α

可以用找规律的方式得到答案。

结论是：对 $1 \sim 2^n$（$n \in \mathbb{N}^+$），一定可以将其分为两组，使得每组的 $0 \sim n-1$ 次幂的和均相等（0 次幂和相等说明了每组都只有 2^{n-1} 个数）。

考虑使用数学归纳法证明这个结论并给出构造。

当 $n=1$ 时显然正确。

假设 $1 \sim 2^n$ 被分为了两份：

$a_1, a_2, \cdots, a_{2^{n-1}}$，$b_1, b_2, \cdots, b_{2^{n-1}}$，且满足 $0 \sim n-1$ 次幂和均相等。

令第一个集合为：

$$a_1, a_2, \cdots, a_{2^{n-1}}, b_1+2^n, b_2+2^n, \cdots, b_{2^{n-1}}+2^n$$

令第二个集合为：

$$b_1, b_2, \cdots, b_{2^{n-1}}, a_1+2^n, a_2+2^n, \cdots, a_{2^{n-1}}+2^n$$

要证明这两个集合的 k 次幂相等，只需对形如 $(a+2^n)^k$ 进行二项式展开。其中 $0 \sim k-1$ 次幂已经证明，对于 k 次幂，前面的不带 2^n 的项会与后面形成互补（由 a 与 b 的 $0 \sim n-1$ 次幂和相等），因此也相等。

有了这个结论，这题就变成了输入一个数，输出一些数了。

β

考虑数位 dp 来求满足一定条件的 β 数个数。

枚举最后的总和 s，这样在记录数位乘积的时候可以直接对 s 取模。

用 $dp[s][w][x][y]$ 表示假设最终的数位和是 s，当前考虑到第 w 位，当前数位乘积为 x，当前数位总和为 y 的方案数。转移枚举下一个数位放了什么即可。

最后，统计进答案里的必须满足数位和与之前假设的数位和假设的相等，即 $s=y$。

令 $M = \log_{10} n \times 9$，则这部分计算可在 $O(M^4)$ 完成。

统计答案时，先预处理出前 k 位数总共有多少个 β 数，然后先确定有多少位，再从高到低逐位确定答案即可。

使用记忆化搜索可以简化代码。

也可以直接记录乘积含有的 $2,3,5,7$ 质因子个数进行 dp，这样可能会更快。

γ

每个格子的取值为 0 或 1，将格子看作未知数，则限制相当于一个方程。所有限制构成一个异或方程组。直接使用高斯消元即可 $O(n^6)$。

使用 bitset 优化消元的过程，即可 $O(\frac{n^6}{w})$。

将第 i 行第 j 列的元编号为 $(i-1)\times m+j$，则每个方程的最左边的非零系数和最右边的非零系数的编号只差不超过 $2m$。这样，在高斯消元的过程中，一个元只会存在于 $2m$ 个方程中（且按顺序排列所有方程，这些方程是连续的），因此只需要枚举这些方程进行消元。同时，由于只涉及 $2m$ 个元，因此消元的时候也只需要对这 $2m$ 个元进行消元。

这样复杂度就降为 $O(n^4)$。

对上述过程用 bitset 优化即可得到 $O(\frac{n^4}{w})$。

δ

以下称一个素数的正整数次幂为素数幂。

能够作为答案的数只有素数幂。

原因是，若有一个数满足 $p\times q$ 是无法表示出来的数（p,q 互质），那么 p 和 q 肯定至少有一个无法被表示出来。这样最终肯定会得到一个素数幂。

由于 n 只有 10^5，因此答案最大不超过第 10^5+1 个质数，大概 1.3×10^6。可以认为这是 $n\log n$ 级别的数。而 1.3×10^6 中的素数幂的总数是 $O(n)$ 级别的数。

这可以推出，如果 a_i 不是素数幂，或者 a_i 很大，那么这个 a_i 就没用了。

我们考虑从小到大枚举区间的右端点 r，并维护 d_i。其中 d_i 表示，左端点 l 落在 $1\sim d_i$ 时，区间 $[l,r]$ 能表示 i。这个 d_i 显然应该是符合条件的最大的。

考虑新加入了一个 p^k。那么对每个 p^a 满足 $a\geq k$ 的，我们可以用类似 dp 的方式使用 $d[p^{a-k}]$ 来更新 $d[p^a]$。

查询的时候，我们实际上是想找到最小的 i，满足 $d_i<l$。那么可以使用线段树维护这个 d 数组，查询的时候直接在线段树上二分即可。修改的时候只需要单点修改。

由于很多数都不是素数幂，因此可以把 d_i 的 i 的定义改成第 i 小的素数幂，这样线段树的大小就只有 $O(n)$ 级别。

由于每个 r 需要进行 $\log$ 次的单点修改，因此时间复杂度为 $O(n\log^2 n+m\log n)$。