路径可以贪心。

给定长度为 n 的序列 a 和长度为 m 的序列 b 。

有一个 $n \times m$ 的网格图，网格图上的点 (i, j) 能走当且仅当 $a_i + b_j \geq 0$ 。并且在这个网格图上只能向右或向下走。

问有多少个 (S, T) ，使得 $(S, 1) \to (T, m)$ 的路径存在。

$n, m \leq 2 \times 10^5$

地图上能走的位置很难描述，比较诡异。

但是找最优路径并不需要描述地图。可以想到一个贪心：从 (x, y) 出发，尝试走到 (x, z) 使得 $b_z > b_y$ ，或者走到 (z, y) 使得 $a_z > a_x$ 。只要存在这样一种走法（并且路上不存在更大的位置），那么直接走过去显然是好的。

先假设 $S = 1, T = n$ ，看怎么判是否合法。可以找到一个 (X, Y) ，使得 $a_X = \max a, b_Y = \max b$ ，然后就只需要判断能否从 $(1, 1)$ 走到 (X, Y) ，以及从 (n, m) 倒着走到 (X, Y) 。这个判断就可以用前后缀最大值什么的贪心。

那么如何对所有 S,T 判断呢？注意 Y 是不变的，而且 m 这一维的所有前后缀最大值都不会变。然后 S,T 之间会有一些变动，但是并不难用数据结构处理。

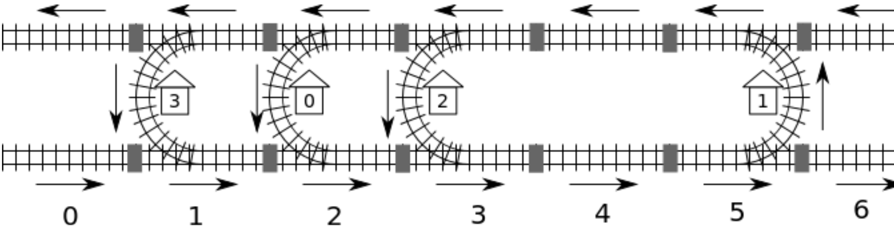
当然，我是不会在这里涉及太多数据结构的细节的（

最后可以抽象出四个条件：

- $\forall i, a_i + b_Y \geq 0$
- $\forall j, a_X + b_j \geq 0$
- $\forall (i,j), (\exists k < i, a_k + b_j \geq 0) \vee (\exists k < j, a_i + b_k \geq 0)$
- $\forall (i,j), (\exists k > i, a_k + b_j \geq 0) \vee (\exists k > j, a_i + b_k \geq 0)$

对每个条件分别可以得到若干个不合法的矩形，然后扫描线即可。

UOJ24. 【IOI2014】Rail



本题是交互题。

铁路被分成了若干段，每一段可能有车站。如果有车站，那么可能是 C 类或 D 类，分别表示不同的轨道弯曲方向和火车走的方向。

两个车站之间的距离定义为从一边走到另一边所经过的最少的连接器数量。连接器就是上面的灰色方块。(容易发现 $dis_{x,y} = dis_{y,x}$ ，所以就不用区分从哪到哪了。)

你现在只知道车站 0 所在的位置，以及它是 C 类。你可以询问两个车站之间的距离，求出所有车站的位置和类型。

$n \leq 5000$ ，询问次数 $\leq 3(n-1)$ 。

首先把两个站点之间的路线分析一下，发现有 4 种情况。对着这 4 种情况想做法会比较清晰。

先把 $dis_{0,i}$ 全部求出来，设 p 是其中的最小值，那么 p 就是 0 右边第一个 D 类站点。

再把 $dis_{p,i}$ 全部求出来。此时对于一个 x ，如果 $dis_{0,x} = dis_{0,p} + dis_{p,x}$ ，那么说明 x 在 p 左边，否则在右边。在 p 左边的站点包含在 0 左边的和在 $0, p$ 之间的，不过这两者很容易区分。

