1001

根据 Dilworth 定理，一张有向图的最长链等于最小反链覆盖。故一个排列的 LIS $\leq k$ 可以看作可以用 k 个不交的不增子序列覆盖这个排列，于是目标变为删除代价和最小的子集使得可以用 k 个不交的不增子序列覆盖剩余元素。这等价于用 k 个不交的不增子序列做覆盖，使得被覆盖的元素的代价和最大。

费用流建模，用 (u,v,w,c) 代表连接 $u \to v$ 的流量为 w 费用为 c 的边。把 i 拆点成 $2i-1$ 和 $2i$，连边 $\forall 1 \leq i \leq n, (S, 2i-1, 1, 0), (2i, T, 1, 0), (2i-1, 2i, 1, -b_i)$，$\forall 1 \leq i < j \leq n$ 且满足 $a_i > a_j$，连边 $(2i, 2j-1, 1, 0)$。只需对 $1 \leq k \leq n$ 求出流量不超过 k 时的最小费用即可，标程直接使用原始对偶费用流配合 n^2 的 `dijsktra`（因为这是稠密图），复杂度是 $O(n^3)$ 的。常规的 SPFA 费用流理论上界是 $O(n^4)$，没有特意去卡（也不会卡），也可通过。

单测时间复杂度 $O(n^3)$。

1002

做过 WF2014 的那个 scenery 对做这题恐怕并没有什么帮助。
考虑一组可行解 $(p_1, q_1, id_1) \sim (p_n, q_n, id_n)$，其中 $1 \leq p_1, q_n \leq m$，
$\forall 1 \leq i \leq n, q_i - p_i + 1 = t_{id_i}, l_{id_i} \leq p_i \leq q_i \leq r_{id_i}$，$\forall 1 < i \leq n, p_i > q_{i-1}$，这代表最终的拍摄计划是第 id_i 处景色被第 i 个拍摄，且占据时间段 $[p_i, q_i]$。

考察一对 $(p_i, q_i, id_i), (p_{i+1}, q_{i+1}, id_{i+1})$，设 $a = id_i, b = id_{i+1}$，如果有 $q_{i+1} \leq r_1$ 且 $l_a > l_b$，我们对拍摄 a, b 这两处景色的顺序做调换，先将 l_a, l_b 均向 p_i 取 $\max$，这样仍然有 $l_a \geq l_b$。那么调换后的方案在 $l_b + t_b - 1$ 这个时间段结束对 b 的拍摄，在 $\max(l_b + t_b + t_a - 1, l_a + t_a - 1)$ 时刻结束对这两张照片的拍摄。而原来的方案 $q_{i+1} \geq \max(l_a + t_a + t_b - 1, l_b + t_b - 1)$ 不难发现其实有 $\max(l_b + t_b + t_a - 1, l_a + t_a - 1) \leq l_a + t_a + t_b - 1 \leq q_{i+1}$，所以这仍然得到可行解。

重复这样的调整直到无法调整时，只考虑 $q_i \leq r_1$ 的前缀，id_i 递减。

类似地，如果有 $p_i \geq l_1$ 且 $r_a < r_b$，也可以对拍摄 a, b 两处景色的顺序做调换从而得到可行解。

所以我们观察到一定存在一组可行解可以被划分为一段前缀和一段后缀，前缀的 id 递减而后缀的 id 递增。按照 i 从 n 到 1 的顺序考虑拍摄 i 的时间是在前缀中还是在后缀中，维护两个指针 x, y 初始 $x = 0, y = m + 1$。如果选择放在前缀 x 变为 $\max(x + 1, l_i) + t_i - 1$，放在后缀 y 变为 $\min(y - 1, r_i) - t_i + 1$，最后的目标是使得 $x \leq r_1, y \geq l_1$ 且 $x < y$。

DP，设 $dp_{i,j}$ 表示放完 $[i, n]$ 后 $x = j$ 的最大 y 即可，时间复杂度 $O(nm)$。

$nm \leq n^2 + m^2$，所以其实 $\sum nm \leq \sum n^2 + \sum m^2 \leq 2 \times 5000^2$，也是比较合理的。

其实这个 DP 恐怕可以用数据结构优化，不过题目还是简单一点好。

1003

设 $dp_{p,S}$ 为给 S 集合内的点填上 $[1, p]$ 内的数且满足题目所述限制，并且 $1 \sim p$ 每个值都至少用了一次的方案数，显然假设对 $\forall 1 \leq p \leq n, S \subseteq \{1, 2, 3...n\}$ 求出了 $dp_{p,S}$，一次询问的答案就是 $\sum_{i=1}^{\min(n,k)} \binom{k}{i} dp_{i,S}$，可以 $O(n)$ 计算。

