省选模拟赛 - 题解

_rqy

2019/07/24

1 绝望

1.1 题目大意

给出若干个数 $a_1 \dots a_n$，要求选出一些组成一个序列（每个数可以选多次），选出的序列的前缀异或递增的前提下长度尽量长。

$n \leqslant 10^5, 1 \leqslant a_i \leqslant 10^{18}$。

1.2 解法 1

爆搜！

无脑爆搜，可以获得 10 分的好成绩。

1.3 解法 2

我们把问题转化一下，考虑维护一个序列的前缀异或。那么相当于最开始有一个数 0，每次从 a_i 中选出一个异或上去，要求每次都要变大，最大化次数。

这样的话，如果 $n \leqslant 15$，可以发现手上的数只有 $O(2^n)$ 种可能的取值，可以 $O(n2^n)$ DP。25 分。

1.4 解法 3

当 $n \leqslant 1000, a_i \leqslant 1000$ 时，手上的数最大到 1023，可以 $O(n \times \max A)$ DP。

结合前面的做法可以获得 40 分。

1.5 解法 4

如果我们拿着一个数 b 并异或上了某个 a_i，什么情况下这个 b 会变大?

记 a 的二进制最高位（或者说，最高的为 1 的位）在 $high(a)$。

考虑 b 和 $b \oplus a_i$，会发现它们最靠前的不同的位恰好在 $high(a)$ 的位置，因此 $b \oplus a_i > b$ 当且仅当 b 的第 $high(a)$ 位为 0。

在第四个子任务中，$10^{15} \leqslant a_i \geqslant 10^{18}$，这意味着 $50 \leqslant high(a_i) \leqslant 59$。

由于所有 a 的 $high$ 都大于等于 50，因此 b 里面更后面的位是完全没有用的，因此可以只记 b 的有用的位的状态。

总复杂度为 $O(n\frac{\max A}{\min A})$。

50 分。

1.6 解法 5

根据刚才的考虑，如果对第 k 位，没有任何一个 $high(a_i) = k$，那么这一位可以忽略掉；因此假设对每个 k 都存在一个 a_i 使得 $high(a_i) = k$。

可以发现，最优解必定满足如下条件：当我们把 b 变成 $b \oplus a_i$ 的时候，如果对某个 $k < high(a_i)$，a_i 的第 k 位为 1，那么 b 的第 k 位是 1。

原因如下：

对于一个 $0 \leqslant k < high(a_i)$，如果 a_i 的第 k 位是 1 而 b 的第 k 位是 0，我们可以找到一个 $high(a_j) = k$ 的 a_j，进行这样的操作：$b \to (b \oplus a_j) \to (b \oplus a_i \oplus a_j) \to (b \oplus a_i)$，可以发现这肯定是合法的并且比 $b \to (b \oplus a_i)$ 更长。

根据这个性质，我们可以认为一个 a_i 的作用，就是把 $high(a_i)$ 位上的一个 0 “变成”后面的一些 0。

因此令 f_k 表示第 k 位的 0 可以操作几次，每次操作就是找到一个 $high(a_i) = k$ 并把 k 这一位的 0 变成一些后面的 0。转移方程如下：

$$f_k = \max_{high(a_i)=k} \left(1 + \sum_{k'<k, a_i \text{ 的第}k' \text{ 位为1}} f_{k'} \right)$$

总复杂度 $O(n \log \max A)$。可以通过本题。

另外，最长前缀异或上升子序列我不会做。

2 easy

2.1 题目大意

一棵双根树是有一个根和一个反根的树。两个双根树的乘积是把第二棵树复制很多遍之后按第一棵树的形态拼起来。

给出一棵双根树，问他和自己乘 k 次得到的树中点对的距离和。

$n \leqslant 10^5, k \leqslant 10^9$。

2.2 解法 1

不需要弄明白乘法到底是啥，只需要知道 $k=1$ 的时候就是本来的树。

$n \leqslant 100$，可以枚举两个点，然后 DFS 求答案。复杂度 $O(n^3)$，5 分。

2.3 解法 2

我觉得大家不会 Naive 的写 $O(n^3)$。因此我给了一个 $O(n^2)$ 的点。一共 15 分。

2.4 解法 3

从这里开始你需要理解明白题意了。

$n^k \leqslant 10^6$，建出树来 $O(n^k)$ 计算即可。

至于怎么 $O(n)$ 计算其实很简单,考虑一个点 u 和它的父亲之间的边被经过了 $(n^k - \mathrm{size}(u))\mathrm{size}(u)$ 次，这东西加起来就可以了。

或者说，只需要求所有点的 size 之和、size^2 之和，就可以计算答案了。

30 分。

2.5 解法 4

$S=T$（$k \leqslant 1000$ 等会再说）。

这个时候你会发现，$A \circ B$ 中的结点，每一棵 B 树的根的 size 是它对应的 A 树上的点的 size 乘以 n_B，而其他结点的 size 就是它在 B 树上的 size。因此

$$\begin{aligned}
&\sum_{u,v} \mathrm{size}^2_{(u,v)} \\
=&\sum_u n_B^2 \mathrm{size}_A^2(u) + n_A \sum_{v \neq S_B} \mathrm{size}_B^2(v) \\
=&n_B^2 \sum_u \mathrm{size}_A^2(u) + n_A(\sum_v \mathrm{size}_B^2(v) - n_B^2)
\end{aligned}$$

对于 size 也类似。由于每次在右边乘上的都是同一棵树，因此 $n_B, \sum_v \mathrm{size}_B^2(v)$ 都是不变的。

只需要记下 $n, \sum \mathrm{size}, \sum \mathrm{size}^2$，就可以矩阵快速幂计算了。复杂度 $O(n + \log k)$。结合前面做法可以得到 40 分。

