赛拟模选省合联

题解

时间： 2022 **年** 3 **月** 12 **日**

你（you）

容易发现题目即是要对由 n 个数构成的合法线性基计数

对于 $X=0$ 的情况，我们可以轻松算出答案

对于 $X\neq 0$ 的情况，注意到对相同的 n,k, 不同的 X 答案相同。所以可以不考虑不能异或出 X 的限制求答案，然后乘上一个简单的贡献即可。

设 $f_{i,j}$ 为考虑前 i 个数，线性基的大小为 j 的合法方案数

考虑每一次在末尾增加一个数，

如果线性基大小不增加，则有 2^j-1 的贡献，

如果增加则有 2^j-1 个不能选，然后有 2^j 个会导致表示出 X，则能贡献 2^k-2^{j+1}

有：

$$f_{i,j}=(2^j-1)f_{i-1,j}+(2^k-2^j)f_{i-1,j-1}$$

记 $F_j=\sum f_{i,j}x^i$。

$$\begin{aligned}F_j&=(2^j-1)F_jx+(2^k-2^j)F_{j-1}x\\&=\frac{2^k-2^j}{1-(2^j-1)x}F_{j-1}\end{aligned}$$

$$f_{i,j}=[x^i]\prod_{t=1}^{j}\frac{(2^k-2^t)x}{1-(2^t-1)x}$$

转化为线性递推，$f_{n,j}=[x^{j-1}](x^{n-1} \mod \prod_{t=1}^{j}(x-(2^t-1)))\prod_{t=1}^{j}(2^k-2^t)$

令 $g_i=(2^i-1)^{n-1}$

注意到多项式中也有中国剩余定理，因而有：

$$\begin{aligned}[x^{m-1}](x^{n-1} \mod \prod_{t=1}^{m}(x-(2^t-1)))&=\sum_{i=1}^{m}g_i[x^{m-1}]\prod_{j\neq i,j\in[1,m]}\frac{x-(2^j-1)}{2^i-2^j}\\&=\sum_{i=1}^{m}g_i\prod_{j\neq i,j\in[1,m]}\frac{1}{2^i-2^j}\\&=\sum_{i=1}^{m}g_i\prod_{j=1}^{i-1}\frac{1}{2^i-2^j}\prod_{j=i+1}^{m}\frac{1}{2^i-2^j}\\&=\sum_{i=1}^{m}g_i2^{i(m-i)}\prod_{j=1}^{i-1}\frac{1}{2^i-2^j}\prod_{j=1}^{m-i}\frac{1}{1-2^j}\end{aligned}$$

令 $a_i=\prod_{j=1}^{i-1}\frac{1}{2^i-2^j},b_i=\prod_{j=1}^{i}\frac{1}{1-2^j}$

$$[x^{m-1}](x^{n-1} \mod \prod_{t=1}^{m}(x-(2^t-1)))=\sum_{i=1}^{m}g_ia_ib_{m-i}2^{i(m-i)}$$

然后按照常规套路拆 $i(m-i)=\binom{m}{2}-\binom{i}{2}-\binom{m-i}{2}$，并令 $v_i=2^{\binom{i}{2}}$

则

$$[x^{m-1}](x^{n-1} \mod \prod_{t=1}^{m}(x-(2^t-1)))=v_m\sum_{i=1}^{m}\frac{g_ia_ib_{m-i}}{v_iv_{m-i}}$$

可以一次卷积求出所有答案。

我（i）

将修改操作差分成点到根的路径加，并按节点编号分块。

询问时散块求和就是 n 次点到根路径的加和 $q\sqrt{n}$ 次点值询问，可以转化成 n 次单点修改和 $q\sqrt{n}$ 次子树求和。可以在 dfs 序上使用 $O(\sqrt{n})$ 区间求和 $O(1)$ 单点修改的分块做。

整块求和就是维护一个数组 f_i 表示块 i 的点权和。预处理 $g_{u,i}$ 表示 u 到根路径上有几个编号在块 i 中的点。修改时将每个块 i 对应的 f_i 加上 $g_{u,i}\times x$ 即可。

时空复杂度都是 $O(n\sqrt{n})$。

【Bonus】

实际上还可以牺牲一定的时间常数来换取空间复杂度上的优势：容易选出 $O(\sqrt{n})$ 个关键点存储 g ，使得每个点的祖先中都有一个与之距离不超过 $O(\sqrt{n})$ 的关键点，然后求普通点的 g 数组时，往上跳到关键点即可。这样空间复杂度变成 $O(n)$ ，不过本题不卡空间所以并不需要这个优化。