因为左右是对称的，下面就只考虑右边了。

因为铁路的特性，我们发现，如果一个 C 类站点 x 右边的第一个 D 类站点是 y ，那么从左边看来，x 和一个在 $2loc_y - loc_x$ (即关于 y 对称) 的 D 类站点是没有区别的，因此从左边尝试分辨 C 和 D 没有前途，只能在右边的内部分辨。

把所有站点按 $dis_{0,i}$ 排序，暂时认为它们都是 D 类站点，给它们确定一个 loc'_i 。从左往右扫，设上一个确定是 D 类站点的位置是 loc_q 。对于当前考虑的 x ，询问 $d = dis_{q,x}$ ，那么它可能是 loc'_x 的一个 D 类站点，也可能是 $loc_q - d$ 的一个 C 类站点。设这两个位置分别为 p_1, p_2 。

我们发现，如果是后者，那么说明 $(p_1 + p_2)/2$ 的位置有一个 D 类站点（这样才能从 p_1 对称到 p_2 ）；而如果是前者，那么为了使得 x 到 p 的距离是 d ，就需要 $(p_1 + p_2)/2$ 有一个 C 类站点。

只要存在 D 类站点，就一定已经确定了。所以只需要看 $(p_1 + p_2)/2$ 是否确定过一个 D 类站点即可。

给定一个无向图，把它定向成 DAG ，然后选取两个拓扑序使得对应位置相等的个数最少，输出方案。

$n, m \le 5 \times 10^5$

对于一个 DAG ，显然答案的下界是这样的点 x 的个数：对于任意一个其他点 y ，要么存在路径 $x \to y$ ，要么存在 $y \to x$ 。

那么对于一个无向图，下界就是度数为 $n-1$ 的点的个数。然后用构造的方法证明这个下界可以取到。

取补图，然后对每棵树分别构造即可。一种简单的构造方法是划分成若干个菊花。

XX Open Cup GP of Zhejiang – L

给定 n, K ，求 $\sum_{i=1}^{n} \mathsf{lcm}(i, i+1, \cdots, i+K)$ 。

$n \le 10^{18}, k \le 30$

设 $\text{lcm}(i, i+1, \cdots, i+K) = \frac{1}{c} i(i+1)\cdots(i+K)$ ，那么容易发现 c 只和 $i \bmod \text{lcm}(1, 2, \cdots, K)$ 有关。可惜 K 太大了。

不过，显然每个质数 p 对 $\frac{1}{c}$ 的贡献是独立的。设 k 为最大的使得 $p^k \le K$ 的数，那么 p 对 $\frac{1}{c}$ 的贡献只和 $i \bmod p^k$ 有关。

打个表会发现，对于很多个 $i \bmod p^k$ ，它们的贡献其实是相同的。因此我们可以选择其中最大的一组设为默认值，而特殊考虑其他的数。

这样，我们就把等价类个数从 $\text{lcm}(1, 2, \cdots, K)$ 降到了大约 10^8 个。可惜这还是太多了。

继续利用不同质数比较独立的性质，考虑 meet in the middle 。只有 10 个质数，所以可以 2^{10} 枚举每个质数是否会对最终模数产生贡献，然后左右两边的合并就比较简单了。

