水题选讲

彭博

北京大学

2023.10

机器人有一个 1 到 n 的概率分布 p ，它从 p 里随机抽取了一个数字 x 。

你需要猜一个数字 y 。如果猜对了，游戏结束。否则，你的记忆会被清除（即你不知道你之前猜了什么），而机器人会以 c 的概率重新抽取一个数字，以 $1-c$ 的概率维持原数字不变。

求出你在最优策略下期望需要猜几次才能结束游戏。不取模。

$n \le 10^5$

显然我们需要选定一个概率分布 q ，每次就从 q 里面抽一个数字出来。

考虑如何计算一个 q 的答案。直接列无限求和式可能是可以做的，但更优雅的做法是设未知数解方程。

设 f_i 表示机器人当前的数字是 i 时期望需要多少步，那么可以列出方程

$$f_i = 1 + (1 - q_i)((1 - c)f_i + c \cdot S)$$

其中 $S = \sum_i p_i f_i$ 。移项得到 $f_i = \frac{1+(1-q_i)cS}{1-(1-q_i)(1-c)}$ 。

把 f_i 线性组合一下即可得到关于 S 的方程

$$S = 1 + \sum_i p_i \frac{1 + (1 - q_i)cS}{1 - (1 - q_i)(1 - c)}$$

可以继续移项得到 S 的封闭形式，但封闭形式关于 q 过于复杂，不太能做。

这就要用到另一个非常重要的技巧：二分答案。

二分答案 ans ，把等号右边的 S 换成 ans 。如果可以把它优化到 $\leq ans$ 那就合法，否则不合法。

虽然右边的式子是依托答辩，但至少它关于每个 q_i 都是 $\frac{a+bx}{c+dx}$ 这样比较清楚的形式。

简单分析系数的正负性，不难发现它关于每个 q_i 都是凸的。而 q_i 需要满足的性质只有非负和相加等于 1 。

于是可以二分斜率，得到每个 q_i 的取值，看它们相加是否 >1 。但两个 log 有点吃不消。

推一下式子可以得到斜率的封闭形式，但有些 q_i 可能会取到负数，这样也不行。

因此最终应该是按 p_i 排序，枚举前几个 q_i 为零，用封闭形式得到斜率，然后返回答案。

1st ucup stage 7 L Directed Vertex Cacti

给定 n 和 m 。

求出有多少个有向图，满足每个点至多只在一个环内，且有恰好 m 条不在任何一个环内的边。

$n, m \le 10^6$

1st ucup stage 7 L Directed Vertex Cacti

首先你不能看错题，其次你不能看错题，最后你不能看错题。

读完题的第一反应可能是数 m 条边的 DAG ，但这并不好数。

一个经典的错误算法是，对于 $n!$ 个拓扑序，每个拓扑序都有 $\binom{\binom{n}{2}}{m}$ 种选边的方法。

其问题在于，一个 DAG 可能有多个拓扑序。

但这题有趣的地方就在于，算重的次数与给这个 DAG 加环的方案数有某种神秘的联系。

可以归纳证明，给 m 条无向边定向并加环的方案数恰好是 $n!$ 。

每次多加一条边 (u, v) ，考虑每种定向和加环的方案。

- 如果 u, v 属于不同的可比较的环，就只有一种定向的方案。
- 如果 u, v 属于不同的不可比较的环，从较小值连向较大值。
- 如果 u, v 属于同一个环，把这个环从 u, v 的位置拆成两个，然后从较大值连向较小值。

