省选 2024 模拟赛 Day 2　题解

2024 年 2 月

目录

A 游戏（game）

操作实际上是对于从前往后数第 i 个数若为 1，则可以删除从后往前数第 i 个数。只用最靠边的 1 来删是最优的，因为 1 边上的 0 一定不会被删掉。

所以答案为

$$n - 2 \times \min_{a_i=1}(i-1, n-i)$$

所以不管怎么删，一定都会走这些步数。

询问是个 Nim 游戏，需要快速求出每个区间的答案。需要支持区间取反，区间求最靠边缘的 1。

最靠边缘的 1 可以拆分成第一个 1 和最后一个 1，这样就可以直接用线段树维护区间第一个出现的 $0,1$ 以及区间最后一个出现的 $0,1$ 即可解决问题。

B 摸牌（cards）

考虑枚举所有状态，计总数为 cnt，枚举每种转移。此时我们可以得到一个 cnt 元一次方程组。通过高斯消元解得每种状态的答案。总复杂度 $\mathcal{O}(cnt^3)$，可以获得 12 分。

考虑归类同样的状态（如 $(0,1,2)$ 和 $(2,1,0)$ 本质相同），此时我们发现本质不同状态数不超过 $p(m)$，其中 $p(i)$ 为拆分数。暴力枚举所有状态列出方程，同样可以使用高斯消元的方式解得每种状态的答案。总复杂度 $\mathcal{O}(p^3(m))$，可以获得 33 分。

$n=2$ 时状态有 m 种。容易发现每种状态只依赖于另外两种状态，于是递推消元即可。可以获得 17 分。

考虑给每个状态一个势能。钦定每一轮后前往转移状态的势能恰好增加 1。不难发现状态的势能依赖于每种数字卡片的数量。令 $f(i)$ 为某种数字卡片数量为 i 时的势能，一个状态的总势能就是 $\sum\limits_{i=1}^{n} f(a_i)$。可以列得递推式:

$$\sum_{i=1}^{n} f(a_i)+1=\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{a_i(m-a_j)}{(n-1)m^2}g(i,j)$$

其中：

- $i=j$ 时，$g(i,j)=\sum\limits_{k=1}^{n} f(a_k)$；
- $i\neq j$ 时，$g(i,j)=\sum\limits_{k=1,k\neq i,j}^{n} f(a_k)+f(a_i-1)+f(a_j+1)$ 。

适当进行展开可得：

$$\begin{aligned}\sum_{i=1}^{n} f(a_i)+1=&\sum_{i=1}^{n}(\frac{m-a_i}{m}\times\frac{m(n-2)+a_i}{m(n-1)}f(a_i))\\&+\sum_{i=1}^{n}(\frac{a_i}{m}\times\frac{m-a_i}{m(n-1)}f(a_i))\\&+\sum_{i=1}^{n}(\frac{m-a_i}{m}\times\frac{m-a_i}{m(n-1)}f(a_i+1))\\&+\sum_{i=1}^{n}(\frac{a_i}{m}\times\frac{m(n-2)+a_i}{m(n-1)}f(a_i-1))\end{aligned}$$

可以带入 $m+1$ 组 $(a_1,a_2,\ldots,a_n)$ 并高斯消元。

考虑继续优化，上式已经将 $f(a_i)$ 变得仅与 $f(a_i-1),f(a_i),f(a_i+1)$ 有关了。将 $f(a_i)$

分离出来，则对于所有 x 满足：

$$\begin{aligned}f(x)+\frac{x}{m}=&\frac{m-x}{m}\times\frac{m(n-2)+x}{m(n-1)}f(x)\\&+\frac{x}{m}\times\frac{m-x}{m(n-1)}f(x)\\&+\frac{m-x}{m}\times\frac{m-x}{m(n-1)}f(x+1)\\&+\frac{x}{m}\times\frac{m(n-2)+x}{m(n-1)}f(x-1)\end{aligned}$$

对原式进行整理：

$$(m^2+x^2-2mx)f(x+1)+(4xm-xmn-m^2-2x^2)f(x)+(xmn-2mx+x^2)f(x-1)+xm-xmn=0$$

令 $g(x)=f(x+1)-f(x)$：

$$(m^2+x^2-2mx)g(x)=(xmn+x^2-2mx)g(x-1)+xmn-xm$$

也就是:

$$g(x)=\frac{(xmn+x^2-2mx)g(x-1)+2mn-xm}{(m^2+x^2-2mx)}$$

将 $x=0$ 带入得：$g(0)=0$。

于是，将 $g(x)$ 递推，再做一遍前缀和即可得到 $f(x)$。

对于每次询问，初态的势能为 $\sum\limits_{i=1}^{n}f(a_i)$，终态是 $f(m)$（假定 $f(0)=0$ 即可得到），相减即为期望。总复杂度 $\mathcal{O}(m+nt)$ 。