2.6 解法 5

树是一条长为 n 的链，两个根在两端点。

乘完还是链。

结合前面的做法可以得到 40/50 分。

2.7 解法 6

考虑类似解法 4，可以发现只有反根的祖先结点的 size 会被影响。

我们记下面一些东西：

- $P_A(i)$ 表示 A 中 i 是否为 T_A 的祖先（是则为 1，不是则为 0）；
- $\text{size}_A(i)$ 表示 A 中点 i 的子树大小；
- $k_A = \sum_i P(i)$ 表示 T_A 的祖先个数；
- $f_A = \sum_i \text{size}_A(i)$ 表示子树大小和；
- $g_A = \sum_i \text{size}_A^2(i)$ 表示子树大小平方和；
- $h_A = \sum_i P_A(i)\text{size}_A(i)$ 表示 T_A 的祖先的子树大小和。

$$
\begin{aligned}
P_{A\circ B}(i,j) &= P_A(i)P_B(j) \\
\text{size}_{A\circ B}(i,j) &= (\text{size}_A(i)-1)n_BP_B(j)+\text{size}_B(j) \\
n_{A\circ B} &= n_An_B \\
k_{A\circ B} &= \sum_{i,j} P_A(i)P_B(j) = k_Ak_B \\
f_{A\circ B} &= \sum_{i,j} \left((\text{size}_A(i)-1)n_BP_B(j)+\text{size}_B(j)\right) \\
&= (f_A-n_A)n_Bk_B+n_Af_B \\
g_{A\circ B} &= \sum_{i,j} \left((\text{size}_A(i)-1)n_BP_B(j)+\text{size}_B(j)\right)^2 \\
&= \sum_{i,j} \left((\text{size}_A(i)-1)^2n_B^2P_B(j)+2(\text{size}_A(i)-1)n_BP_B(j)\text{size}_B(j)+size_B^2(j)\right) \\
&= (g_A-2f_A+n_A)n_B^2k_B+2(f_A-n_A)n_Bh_B+n_Ag_B \\
h_{A\circ B} &= \sum_{i,j} P_A(i)P_B(j)\left((\text{size}_A(i)-1)n_B+\text{size}_B(j)\right) \\
&= (h_A-k_A)n_Bk_B+k_Ah_B
\end{aligned}
$$

直接记下 n_A, k_A, f_A, g_A, h_A，矩阵快速幂（或者直接套这个式子算）即可。

复杂度 $O(n+\log k)$，足以通过本题。

~~我觉得这是本次比赛最有意思的题。~~

3 remember

3.1 题目大意

求

$$\sum_{\emptyset \subsetneq S \subseteq \{1 \dots n\}} \sum_{i=1}^{\gcd(S)} i^k$$

$n \leqslant 10^9, k \leqslant 10000$。

3.2 解法 1

枚举集合，暴力计算。可以获得 10 分。

3.3 解法 2

对每个数计算 gcd 为它的个数。由于 $i | \gcd(S)$ 的方案数是 $2^{\lfloor n/i \rfloor} - 1$（就是只选 k 的倍数），所以根据莫比乌斯反演，$i = \gcd(S)$ 的方案数就是

$$\sum_{i|t} \mu(t/i)(2^{\lfloor n/t \rfloor} - 1)$$

$O(n(\log n + \log k))$ 即可。30 分。

3.4 解法 3

对于 $k = 0$ 的情况，$F_k(n) = n$，所以就是要求

$$\sum_{i} i \sum_{i|t} \mu(t/i)(2^{\lfloor n/t \rfloor} - 1)$$

交换求和顺序，并发现 $\sum_{i|t} i\mu(t/i) = \varphi(t)$，于是答案就是

$$\sum_{t} \varphi(t)(2^{\lfloor n/t \rfloor} - 1)$$

对 $\lfloor n/t \rfloor$ 数论分块，杜教筛求 φ 前缀和即可。

结合前面的做法可以得到 45 分。

3.5 解法 4

k 更大的时候，类似于 $F_0 * \mu = \varphi$，我们定义一个 $G_k = F_k * \mu$。可以发现要求的就是

$$\sum_{t} G_k(t)(2^{\lfloor n/t \rfloor} - 1)$$

换句话说，只需要求 G_k 在每个 $\lfloor n/t \rfloor$ 处的前缀和。

由于杜教筛要求找到一个 g 使得 g 和 $f * g$ 都容易求前缀和，很容易想到取 $g = \mathbf{1}$，这样 $G_k * \mathbf{1} = F_k$ 是一个多项式，看上去有规律了很多。

当 $k \leqslant 3$ 的时候，可以手算所有情况下的 F_k 的前缀和，期望得分 60 分。

~~当 $k \leqslant 20$ 的时候，可以手算所有情况下的 F_k 的前缀和，期望得分 70 分。~~

3.6 解法 5

手算小的情况可以发现 F_k 是 $k+1$ 次多项式，其前缀和是 $k+2$ 次多项式，因此可以利用拉格朗日插值公式在 $O(k)$ 的时间内求出单个函数值。

这样的话，预处理 $n \leqslant S$ 的情况，可以发现复杂度为

$$O(S(\log n + \log k) + \frac{n}{\sqrt{S}} + \frac{nk}{S})$$

随便取一个 $S = n^{2/3}$ （也是杜教筛的标准取法）可以达到

$$O(n^{2/3}(\log n + \log k) + n^{1/3}k)$$

加一些小技巧（比如只对素数做快速幂，线性求逆元，快速莫比乌斯反演），改一改 S 的取值，可以优化到

$$O\left(\sqrt{nk\left(\frac{\log k}{\log n} + \log\log n\right)}\right)$$

总之现场调一调最优的 S 就可以通过本题。