可以发现新图与原图一一对应。

在 n 轮剪刀石头布的游戏里，Alice 分别出了 A_S, A_R, A_P 次剪刀、石头、布，Bob 出了 B 次，Carol 出了 C 次。

已知 n 轮游戏每一轮都是平局，求出有多少种本质不同的游戏，模 998244353 。

$n \le 1.5 \times 10^6$

平局共有 9 种，限制也恰好有 9 个，看起来很有希望！

可惜了，把 9×9 的矩阵列出来之后发现它的秩只有 7，也就是说要枚举两个变量才能解出每一个变量。

但是这题的关键就在于，看出缺失的两个自由度是出在哪里。

$$SSS + RRR + PPP = SRP + RPS + PSR = PRS + RSP + SPR$$

这三组之间可以任意转换，这就是那两个自由度的来源。

这就是一个 $x + y + z = constant$ 的限制。把 $x + y$ 用卷积卷起来，乘上 z 的贡献即可。

两个人在一个 n 个点 m 条边的有向图上走路。图上有 K 个关键点 $p_1, \cdots, p_K$ ，在 p_i 上有价值为 w_i 的宝石。

两个人轮流行动，每次沿着一条边走一步（不能不走），并取走走到的点的宝石。当所有宝石都被取走或游戏陷入循环时结束。

Alice 想要最大化 Alice 拿到的价值和减去 Bob 拿到的价值和，而 Bob 想最小化这个值。求出游戏结果。

$n \leq 30, K \leq 10$

有环的博弈题。

如果只有必胜、必败和平局三种情况，可以倒推：能走到必败态的是必胜态，只能走到必胜态的是必败态，剩下推不动的全都是平局。

带权值的其实也一样。设状态的转移是 $dp_x = \max_v - dp_v$ ，那么对每个强连通块分别倒推：出边全部确定的点也可以被确定，而如果不存在这样的点，就把当前权值已经最大的点拿出来，如果它权值为正就把它定下来，否则剩下的点全都定为 0 。

难点在于怎么写的好看。

给定 n 个串 $s_1, \cdots, s_n$ 和一个目标串 t 。

初始你有一个空串，一次操作可以往后面加上某个串 s_i 或删去最后 K 个字符，问得到串 t 需要多少次操作。

$\sum |s_i| \le 10^6, \sum t_i \le 5000, K \le 5000, Q \le 100$

直接设 dp_i 表示搞定了 t 的前 i 位需要多少步。

转移就是加上某个串 s_i ，然后一通加加减减，剩下 s_i 的某个前缀。

现在有两个问题：一是串的个数太多，枚举合法 s_i 复杂度会起飞；二是要求出对每个 l ，一通加加减减把最后 l 个字符消掉所需的最少次数。

一比较好处理，只需要把所有 s_i 建出字典树，那么字典树上每个节点显然只需要做一次就可以把答案记忆化下来。一个很深刻的结论是，字典树上，所有节点的子树叶子个数和是 $O(\sum |s_i|)$ 的，因此只需要暴力枚举子树的叶子即可。