C 轻涟（vaguelette）

1. 第一部分

首先思考 $F(T,a)$ 怎么求。如果一个方案中两条路径 $a \to b, c \to d$ 相交了，肯定是不优的。把两条路径异或起来，可以得到 $a \to c, b \to d$ 的不相交路径，这样做更优。按照这样调整，最优方案中所有路径都是不交的。换句话说，一条边至多被经过一次。

这提示我们对每条边分别考虑。对于边 $x \to fa_x$，把它断掉后会形成两棵子树，设两棵子树在 a 中出现次数分别为 k 和 $m-k$。如果 k 为偶数，那两棵子树之间可以内部匹配，这条边就不会被经过。否则必然有一条路径经过它。

这样我们就得到了一条边产生贡献的条件：它的子树结点在 a 中的出现次数为奇数。

继续深入探讨。在 $G(T,a)$ 中，假设 $siz_x = s$，求 $x \to fa_x$ 的贡献次数。不妨枚举 a 中恰好有 i 个数属于 x 的子树，这样的方案数就是 $\binom{m}{i}s^i(n-s)^{m-i}$。先除去 $i=0$ 和 $i=m$ 的情况，然后考虑有多少子序列符合条件。对于 s 中的部分，它要选出奇数个数，根据对称性，答案是 2^{i-1}。对于 s 外的部分，要保证子序列长度为偶数，根据对称性答案是 2^{m-i-1}。综上，我们可以列出式子:

$$\begin{aligned} w(n,m,s) &= \sum_{i=1}^{m-1}\binom{m}{i}s^i(n-s)^{m-i}2^{i-1}2^{m-i-1} \\ &= 2^{m-2}(n^m - s^m - (n-s)^m) \end{aligned}$$

我们惊讶地发现，一条边的贡献次数只跟它子树的大小和总共的结点个数有关。

分子树内和子树外两部分考虑。设 $f_{i,j}$ 表示以点 i 为根，仅考虑 i 的子树，连通块大小为 j 的方案数；$g_{i,j}$ 表示以点 i 为根，不考虑 i 的子树，连通块大小为 j 的方案数。枚举每条边 $i \to fa_i$，只需要将 f_i 和 g_i 拼起来就可以得到答案。朴素实现是 $\mathcal{O}(n^3)$ 的。

使用任意模数 NTT 可以做到 $\mathcal{O}(n^2\log n)$ 。

考虑优化。注意到 f_i, g_i 和它们的转移相当于多项式乘法，且它们的次数都 $\le n$，于是只需要维护 $0 \sim n$ 这 $n+1$ 个点值就可以得到这个多项式。同时，单次乘法就变成了点值对应相乘，变成了 $\mathcal{O}(n)$。注意到总的乘法次数是 $O(n)$ 级别的，所以这一部分的时间复杂度为 $\mathcal{O}(n^2)$。

问题在于我们还要去还原多项式再计算贡献，造成了时间复杂度瓶颈。这个问题也是好解决的，只需要把每个点的多项式乘上对应的边权后再加起来，最后统一还原，就可以做到 $\mathcal{O}(n^2)$。

2. 第二部分

第二问也是类似的，考虑每个连通块的贡献。枚举需要算贡献的边 $x \to fa_x$，按点的编号从小到大进行 dp，设 $f_{i,j,k}$ 表示考虑到点 i，上面的连通块大小为 j，下面的连通

块大小为 k 的方案数。

对于 $i < x$ 的部分，它只能连向上面的连通块或者不在连通块内，方案数为 j 或 $i-1$。对于 x 时，需要连接上面的部分，方案数是 j。当 $i > x$ 时，可以选择不连，连上面或者连下面，方案数分别是 $i-1/j/k$。朴素实现是 $\mathcal{O}(k^4)$ 的。

注意到我们的转移跟 x 并没有太大关系。考虑优化状态，设 $f_{i,j,k,0/1}$ 表示考虑到点 i，上面为 j，下面为 k，是否确定 x 的方案数。注意确定 x 时需要乘上 val_x。总时间复杂度 $\mathcal{O}(k^3)$。