因为有自然数幂和，所以合并的时候再乘一个 k^2 的复杂度，问题不大。

CF1534H Lost Nodes

这是一个交互题。

首先给你一棵 n 个点的树。

交互库手上有两个点 a, b ，并且给了你链 a, b 上的一个点 f 作为提示。

你每次可以选择一个点 x 向交互库提问：如果以 x 为根，那么 a, b 的 lca 是谁。

你需要确定 a, b 。

在告诉你 f 之前，你需要求出在最坏情况下，需要使用多少次操作。然后你就只能使用不超过这么多次操作了。

$n \leq 10^5$

不妨先枚举 f ，考虑已知 f 时在最坏情况下要多少次。

同时有 a 和 b 其实和只有一个差不多，所以接下来考虑只有 a ，而 $b=f$ 的情况。

此时，只要把 f 设为根，询问一个点 x 的结果就是 $\mathrm{lca}(x,a)$ 。

然后无脑设 dp_x 表示为了确定 a 是否在 x 子树内，以及如果在则确定 a 是多少，要用几次询问。

容易发现，站在 x 这里，一定是按照儿子的 dp 值从大到小问。那就乱搞就行了。

给定 n, K 。有一棵以 1 为根的 n 个点的有根树，i 的父亲是 fa_i 。你要在每个点上写一个正整数 a_i ，使得

- $\forall 2 \le i \le n, a_{fa_i} \bmod a_i = 0$
- $\prod a_i \le K$

求写正整数的方案数模 998244353 。

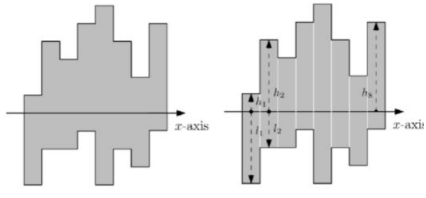
$n \le 10^3, K \le 10^{12}$

对每个质数分别考虑，可以得到 f_k 表示凑出 p^k 的方案数。

注意到，除非这个质数只在 1 出现，否则必然有 $k>1$ 。

所以可以先强行令 $a_1=\text{lcm}(a_2,\cdots,a_n)$ ，这样得到的数必然每一个质数至少出现两次，也就是 powerful number 。这样的数只有 $O(\sqrt{K})$ 个，全部枚举出来，然后再给 a_1 随便乘上点什么就行了。

有一个长成这个样子的铁片。



你每次可以横向或者纵向切一刀，把一个铁片分成两个部分。这一刀不能横跨多个铁片。

问最少切多少刀才能使得每一个部分都是长方形。

$n \leq 2.5 \times 10^5$

通过手玩，可以想起来一个铁片是长方形当且仅当它是凸包，从而我们的目标就是使得没有向内凹的角。

可以发现一次总是能消除一个向内凹的角，所以就是要最大化一次消掉两个的刀数。

发现这样的刀只有 $O(n)$ 种，而且只有横竖会相交。

想起来二分图的最大独立集就是 $n-$ 最大匹配，而最大匹配可以贪心。

做完了。

有一棵 n 个点的树，树上有 K 个老鼠和 K 个洞。保证老鼠位置两两不同且不相邻，洞也是。

每个时刻一个老鼠可以往相邻的点移动一步。移动之后仍然需要保证两两不相邻（即使处在洞中也不能相邻）。

问能否让最终的状态每个洞都有恰好一只老鼠。

$n \leq 2000$

注意到操作是可逆的，所以可以选取一个中间状态，只要两边都能走到这个中间状态就合法。

一个显然的直觉是要把一些老鼠尽量往叶子塞，这样它们就能对其他老鼠造成尽可能小的影响。

因此可以随便拿一个挂在边上的菊花出来，然后看两边是否都能往里面同一个点塞一只老鼠。不管是否可以，塞完之后这个菊花都不会再对外界造成任何影响了。

容易 $O(n)$ 判断是否可行，所以复杂度 $O(n^2)$ 。

给定一个长度为 n 的数列 a 。你从第 i 个点可以一步跳到 $[i+1, i+a_i]$ 的任意一个点。

q 次询问 l, r, k ，求出删去至多 k 个点之后（点会重新编号），从删点之前的 l 跳到 r 最少要几步。

$n, q \leq 2 \times 10^4$

有一个哈希函数，描述如下：

设 A 是一个 $2n$ 位的非负整数，二进制表示为 $(a_{2n-1}a_{2n-2}\cdots a_1a_0)_2$ 。

从 A 造一个 B 出来，B 也是 $2n$ 位非负整数，其二进制表示 $(b_{2n-1}b_{2n-2}\cdots b_1b_0)_2$ 满足

- $b_i = a_i \oplus a_{2i+1}, 0 \le i < n$
- $b_i = a_i \oplus a_{4n-2i-2}, n \le i < 2n$

