2024年广东省大学生程序设计竞赛(GDCPC)暨CCPC广州邀请赛

2024 年 5 月 26 日

清华大学学生算法协会

感谢 Mys.C.K., liuzhangfeiabc, Itst, SpiritualKhorosho, EliminateSpace, xiaolilsq, gyh-20, JohnVictor36, Renshey, QAQAutoMaton 负责命题和验题工作.

A 祥子的工作 by *johhhh*

B 分解重构 by *SpiritualKhorosho*

C 小班课 by *gyh20*

D 树上的数 by *xiaolilsq*

E 计数题 by *rsy*

F 集合划分 by *xiaolilsq*

F 集合划分 by *xiaolilsq*

F 集合划分 by *xiaolilsq*

F 集合划分 by *xiaolilsq*

F 集合划分 by *xiaolilsq*

A 祥子的工作 by *johhhh*

题目描述

已知$X_1, X_2, \cdots, X_n$满足一个离散的概率分布，且独立同分布，
求$\Pr[X_1 + X_2 + \cdots + X_n \geq a]$

算法

签到题。

设$p_{i,j}$表示前i天恰好赚j分的概率，有递推公式$p_{i,j} = \frac{1}{101}\sum_{k=0}^{100} p_{i-1,j-k}$，拿个前缀和优化就可以.

时间复杂度$O(100na)$

丰川祥子的姓名第一个字拼起来（丰祥）意外地像一个中国名字

B 分解重构 by *SpiritualKhorosho*

简要题意

给定序列 $a_1, \cdots, a_n$，求满足以下要求的序列 $b_1, \cdots, b_n$ 的数量:

- $\prod_{i=1}^{n} b_i = m$；
- $1 \le \sum_{i=1}^{n} [a_i \neq b_i] \le l$，其中 $[x]$ 当 x 成立时为 1，否则为 0.

$1 \le l \le n \le 100, 1 \le a_i \le m \le 10^{11}$.

我会暴力

不难想到 DP: 记 $f(i,j,k)$ 表示 $b_1,\cdots,b_i$ 中有 j 个与 $a.$ 不同，前缀乘积为 k 的方案数. 该 DP 中，显然 k 为 m 的因数时状态才有意义.

枚举 b_i 转移:

$$f(i,j,k)=\sum_{d|m} f\left(i-1,j-[d\neq a_i],k/d\right)$$

直接实现上述 DP，复杂度为 $O\left(nld^2(m)\right)$，其中 $d(m)$ 表示 m 的因数个数. 在本题数据范围下，$d(m)\leq 4032$，$d^2(m)\leq 16,257,024$，在 $m=97,772,875,200$ 时取等.

显然，这一复杂度无法通过本题.

优化一

观察转移方程

$$f(i,j,k)=\sum_{d|m} f\left(i-1,j-[d\neq a_i],k/d\right)$$

考虑主动转移，即枚举 m 的两个因数 k/d 和 d 进行统计. 如果只枚举满足 $(k/d)\cdot d\leq m$ 的那些因数组合，则可以少处理一些无效转移.

这个优化可以将需要处理的因数对数从原来的 $16{,}257{,}024$ 对降低到 $8{,}130{,}527$. 应该没有人只写这个优化可以通过本题.

优化二

$$f(i,j,k)=\sum_{d|m} f\left(i-1, j-[d\neq a_i], k/d\right)$$

更进一步地，我们只需要预处理那些满足 $(d_1\cdot d_2)|m$ 的因数对 (d_1,d_2) 进行转移即可.

打表可知，这个优化可以将需要处理的因数对数从原来的 $16,257,024$ 对降低到 $816,480$（恰好在同一个 m 时取到最大值）.

出题人写的这个做法在 OJ 上需要 1.5s 左右，希望没有人用这个做法松过本题.