$dp_{p,S} = \sum_{T \subseteq S, T \neq \varnothing} dp_{p-1, S-T} f(T, S)$，其中 $f(T, S) = 1$ 当且仅当不存在 $u \in S, v \in T$ 使得存在 $u \to v$ 的边。直接枚举子集转移 $O(n3^n)$，~~当然不能通过~~ 貌似有一些乱搞高手通过了。

注意到给定的图是一个 DAG，那么求出原图的一个拓扑序并进行重标号总能将所有边变为从编号较小的点指向编号较大的点。按照编号从小到大顺序考虑是否要加入 u，限制是加入 u 前点集内不能有指向 u 的边（即使再加入了一些编号大于 u 的点也不可能有点指向 u），时间复杂度降为 $O(2^n n^2)$。

单测的总时间复杂度为 $O(2^n n^2 + qn)$。

验题人给出了 $O(n(1+\sqrt{2})^n + 2^n n^2 + qn)$ 的做法，大体思路是假设 S 确定，T 其实如果是 S 中无入度点构成的集合的子集才合法，那么这是一个类似 这题 的问题，不过现在保证问号的数量加上 1 的数量不超过 $|S|$，所以总可以 $O(2^{\frac{|S|}{2}})$ 求出和。这个做法的优势在于不需要保证整张图是 DAG。

1004

记 $f(i,j)$ 为 $p=i, q=j$ 时最多能摧毁多少个敌军单位，可以发现 $f(i,j)$ 很容易 $O(nm)$ 计算：枚举每个 p 行 q 列的矩形区域，如果均为 0 将这些位置均标记有贡献，最后数一下有贡献的位置是否 $\geq k$ 即可。

显然 $f(i,j)$ 在每一维上均具有单调性：$\forall 1 < i \leq n, 1 \leq j \leq m, f(i,j) \leq f(i-1,j)$，$\forall 1 \leq i \leq n, 1 < j \leq m, f(i,j) \leq f(i,j-1)$，这是因为 $i > 1$ 时轰炸一个 $i \times j$ 的矩形可以转化为轰炸两个 $(i-1) \times j$ 的矩形，列也同理。从低到高枚举 i，$f(i,j) \geq k$ 的 j 总是一段前缀，且在 i 增加时前缀长度总只缩减，于是只需要真的计算 $n+m$ 个 $f(i,j)$ 即可求出原问题的答案，这样复杂度为 $O((n+m)nm)$，希望这个做法即使除 w 也无法通过。

考虑加速计算 $f(i,j)$，观察：任意一个 $i \times j$ 的矩形区域总恰好经过一个编号是 i 倍数的行，称这些行为关键行。将所有行划分为 $\left\lceil \frac{n}{i} \right\rceil$ 组，每一组内的格子如果被可行的矩形区域覆盖，要么该矩形区域穿过该组内的关键行，要么穿过上一组的关键行。考虑一个关键行上的格子 (x,y)，对于所有左边界是 y 且穿过这个格子的矩形区域，设其覆盖行编号 $[l,r]$，可以发现 l,r 各自有上下界：如果对每个格子计算出其向上最多可以延申出多长的全 0 区域记为 $u_{x,y}$，那么 $l \geq x - \min_{k=y}^{y+j-1} u_{x,k} + 1$，下界同理。当 $r-l+1 \geq p$ 时，穿过 (x,y) 左边界为 y 的 $i \times j$ 矩形区域的并实际上覆盖了 $l \sim r$ 行的 $y \sim y+j-1$ 这些列；而 $r-l+1 < p$ 时则没有任何合法矩形区域。

用单调队列可以 $O(m)$ 算出一行关键行的所有格子的上下界。得到上下界以后，取出一组，每个矩形区域一定 cover 一组内一段前缀/后缀的行中的一段长度为 j 的列区间，再用单调队列就可以 $O(m)$ 计算每列被覆盖的前/后缀长度。

于是在 $O(nm)$ 预处理后可以 $O(\left\lceil \frac{n}{i} \right\rceil m)$ 计算 $f(i,j)$，显然存在对称算法可以 $O(n\left\lceil \frac{m}{j} \right\rceil)$ 计算 $f(i,j)$。

直接对原来的通过求 $n+m$ 个 $f(i,j)$ 求出原问题答案的算法施加这个优化（使用理论上更快的算法），可以证明时间复杂度为 $O(nm\log(nm))$。

证明：算法流程可以描述为，初始令 $j=m$，增量枚举 i，只要 $f(i,j) \geq k$ 不断减少 j。考虑这样计算时间复杂度：如果 $f(i,j) \geq k$，下一步会让 i 增加，认为这一步消耗了 $O(\left\lceil \frac{n}{i} \right\rceil m)$ 时间；如果 $f(i,j) < k$，j 会减少 1，认为这一步消耗了 $O(n\left\lceil \frac{m}{j} \right\rceil)$ 时间。那么由调和级数容易发现复杂度不超过 $O(nm\log(nm))$，而这样的估计只会偏大。

