题解

decoqaq

A

$f(x)$ 是数位积，当 $f(x) \neq 0$ 时，x 的每个数位都是 $1-9$ 的整数，考虑对 gcd 有贡献的质数只有 $2, 3, 5, 7$，所以可以表示为 $f(x) = 2^a 3^b 5^c 7^d$ 这种形式。

于是考虑对于所有 (a, b, c, d) 分别求出满足 $f(x) = 2^a 3^b 5^c 7^d$ 的 x 的个数，然后用类似四维前缀和的方法求出答案。

考虑数位 dp，记 $f[i][0/1][a][b][c][d]$ 表示当前处理从高到低的第 i 位，0/1 表示高位是否达到上界，当前为 (a, b, c, d) 的方案数，转移枚举数字即可，复杂度 $O(log_{10}^5 n)$。

记 dp 结果为 $g[a][b][c][d]$，答案即对所有 (a_1, b_1, c_1, d_1) 和 (a_2, b_2, c_2, d_2) 且其 gcd 小于等于 k 的数之和，可以用四维前缀和优化。再记 $h[a][b][c][d]$ 表示选两个 (a, b, c, d) 且至少有一维两者都大于下界，也可以用类似的方法计算。最后计算答案即可。

B.

10pts

模拟整个变换过程即可，时间复杂度 $O(n^3)$。

30pts

可以发现，第 i 个位置的数经过一次变换后会到达第 $2i\ mod\ (n+1)$ 的位置，假如长度为 n 的序列经过 k 次变换后，第一个位置的数回到了第一个位置上，那么就有

$2^k \equiv 1\ (mod\ (n+1))$

对于其他任何一个位置的数就有

$i * 2^k \equiv i\ (mod\ (n+1))$

因此，如果第一个数回到了第一个位置，那么其它数也会归位，就得到了初始排列。所以我们只需要枚举第一个数就好了。时间复杂度 $O(n^2)$。

40pts

观察上面出现过的式子

$2^k \equiv 1\ (mod\ (n+1))$

由于要经过 n 次变换后恰好首次回到初始排列，那么 n 就是满足上述式子中最小的 k。由欧拉定理可以得到

$2^{\phi(n+1)} \equiv 1\ (mod\ (n+1))$

如果 $n+1$ 不是质数，那么 $\phi(n+1)$ 就会小于 n，此时 n 就一定不是最小的 k 了。所以只有当 $n+1$ 是质数时，我们才去判断 n。时间复杂度 $O(\frac{n^2}{logn})$。

50 or 60pts

由于上面几个做法中，判断过程都是 $O(n)$ 的，效率极其低下，因此考虑优化。

回顾判断过程：

我们想知道下列式子中 n 是否为最小满足条件的 k

$2^k \equiv 1\ (mod\ (n+1))$

由于 $n+1$ 是个质数，所以 n 一定满足上述式子。如果最小的 k 不是 n，那么这个最小的 k 一定是 n 的约数。因此我们枚举 n 的约数，然后快速幂判断即可。判断个数为 $O(\frac{n}{logn})$，约数个数均摊为 $O(logn)$，快速幂为 $O(logn)$，总复杂度就是 $O(nlogn)$ 了。

100pts

判断过程可以进一步优化。

把 n 质因数分解得到 $n = p_1^{q_1}p_2^{q_2}...p_t^{q_t}$，如果最小的 k 不是 n，由于 k 是 n 的约数，那么 k 就是 $\frac{n}{p_1},\frac{n}{p_2},...,\frac{n}{p_t}$ 其中至少一个数的约数，即这些数中有至少一个满足 $2^{\frac{n}{p_i}} \equiv 1\ (mod\ (n+1))$。也就是说，我们只需要枚举 n 的所有质因数就好了。

不同于约数个数，在 10^7 范围内，不同的质因数个数最多的数也仅仅只有 8 个，其它的数则远远达不到这个值。因此，在完全算满的情况下，判断个数为 $O(\frac{n}{logn})$，质因数个数为 $O(8)$，快速幂为 $O(logn)$，总复杂度也仅仅只有 $O(8n)$，足以通过所有数据。

C.

我们把操作看成一棵 k 叉树，其中每个节点有权值，所有叶子节点 (共 $n+m$ 个) 就是 0 或 1。

除了叶子节点外的所有节点就代表一次合并，权值就是他们的平均值。

设一开始 0 点的深度分别为 $x_1, x_2 ... x_n$，1 的深度为 $y_1, y_2 ... y_m$。

那么根节点的权值为 $\sum(\frac{1}{k})^{y_i}$，而如果我们将所有点的权值改为 1，则根节点权值也为 1，那么有 $\sum(\frac{1}{k})^{x_i} + \sum(\frac{1}{k})^{y_i} = 1$，而如果满足这个条件，则一定可以构造出一种方案。

那么问题转化为有多少个 z 能写成 n 个 $(\frac{1}{k})^x$，$1-z$ 能写成 m 个 $(\frac{1}{k})^y$ 相加的形式。

我们将 z 表示为 $(0.c_1c_2...)_k$，那么若不进位 $\sum c = m$，而进位的话进位一次就减去 $k-1$，那么 $\sum c = m (\bmod\ k-1)$。假设小数有 len 位，那么 $1-z$ 的和应为 $(len-1)(k-1) + k - \sum c = len(k-1) - \sum c + 1$。

然后设 $f_{i,j}$ 表示到第 i 位，目前和为 j 的方案数即可，因为末尾不能为 0，所以要多开一维记一下最后一位是否为 0。