他（it）

【做法 1】

手玩一下 $k=2$ 时的矩阵：

$$L_{n+1}^2=(L_n+L_{n-1}+1)^2=L_n^2+L_{n-1}^2+2L_nL_{n-1}+2L_n+2L_{n-1}+1$$

L_n^2, L_{n+1}^2 肯定是需要维护的。

还需要在矩阵中维护 L_nL_{n+1} ，它可以被表示为 $(L_n+L_{n-1}+1)L_n=L_n^2+L_nL_{n-1}+L_n$ ，容易被表示出来。

现在考虑一般的情况：

$$L_{n+1}^k=(L_n+L_{n-1}+1)^k=\sum_{i=0}^{k}\sum_{j=0}^{k-i}\binom{k}{i,j,k-i-j}L_n^iL_{n-1}^j$$

需要维护所有的 L_iL_j ，只有 $O(k^2)$ 个。

矩阵中转移的系数，是容易 $O(k^4)$ 计算的，总复杂度 $O(k^6\log n)$ 。

【做法 2】

观察斐波那契数列 $F_0=1,F_1=1,F_{n+1}=F_n+F_{n-1}$ ，容易发现 $L_n=2F_n-1$ ，归纳易证。

设 $S_{k,n}=\sum_{i=0}^{n}L_i^k$ 。

$$S_{k,n}=\sum_{i=0}^{n}(2F_i-1)^k=\sum_{i=0}^{n}\sum_{j=0}^{k}\binom{k}{j}(2F_i)^j(-1)^{k-j}=\sum_{j=0}^{k}\binom{k}{j}(-1)^{k-j}2^j\sum_{i=0}^{n}F_i^j$$

希望快速计算 $\sum_{i=0}^{n}F_i^j$ 。

$$F_{n+1}^k=(F_n+F_{n-1})^k=\sum_{i=0}^{k}\binom{k}{i}F_n^iF_{n-1}^{k-i}$$

需要维护 $F_{n+1}^iF_n^{k-i}$ 。

$$F_{n+1}^iF_n^{k-i}=(F_n+F_{n-1})^iF_n^{k-i}=\sum_{j=0}^{i}\binom{i}{j}F_n^{k-i+j}F_{n-1}$$

是已经维护了的东西。

时间复杂度 $O(k^4\log n)$ 。

【做法 3】

有 $F_n=\frac{(\frac{1+\sqrt{5}}{2})^n-(\frac{1-\sqrt{5}}{2})^n}{\sqrt{5}}$ ，为方便起见，设 $p=\frac{1+\sqrt{5}}{2},q=\frac{1-\sqrt{5}}{2}$ ，注意到 $pq=-1$ 。

$$5^{\frac{k}{2}}F_n^k=(p^n-q^n)^k=\sum_{i=0}^{k}\binom{k}{i}(-1)^{k-i}(p^iq^{k-i})^n$$

显然 F_n^k 是 $(p^iq^{k-i})^n$ 的线性组合，又有 L_n^k 是 F_n^k 的线性组合，L_n^k 也是 $(p^iq^{k-i})^n$ 的线性组合。

考虑构造 $G(x)$ ，使得 p^iq^{k-i} 是它的根。

由于 $pq=-1$ ，$p^iq^{k-i}=p^{2i-k}$ 。

可以构造 $G(x)=\prod_{i=-k}^{k}(x-p^i)$ ，由于后面的一些步骤需要，我们把它写成 $G(x)=\prod_{i=-k}^{k}(x^2-p^{2i})$ 。

若 $G(x)=0$，则 $G(x)x^{n-4k-2}=0$ 。

有

$$x^n=-\sum_{i=0}^{k}g_ix^{n-4k-2+2i}$$

设 $a_n=(p^iq^{k-i})^n$ ，p^iq^{k-i} 为 $G(x)$ 的根，有

$$a_n=-\sum_{i=0}^{k}g_ia_{n-4k-2+2i}$$

L_n^k 是 a_n 的线性组合，也满足这个递推式。

现在考虑如何求出 g_i ：

$$5F_t^2=(p^t-q^t)^2=p^{2t}-2p^tq^t+q^{2t}=p^{2t}+p^{-2t}-2(-1)^t$$

因此

$$p^{2t}+p^{-2t}=5F_t^2+2(-1)^t$$

得到

$$(x^2-p^{2t})(x^2-p^{-2t})=x^4-(p^{2t}+p^{-2t})x^2+1=x^4-(5F_t^2+2(-1)^t)+1$$

最后有

$$G(x)=(x^2-1)\prod_{t=1}^{k}(x^4-(p^{2t}+p^{-2t})x^2+1)$$

递推仍然用矩阵快速幂求，不过矩阵大小为 $(4k+3)\times(4k+3)$ ，需要考虑常数。

时间复杂度 $O(k^3\log n)$ 。