时间复杂度单测 $O(nm\log(nm))$。

1005

先用叉积求出每个凸包的面积，显然套娃只能让面积大的凸包在外侧，按面积排序后转化为 $n-1$ 次判定问题。顺序枚举大凸包的每一条边，用指针维护这条边切到小凸包的点，就可以得到一个小凸包位移的半平面限制。得到大凸包点数个限制后求半平面交。由于解集为开集，所以只要求出来的凸包（或凸壳）的面积为正则有解。复杂度 $O(n\log n + m)$ 或 $O(n\log n + m\log m)$。

注意本题值域很大而且出题人故意卡了精度，在求直线交点时如果使用浮点数，即使开 long double 应该精度还是不够（半平面交需要 v^4 级别的精度，如果有更高精度的浮点数可能可以保证正确性），建议使用全整数（需要 `int128`）。

1006

设 g_i 表示所有选手做题并集恰好为一个已经确定的大小为 i 的集合的方案数，显然答案为 $\frac{\sum_i \binom{m}{i}\frac{g_i m}{m-i}}{\prod_i \binom{m}{a_i}}$

。

考虑容斥，设 f_i 表示钦定所有选手做题的并被一个已经确定的大小为 i 的集合包含的方案数，那么由二项式反演显然有 $g_i=\sum_{j=0}^{i}\binom{i}{j}(-1)^{i-j}f_j$。故求出 f 后可以一次卷积得到 g。

根据定义，$f_i=\prod_j\binom{i}{a_j}=\frac{i!^n}{\prod_j a_j!(i-a_j)!}$。其中大部分值都是好算的，只有 $\prod_j(i-a_j)!$ 比较难算，考虑如何快速算这个。

记其为 h_i，不难发现 $\frac{h_i}{h_{i-1}}=\prod_j(i-a_j)$，也就是说对于每个 i 我们能算出右边这堆东西就能递推出 h_i 。而右边是一个度为 m 的多项式，不妨考虑使用分治乘求出该多项式，然后使用多项式多点求值，把所有 i 带进去就能算出右边的所有值了。

至此，原问题得解，设 n,m 同阶，总时间复杂度 $O(n\log^2 n)$，常数比较大，所以时限给的比较宽松。

此外，本题还存在基于离散对数和任意模数卷积的单 $\log$ 做法，此处不详细介绍。

1007

首先特判两人在两个连通块里的情况，是简单的，除了孤立点的情况外，必然可以一直沿着一条边反复横跳无限走。接下来默认两人在同一连通块。

不难发现，如果一个人走到了环上，那么他就立于不败之地。同样不难发现，如果一根人走到了连接两个环的一根链上，他也不会输，因为其两边必然只能有最多一边可能被对手阻挡（不然就成环了），那么他一定能走到一个环上。我们称这两类点为好点。

所以我们对于两人分别求出，自己能否阻挡对方进入某个好点。

首先先判断掉该连通块是树的情况，此时不存在好点，胜负只与初始 Alice 和 Bob 的距离的奇偶性有关。举个例子：假设初始时两人距离为偶数，那么轮到 Alice 时，二人距离奇偶性仍是偶数，而又不可能为 0，因此距离至少为 2，故 Alice 一定可以行动，立于不败之地，把 Bob 逼入叶子取得胜利。

否则，我们先求出好点，可以一直剥一度点，剩下的都是好点。还能顺便求出每个点到最近好点的距离，稍后会用到。

对于一个人，考虑为了阻挡对手进入好点，他必须尽快赶到离对手最近的好点。同时，与树的情况类似的，还需要保持距离的奇偶性正确，才能把对手逼到某个一度点，不然他会被迫后退，让对手走到好点。所以相当于要求他的初始位置到离对手最近的好点，且奇偶性给定的最短路长度，并于对手要走的距离进行比较。由于起点或终点固定，故可以分层图 bfs 解决，总复杂度线性。

1008

显然，有如下的两种特殊情况：

- $k=1$，此时每人只能操作一个，那么显然 n 为奇数 Alice 必胜，n 为偶数 Bob 必胜；
- $k=n$，此时 Alice 一次可以全部操作，那么 Alice 必胜。

除此之外，Alice 无论第一步如何操作，Bob 都有一种方式，使剩下未操作的分成两个一样长的连续段（长度可以为 0），根据奇偶性取最中间的 1 或 2 个即可。此后无论 Alice 怎么操作，Bob 直接在另一段的进行对称的操作，那么不能操作的只能是 Alice，故 Bob 必胜。