推式子

$$
\begin{aligned}
f(i,j,k) &= \sum_{d|m} f\left(i-1, j-[d \neq a_i], k/d\right) \\
&= \sum_{d|m} f\left(i-1, j-1, k/d\right) - f\left(i-1, j-1, k/a_i\right) + f\left(i-1, j, k/a_i\right) \\
&= \sum_{d|k} f\left(i-1, j-1, d\right) - f\left(i-1, j-1, k/a_i\right) + f\left(i-1, j, k/a_i\right)
\end{aligned}
$$

求和符号的部分可以看成在 m 的素因数上高维前缀和，后面部分对 a_i 预处理后可以 $O(1)$ 计算. 由此可将 DP 复杂度降至 $O(nld(m)\omega(m))$，其中 $\omega(m)$ 表示 m 的不同素因数个数.

$2 \times 3 \times 5 \times \cdots \times 29 = 6,469,693,230 < 10^{11} < 6,469,693,230 \times 31$，故 $d(m)\omega(m) \leq 40320$，可以快速通过本题. 总复杂度还要加上求所有因数的复杂度，标程是直接 $O\left(\sqrt{m}\right)$ 处理的.

高维前缀和

由于每个数 a 可以被唯一分解为 $a = \prod_{p\in\mathbb{P}} p^{c_p}$，故可以将 a 看成无限维坐标中的一个点 $(c_2, c_3, c_5, c_7, \cdots)$.

求 $\sum_{d|k} f(\cdot, \cdot, d)$ 相当于一个高维矩形区域求和，仿照二维前缀和的处理方式求 $\omega(m)$ 遍即可.

C 小班课 by *gyh20*

考虑我们如何处理一次询问.

考虑我们如何处理一次询问.

任意时刻，一些小班是满员的，其余的未满员. 我们可以假设每个人都去到自己意向最高的小班，直到某一个小班变满，可以发现，总共只会有 m 次小班变满的过程. 每一次我们可以直接二分查找找到这个位置.

考虑我们如何处理一次询问.

任意时刻，一些小班是满员的，其余的未满员. 我们可以假设每个人都去到自己意向最高的小班，直到某一个小班变满，可以发现，总共只会有 m 次小班变满的过程. 每一次我们可以直接二分查找找到这个位置.

具体二分查找时，我们处理的询问为，在区间 $l \sim r$ 中有多少人满足其在小班集合 S （S 为当前未满员的小班构成的集合）中优先度最高的是 x. 可以发现这个区间信息量是 $O(2^m m)$ 的，使用线段树维护，总复杂度为 $O(2^m mn\log n + qm^2 \log n)$.

D 树上的数 by *xiaolilsq*

D 树上的数 by *xiaolilsq*

如果每条边上最后的数字都大于 0，那么有一种众所周知的构造方式使得每条边上的数字都为断掉这条边后剩下两个连通块中大小较小的那个连通块大小（具体而言就是找到重心，然后各个子树中互相配对操作），这样操作下来在 $n \geq 3$ 的情况下出现次数最多的数字为 1，其出现次数为叶子数量.

D 树上的数 by *xiaolilsq*

如果每条边上最后的数字都大于 0，那么有一种众所周知的构造方式使得每条边上的数字都为断掉这条边后剩下两个连通块中大小较小的那个连通块大小（具体而言就是找到重心，然后各个子树中互相配对操作），这样操作下来在 $n\geq 3$ 的情况下出现次数最多的数字为 1，其出现次数为叶子数量.

事实上在 $n\geq 3$ 的时候显然每个叶子连上去的边数字只能为 1，所以我们的构造已经达到最优了.

D 树上的数 by *xiaolilsq*

如果每条边上最后的数字都大于 0，那么有一种众所周知的构造方式使得每条边上的数字都为断掉这条边后剩下两个连通块中大小较小的那个连通块大小（具体而言就是找到重心，然后各个子树中互相配对操作），这样操作下来在 $n \geq 3$ 的情况下出现次数最多的数字为 1，其出现次数为叶子数量.

事实上在 $n \geq 3$ 的时候显然每个叶子连上去的边数字只能为 1，所以我们的构造已经达到最优了.

额外需要注意的是 $n = 2$ 的时候最优答案为 1，比叶子数量少 1.