然后又从 B 造一个 C 出来，$C = B \oplus R(B)$ ，其中 $R(B)$ 是把 B 循环右移一位的结果。

最终，定义哈希函数 $h(A) = (239A + 153C) \bmod (2^{2n-1} - 1)$ 。

给定 n, H ，求一个 A 使得 $h(A) = H$ 。保证答案存在。

$n \leq 16$

$2n = 32$ ，就差一点就能枚举了。

考虑 meet in the middle ，但是因为哈希函数比较诡异，朴素做法很难合并两边。

注意到一个 a_i 最多只会影响 C 的 4 个位，并且这 4 个位的位置还挺有规律。设 s_i 表示 a_i 能影响到的 c_j 的集合。

把 $2n$ 个位分成三个集合 S_1, S_2, S_3 ，使得 $\forall x \in S_2, \forall y \in S_3, s_x \cap s_y = \varnothing$ 。这样在枚举了 S_1 的状态之后就可以分别枚举 S_2 的状态和 S_3 的状态，它们对 $239A + 153C$ 的贡献是相互独立的。

复杂度 $O(2^{2n-\min(|S_2|,|S_3|)})$ 。

一个简单的分类方法是

$$S_2 = \{x \mid \max(s_x) < n\}$$

$\mathsf{S}_3 = \{y \mid \min(s_y) \geq n\}$ $\min(|S_2|, |S_3|) = 6$ 。

CF1515I Phoenix and Diamonds

有 n 种物品，第 i 种物品的体积是 w_i ，价值是 v_i ，有 a_i 个。

q 次询问，每次有三种可能：

- 第 i 种物品增加 k 个。
- 第 i 种物品减少 k 个。
- 给出总体积 W ，问“贪心”地拿物品能拿多少价值。贪心：先拿价值最大的，若相同则拿体积最小的。

$n \leq 2 \times 10^5, q \leq 10^5$

CF1500F Cupboards Jumps

给定一个长度为 $n-2$ 的序列 w ，你需要构造一个长度为 n 的序列 a ，或报告无解。

a 要满足：$\max(a_{i-1}, a_i, a_{i+1}) - \min(a_{i-1}, a_i, a_{i+1}) = w_i$ 。

$n \le 10^6$

给定一棵 n 个点的树，每个点有一个集合 S_i ，初始 $S_i = \{i\}$ 。

有 q 次操作，每次选定树上的一条边 (u, v) ，令 $S_u := S_u \cup S_v, S_v := S_u \cup S_v$ 。

最后对于每个 i 输出，有多少个 j 使得 $i \in S_j$ 。

$n, q \leq 2 \times 10^6$

有一个 n 个点的完全图 G ，每条边是黑色或白色。

你需要找到一个哈密顿回路，使得这个回路上边的颜色恰好被分成两段，一段是黑色，另一段是白色。

$n \le 3000$

给出一棵树，设第 i 个点的度数是 d_i 。

问有多少对 (x, y) ，使得存在一种选出 x 个点的方式，它们的度数加起来恰好为 y 。

$n \le 2 \times 10^5$

树的具体形态显然用处不大：只要 $\sum d_i = 2n - 2$ 都可以构造出一棵合法的树。

我们的主要问题出在多了一维 x 。因为什么性质也找不出来，尝试把 x 这一维消掉。

先把所有 d_i 都减掉 1 ，那么 $\sum d_i = n - 2$ 。我们尝试证明：对于总和 y ，设 $m(y), M(y)$ 分别是最小和最大的 x ，那么 $x \in [m(y), M(y)]$ 都有合法方案。

我们的证明主要和 0 的个数 z 有关。因为 $m(y)$ 必然一个 0 都没选，而 $M(y)$ 选了 z 个 0 ，所以只要 $M(y) - m(y) \le 2z$ ，就可以通过调整 0 的个数构造出每个 x 。

发现对于任意一个集合 S ，都有 $-z \le y(S) - x(S) \le z - 2$ ，所以 $-z \le y - M(y) \le y - m(y) \le z - 2$ ，所以 $M(y) - m(y) \le 2x - 2$ ，然后就证完了。