1009

先判一下 A 在 B 中出现多次，以及 B 在 A 中出现多次的情况，此时一定无解。

否则如果一个串包含了另一个串，那么答案显然就是较长的串。

否则互不包含，我们可以考虑找到一个 A 的最长前缀，使得其是 B 的一个后缀，然后我们把 B 去掉这个后缀后和 A 拼接得到一个字符串，我们可以发现这个串满足 A 和 B 各出现恰好一次。如果有串出现了多次，可以发现存在一个更长的符合条件的前缀，矛盾。

同理我们交换 A, B 也能得到一组解。容易说明这两组解中长度较短的就是答案。

实现的时候可以 KMP 或者哈希（请不要使用自然溢出）。时间复杂度 $O(|A|+|B|)$。

1010

题意是给定 a_i, b_i 构造 c_i 满足 $(a_i \text{ xor } c_{i-1}) + (b_i \text{ xor } c_{i-1}) = c_i$，特殊记 $c_0 = c_q$。

考虑确定 c_i 二进制表示下的最低位，可以发现无论 c_{i-1} 的最低位是多少，上述式子的左边部分的最低位确定，因此可以确定 c_i 的最低位。

同理地，确定最低位后我们将所有最低位删除并递归即可。这样可以依次确定 c 的每一位。

这个过程在最高位确定后停止，容易发现答案的个数唯一。因此直接输出即可。

1011

最极端的情况是“你”每个比赛都获得 0 分，排名在“你”之前的选手获得 1 分，排名在“你”之后的选手获得 0 分。这样可以得到一个选手如果最终排名在“你”之前，当且仅当其存在一场比赛排名在”你“之前。

所以我们直接计算每场比赛在”你“之前的选手人数总和，由于重复的人无效所以需要对 $m-1$ 取 $\min$，最后再和最终排名比较一下即可。

另一种等价的理解方式是如果 $k=m$ 那么无论怎么样都能金牌，否则直接把所有排名 -1 相加和 $k-1$ 比较。

1012

对所有 $p_i \neq 0$ 的 i，$i \to p_i$ 连边，可以发现图可以被分为一些环和恰好 k 条路径，k 条路径每条恰好有一个点未被指向，一个点没有出度（这是这条路径中唯一的 $p_i = 0$ 的点）。

因为只关心最终图的环结构，一种方案可以视作把 k 条路径组成的集合做集合划分为 $S_1, S_2 \dots S_m$，每个集合内的路径拼起来拼成新图一个环，易知这样对应 $\prod_{i=1}^{m}(|S_i|-1)!$ 种能填出来的排列。将 k 条路径的一个子集拼起来成环后得到的颜色数容易用 `bitset` $O(\frac{n2^k}{w})$ 计算，直接枚举 $Bell(k)$ 种集合划分计算贡献复杂度 $O(Bell(k)poly(k))$，显然无法通过。

重述一下问题：给出 $\{1,2,3...k\}$ 所有非空子集对应的权值 a（也就是颜色数），把集合 $\{1,2,3...k\}$ 进行集合划分，对每种 i 求出有 cnt_i 个子集的 a 为 i，贡献是 $\sum_{i=1}^{n} val_{i,cnt_i}$（$val$ 可以看作就是任意的 $n \times (k+1)$ 的矩阵，因为要考虑已有的颜色数为 i 的环的影响）乘上一个系数是所有划分出的子集大小减 1 的阶乘之积。对所有划分方案贡献求和。

现在枚举一个 i，考虑所有 a 为 i 部分的贡献，$a_0 \sim a_{2^k-1}$ 中会有一些 $a_j = i$，把这些 j 记作集合 P。

进行 DP，设 $f_{S,m}$ 为选一些 $x_1, x_2 \dots x_m \in P$ 使得 x_i 构成一组 S 的划分的所有方案的系数和（子集大小减 1 的阶乘之积）。

每次从 P 中取出一个元素 x_i 更新 f，注意到这个 x_i 能进行的转移是：枚举所有 x_i 的超集 S，将 $f_{S,m}$ 加上 $f_{S-x_i,m-1}(|x_i|-1)!$，所以这个转移容易做到总复杂度（对于所有 i 的代价相加） $O(3^k k)$。

记 ans'_m 为所有 $f_{S,m}(k-m)!$ 之和，可以发现 ans'_m 其实相当于对所有可能的子集划分方式，如果其实际上有 s 个子集的 a 为 i，那么会被计算 $\binom{s}{m}$ 次，再二项式反演一下即可求出恰好有 m 个子集的 a 为 i 的方案数 ans_m。

时间复杂度 $O(\frac{n2^k}{w}+nk^2+3^kk+n)$。