二需要跑一个模 K 的同余最短路，但这题并没有保证 $\sum K$ 的大小，因此每组询问都跑 $O(K^2)$ 的暴力会飞。这比较麻烦。

1st ucup stage 20 E Strange Keyboard

注意到 $\sum |s_i| \le 10^6$，因此长度不同的 s_i 的个数只有约 1500 个，也就是说跑同余最短路时只有 1500 种边。

同余最短路也有很深刻的性质：我们只关心边的集合而不关心走的顺序！

按顺序加入每一种边，一种边会把 K 个点划分成若干个环，每个环显然可以线性更新答案，于是就做完了。

给定一棵以 1 为根的有根树，每个点都只能给自己的祖先投票，不能不投，也不能投给自己。

求出哪些点可以成为得票数严格最多的点。

$n \leq 10^6$

判断一个点 x 是否合法的简单的：x 子树里的点显然都应该投给 x，然后把 x 子树删掉，对剩下的点贪心把票从下往上推即可。

为了进一步优化，需要注意到：判断不同的 x 时，如果他们的子树大小相同，那么判断的过程是几乎相同的。

比如，如果 x 和 y 的子树大小相同，那么判断 x 的时候 y 的子树至多只会多出一票往上推（相比于判断 y 的时候），几乎等于没有。

进一步观察，如果 x 的子树大小比 y 大，那么判断 x 时 y 子树就和不存在一样。

受这种现象启发，我们直接二分答案 w ，看是否存在一种方案使得每个点的得票数都不超过 w 。这只需要一遍贪心。

不难发现子树大小 $> w$ 的点一定合法，而 $< w$ 的点一定不合法，就只剩下子树大小恰好 $= w$ 的点需要判断了。

1st ucup stage 20 H Treelection

设 x 的子树大小为 w 。判断能否每个点的子树大小都 $\leq w-1$ 时，x 可能会往上送两票；但判断 x 能否成为严格最多时，x 只会往上送恰好一票，这之间的差距决定了 x 能否成为答案。

因此只需要判断，是否 x 子树减少一票，会使得 x 到根的路径上每个点都少往上推一票，从而最终使得根的票数从 $w-1$ 降为 $w-2$ 。这只需要看 x 到根的最小值即可。

1st ucup stage 20 J Talk That Talk

给定质数 p ，定义一个长度为 $p-1$ 的 01 序列 $s_1,\cdots,s_{p-1}$ ，其中 $s_i=1$ 当且仅当 i 是模 p 的二次剩余。

给定正整数 t ，求出有多少对 $i<j<k$ 使得 $j-i=k-j\le t$ 且 $s_i=s_j=s_k$ 。

$p\le 10^{12}, t\le 10^6$

根据二次剩余众所周知的结论，可以设 $s_i = i^{(p-1)/2}$ 。

从 01 改成 ±1 之后，三个数相等的限制也可以被拆为 $(s_i s_j + s_j s_k + s_i s_k + 1)/2$ 。

问题只剩下求出 $\sum_{l=1}^{t} \sum_i i^{(p-1)/2}(i+l)^{(p-1)/2}$ ，以及把跨过 0 的部分减掉。

对于 $0 \le r < p-1$ ，有 $\sum_{i=0}^{p-1} i^r = 0 \pmod p$ ，因此整体求和非常好做。

跨过 0 的部分只有 t 个位置，随便算一下即可。

1st ucup stage 22 A Moniphant Sleep

有两个数字 a 和 c ，初始 $a = 0, c = \infty$ 。

有 4 种操作:

1. 令 $a := a + 1$ 。
2. 令 $a := a - 1$ 。如果此时 $c > a$ ，令 $c := \infty$ 。
3. 令 $c := \min(c, a)$ 。
4. 如果 $c \leq a$ ，令 $a := c$ ，然后令 $c := \infty$ 。

你需要维护 n 组数字 $a_1, \cdots, a_n, c_1, \cdots, c_n$ ，每次操作都作用在一个区间上，并支持单点查询 a_i 。

$n, Q \leq 5 \times 10^5$

令 $d=a-c$ 。特别地，如果 $c=\infty$ ，令 $d=\infty$ 。不难发现 a,c 都不是很重要，只需要维护当前的 d 以及之前的 Δa 之和。