你有 m 个物品，每个物品有一个权值。

有 n 个人，每个人也有一个物品，同样也有一个权值。这个人还有一个类型。

你会把你自己的前 K 个物品放到桌面上，然后这 n 个人顺序走进来。每次一个人会先把自己的物品放在桌面上，然后

- 如果他是类型一，那么他会拿走权值最大的物品。
- 如果他是类型二，那么他会拿走权值最小的物品。

这 n 个人都走完之后，你会获得剩下的 K 个物品。

对于每一个 K，求出最后拿到的权值之和。

$n \leq 2 \times 10^5$

有一个长度为 n 的序列 a ，初始全部为 0 。

你每次可以做两种操作之一：

- 选择一个 k 和一个长度为 k 的单调不降的序列 x ，对于 $1 \le i \le k$ 令 $a_i := a_i + x_i$ 。
- 选择一个 k 和一个长度为 k 的单调不降的序列 y ，对于 $1 \le i \le k$ 令 $a_{n-i+1} := a_{n-i+1} + y_i$ 。

给出长度为 n 的序列 A ，用最少的操作次数把 a 变成序列 A 。问最少次数是多少。

$n \le 2 \times 10^5$

SRM789 Hard

有 n 堆石子，第 i 堆石子有 a_i 个。

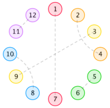
两个人轮流取石子，每次只能从一堆石子取，不能不取，不能行动者输。

与经典游戏不同的是，给出一个集合 S ，如果某个人某一次取的石子个数 x 使得 $x \in S$ ，那么另外一个人下一次也必须在同一堆石子行动。

$n \leq 10^5, a_i \leq 100$

有一个大小为 $2n$ 的环，你要把点分成若干组，每组两个点，两个点颜色相同当且仅当在同一组。

你需要满足同组的两个点要么距离为 n （即在环上是对面的），要么相邻，要么隔一个点。并且同组的两个点的对面的两个点也需要是同组的。



对于一个方案，定义权值为：从对面的点的位置切开，每一段的长度乘积。如这个方案的权值是 9 。

求权值之和对大质数取模。$n \le 50000$ 。

CF741C Arpa's overnight party and Mehrdad's silent entering

有 n 对情侣围成一圈坐在桌子边上，食物有两种，要求情侣不能吃同一种食物，并且桌子上相邻的三个人的食物必须有两个人是不同的。

构造一种可行分配方式。

$n \le 10^5$

你现在要整理书架上的 n 本书，每本书有一个颜色 a_i ，当每种颜色的书都摆在一起时书架上便整齐了，你每次可以将一本书放到序列最右端，问使书架上整齐的最小操作数。

$n \leq 5 \times 10^5$

CF1407E Egor in the Republic of Dagestan

给出 n 个点 m 条边的有向图，边有黑白两种颜色。

现在要给点染色，每个点染成黑或白色。白点只能走它连出去的白边，黑点只能走它连出去的黑边。

问是否存在一种染色方案，使得不存在 $1 \to n$ 的路径。若不存在这样的染色方案，最大化 $1 \to n$ 的最短路径长度。构造染色方案。

$n, m \le 5 \times 10^5$

CTS2022 D1T2

有一个 n 个点的树，每个点有一个点权 a_x 。

你每次可以选择一个点 x ，令 $a_x := -a_x + \sum_{(x,v)\in E} a_v$ ，然后把 a_v 清零。

你可以执行任意次操作。你需要使得最终只有至多一个点的点权非零，并最大化这个点权。

$n \le 10^5$

三个人玩 k 轮石头剪刀布。

一个“策略”：对于每个 $1 \le i \le k$，在第 i 轮出什么。

给出长度为 3^k 的数组 a, b，第一个人会以 a_i 的概率选择第 i 个策略，第二个人会以 b_i 的概率选择第 i 个策略。

对于每一个策略，求出第三个人如果选择这个策略，有多大的概率使得 k 轮中存在至少一轮使得他是这一轮的唯一赢家。

$k \le 12$

Unknown

有 n 个点，初始没有边。

给定 m 个集合，对于每个集合 S ，你都可以选择一组 $u, v \in S$ ，然后连边 (u, v) 。

最小化最终的图中树的个数。注意是树而不是连通块。

$n, m \le 10^5$