# 数正方体（cube.in/out）

首先显然要求 $a_i \geq a_{i+1}$。

考虑模拟，枚举一个 $a_i$，考虑它相对应的横向数量的个数，并添加 $a_i$ 的限制或横向数量的限制。利用两个指针维护即可。

也有别的简单写法，但都是类似的模拟。

# 将不可解的我的一切（myself.in/out）

首先，对于任意一个数，都可以在最后加零而数位和不变，于是 $0 \in T$。

接下来只需要讨论每一位都不为 $0$ 时的答案。如果一个数 $c \in T$，那么将 $c$ 分解为质因数乘积后，其每个质因数都小于 $k$，并且质因数乘次数的和小于等于 $n$。如果不足 $n$，可以补充很多 $1$，所以也是可以的。

考虑对质因数 dp。令在当前质因数的和为 $n$ 时的乘积方案数为 $f_n$。转移只需要枚举下一个质因数，做完全背包就可以了。最后要求的答案是 $f$ 的前缀和再加一。

对于多组测试数据，可以离线按照 $k$ 大小排序处理。最后要求的前缀和事实上可以看作一开始让所有 $f$ 都为 $1$ 之后直接 dp。于是我们在 $O(t + \frac{nk}{\log k})$ 的复杂度内解决了该问题。（$k$ 以内的质数个数是 $O(\frac{k}{\log k})$ 的）

# 基础矩阵问题（matrix.in/out）

## $n \leq 20$

当方程数量很小时，我们考虑逐个确定 $X$ 每个分量的值。

假设我们已经确定了 $X$ 的前 $i$ 个分量，即令 $X_i = [x_1, x_2, \ldots x_i, 0, 0, 0, \ldots]^T$，此时可以计算得到 $b_i = AX_i$。$b$ 是一个 $n$ 维 01 向量，只有 $2^n$ 种取值，可以 dp：

$f_{i,S}$ 表示确定了前 $i$ 个分量，此时计算出的 $b_i$ 的状态是 $S$ 时，$C \cdot X_i$ 的最大值。

转移时考虑确定 $X$ 的下一个位置填 $0$ 还是 $1$，可以定出 $b_{i+1}$，确定接下来转移的状态。时间复杂度 $O(2^n \times m)$。

# $m \leq 40$

这个子任务限制了变量数的规模。我们考虑把 $X$ 分成上下两块，进行 meet in the middle。

设 $X = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ v \end{bmatrix}$，那么
$AX = A\begin{bmatrix} u \\ 0 \end{bmatrix} + A\begin{bmatrix} 0 \\ v \end{bmatrix} = s + t = b$。这里加法是模 2 意义下的。

让 $u$ 的长度为 $\left\lfloor \frac{m}{2} \right\rfloor$，则 $v$ 的长度是 $m - \left\lfloor \frac{m}{2} \right\rfloor$。

暴力枚举 $u$ 的取值，能算出对于任意的 $s$，最大的 $C \cdot \begin{bmatrix} u \\ 0 \end{bmatrix}$ 的取值 $L_s$。同理暴力枚举 $v$ 的取值，可以算出对任意的 $t$，最大的 $C \cdot \begin{bmatrix} 0 \\ v \end{bmatrix}$ 的取值 $R_t$。

当且仅当 $s + t = b$ 时，$\begin{bmatrix} u \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ v \end{bmatrix}$ 构成了一组 $AX = b$ 的合法解，此时最优的取值是 $L_s + R_t$。

因此枚举 $s$，则 $t = b - s = b + s$（加法是模 2 意义下的，可以看成两个压位整形做异或）。如果我们用 $S$ 表示状态 $s$ 的二进制压位值，$T$ 表示状态 $t$ 的二进制压位值，则对于一个 $s$，可以用 $L_S + R_{S \bigoplus T}$ 更新答案。时间复杂度 $O(2^{m/2} \times n)$。

省流：

这个问题相当于确定 $m$ 个变量满足 $n$ 个限制。

把这 $m$ 个变量拆成前一半和后一半，分别做。

暴力枚举出前一半变量的值，可以算出这些变量对限制的贡献 $s$，贡献为 $s$ 时这些变量对答案的贡献记为 $L_s$。类似地，可以算出后半部的的贡献 $R_t$
。
我们要求前后两部分对限制的贡献拼起来是 $b$，因此要求 $s \oplus t = b$。
分别算出 $L_s$ 和 $R_t$ 的所有状态，找到 $\max\{L_s + R_t | s \oplus t = b\}$ 即可。

## 总的做法

注意到上面两种数据规模总能取到其中一个，因此根据输入的 $n$ 的大小选取两个做法之一就能得到复杂度正确的做法。

# 人类为人类而写的文字（people.in/out）

## 算法 1

直接模拟退火！期望得分：？

## 正解

其实大部分特殊性质都是正解去掉一些东西。

首先给出答案的结论：将初始序列拆成任意多份和拆成小于等于三份时，答案序列长度相同。接下来构造证明之。

可以枚举一个起始值，求得一个最大的答案序列长度。为方便叙述，认为序列中的元素是 $k$ 种颜色。设开始的颜色是 $s$，结束的颜色是 $e$，在颜色上运算时均模 $k$。

之后将不必要的段删去。具体的，记 $a_i$ 为颜色 $i$ 多余的个数。随后从一个 $a_i = 0$ 的颜色 $i$ 开始，找到第一个 $a_j \neq 0$ 的颜色 $j$。可以证明，一定有一个未被划分过第一段的原始序列的开始颜色是 $j$。然后贪心选这个序列直到这个序列无了，或是到达了一个 $a_k$ 为 $0$ 的位置。

证明：当颜色 $j \neq e+1$ 时，此时的 $j-1$ 的个数小于 $j$ 的个数，因此必然存在一个以 $j$ 为开始的序列；假如这个序列已经被割掉第一段了，那这个第一段能够延长，是不可能的。而 $j = e+1$ 是不可能出现的，因为如果出现，那么 $a_j \neq 0$，那么答案序列可以继续增大。

之后直接开始构造，一段段向答案序列填。

若存在以当前需要的颜色为开头的序列，则直接贪心填完。

若不存在，则当前需要的颜色一定是 $e+1$，原因类似上面的证明。然后可以直接随意找到一个含有 $e+1$ 的序列，贪心的选第一个 $e+1$ 和后面的所有元素。每一个序列不会被这样处理大于一次，因为处理完一次之后就不存在颜色 $e+1$ 了。