相应地可以把四种操作转化为对 d 的操作，以及返回一个 Δa 。

发现一个性质：在出现操作 4 之前，Δa 与 d 无关；而出现操作 4 之后，新的 d 与最开始的 d 无关。

于是可以维护分段函数 $f(d)=(d',\Delta a)$ ，表示一开始的 d 经过一大堆操作之后会变成 d' ，并且在过程中给 a 加上了 Δa 。

根据性质，d' 和 Δa 必有一个是常数，而不难发现另一个不是常数的函数的形式也比较简单，因此这东西是可以维护的。

2nd ucup stage 2 M Hardcore String Counting

给定长度为 n 的字符串 s ，求出有多少个长度为 m 的字符串 t ，使得 s 在 t 的第一次出现位置就在最后。

$n \leq 10^5, m \leq 10^9$

这题本身没啥意思，但要记得这东西是能做的。

设 dp_m 表示长度为 m 的答案。直接用 26^{m-n} 作为初值，然后要把之前就出现过的情况给减掉。

如果第一次出现与最后一次无交，其贡献是 $\sum_{i=n}^{m-n} dp_i 26^{m-n-i}$ 。

否则，就是枚举 border ，贡献为 $\sum_{i \in border} dp_{m-i}$ 。

虽然递推式有个前缀和，但相邻的递推式相减就可以消掉，剩下就是一个平平无奇的线性递推。

2nd ucup stage 2 C Many Many Cycles

给定一个 n 个点 m 条边的带权无向图，求出其所有简单环的权值的 gcd 。

$n \leq 5000, m \leq 10000$

先说正解。

随便拿一棵生成树出来，随便选几条非树边，它们异或起来就可以得到若干个简单环的和。

但事实上，我们只需要考虑一条和两条的情况。这是因为多条非树边的交一定可以表示为两条非树边的交。

因此只要考虑了一条和两条，就可以线性组合出任意非树边的交（的两倍），自然也就可以组合出任意非树边的 xor 。

复杂度 $O(m^2)$ 。

2nd ucup stage 2 C Many Many Cycles

然后说优化。

给每个非树边一个随机权值，把每个树边的权值定为覆盖它的非树边的权值的异或和，那么权值相同的边只能同时出现在环上，或同时不出现。

显然每一组边的权值和的 gcd 一定是答案的约数。可以证明，答案只能是它或它的两倍。

证明其实并不难，用一条边和两条边的线性组合乱搞搞，就可以容斥出每一类边的权值和的两倍。

只需要判断是不是两倍即可。

2nd ucup stage 2 C Many Many Cycles

反正每一类边只能同时出现或同时不出现，不妨把一类边的权值和集中到一条边上，再除掉 gcd 。

现在就只需要判断是不是每个环的权值和都为偶数了。

拿并查集乱搞搞即可。

2nd ucup stage 3 H Hurricane

给定一个 n 个点 m 条边的无向图 G ，设其补图为 G' 。

对于 $k = 1, 2, \cdots, n-1$ ，求出有多少对点在 G' 的最短路恰好为 k 。

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

2nd ucup stage 3 H Hurricane

显然只有 m 对点的最短路不是 1 。

更进一步，如果两个点在原图的度数和小于 n ，那么它们的最短路至多是 2 。

否则，必有一个点的度数至少是 $n/2$ ，而这样的点只有 m/n 个。

对每个这样的点用 $O(n+m)$ 的时间把单源最短路跑出来即可。

对两个长度相等的序列 a, b ，设 $a \cdot b = \max_i a_i + b_i$ 。

给定长度为 n 的序列 a 和长度为 $n+m$ 的序列 b ，求出 a 与 b 的每个长度为 n 的子串的点积。从左往右输出，强制在线。

保证 a, b 在值域范围内均匀随机。

$n, m \leq 10^6, 1 \leq a_i, b_i \leq 10^9$

2nd ucup stage 3 K Sequence Shift

设 $W = 10^9$ 。

因为数据随机，毛估估一下，答案应该挺接近 $2W$ ，而这只能在两个数都很大的时候取到。

设定一个阈值 T 。只要答案大于 $2W - T$ ，两个数就必须都大于 $W - T$ ，而这种数只占所有数的约 T/W 。

然而根据生日悖论，只要 $T/W = \Omega(1/\sqrt{n})$ ，就有很大概率存在一个位置的两个数都大于 $W - T/2$ ，从而使得答案大于 $2W - T$ 。

因此，合理选择阈值 T ，就可以只保留 $O(\sqrt{n})$ 个数，它们只会产生 $O(n)$ 对匹配，暴力做即可。

运气不好的子串可能一个匹配都没有，但反正这样的子串很少，出现这种情况时 $O(n)$ 暴力扫一遍就行。