D 树上的数 by *xiaolilsq*

接下来考虑有 0 的情况，那么所有操作都不能跨过钦定为数字 0 的边，所以我们完全可以把这些数字为 0 的边断掉，然后剩下若干个连通块，数字 1 的出现次数为这些连通块叶子数量之和减去大小为 2 的连通块.

D 树上的数 by *xiaolilsq*

接下来考虑有 0 的情况，那么所有操作都不能跨过钦定为数字 0 的边，所以我们完全可以把这些数字为 0 的边断掉，然后剩下若干个连通块，数字 1 的出现次数为这些连通块叶子数量之和减去大小为 2 的连通块.

我们可以证明，如果剩下多个连通块，我们把它们连接起来数字 1 的出现次数不会更多，而数字 0 的出现次数会减少，所以存在最优解满足数字非 0 的那些边组成的是一个连通块.

接下来考虑有 0 的情况，那么所有操作都不能跨过钦定为数字 0 的边，所以我们完全可以把这些数字为 0 的边断掉，然后剩下若干个连通块，数字 1 的出现次数为这些连通块叶子数量之和减去大小为 2 的连通块.

我们可以证明，如果剩下多个连通块，我们把它们连接起来数字 1 的出现次数不会更多，而数字 0 的出现次数会减少，所以存在最优解满足数字非 0 的那些边组成的是一个连通块.

不妨钦定数字 1 的出现次数，也就是钦定叶子数量，我们的目的是尽可能减少数字 0 的出现次数，也就是说我们要找到一个满足叶子数量限制的尽可能大的连通块，这使用长链剖分贪心做就行了.

D 树上的数 by *xiaolilsq*

接下来考虑有 0 的情况，那么所有操作都不能跨过钦定为数字 0 的边，所以我们完全可以把这些数字为 0 的边断掉，然后剩下若干个连通块，数字 1 的出现次数为这些连通块叶子数量之和减去大小为 2 的连通块.

我们可以证明，如果剩下多个连通块，我们把它们连接起来数字 1 的出现次数不会更多，而数字 0 的出现次数会减少，所以存在最优解满足数字非 0 的那些边组成的是一个连通块.

不妨钦定数字 1 的出现次数，也就是钦定叶子数量，我们的目的是尽可能减少数字 0 的出现次数，也就是说我们要找到一个满足叶子数量限制的尽可能大的连通块，这使用长链剖分贪心做就行了.

同样需要注意 $n \le 3$ 可能要特殊处理.

E 计数题 by *rsy*

E 计数题 by *rsy*

定义无向图 G 的权值 $f(G)$ 为 G 中所有连通块大小的平方和.

对于 $1 \sim n$ 的所有错排 p，求 $f(G(p))$ 之和对 998244353 取模后的结果.

$2 \le n < 998244353$.

E 计数题 by *rsy*

考虑计算每个连通块（轮换）的贡献:

$f(n)=\sum_{k=2}^{n} k^2\binom{n}{k}(k-1)!g(n-k)$，其中 $g(n)$ 为长度为 n 的错排个数.

E 计数题 by *rsy*

考虑计算每个连通块（轮换）的贡献:
$f(n)=\sum_{k=2}^{n}k^2\binom{n}{k}(k-1)!g(n-k)$，其中 $g(n)$ 为长度为 n 的错排个数.

$$f(n)=\sum_{k=2}^{n}k^2\binom{n}{k}(k-1)!g(n-k)$$
$$=\sum_{k=2}^{n}k^2\frac{n!}{(n-k)!k!}(k-1)!g(n-k)=n!\sum_{k=2}^{n}k\frac{g(n-k)}{(n-k)!}$$

记 $A(x)=\sum_{k=2}^{\infty}kx^k=\frac{x}{(1-x)^2}-x$，$G(x)=e^{\ln\frac{1}{1-x}-x}=\frac{e^{-x}}{1-x}$，则
$f(n)=n![x^n]A(x)G(x)=n![x^n]\left(\frac{xe^{-x}}{(1-x)^3}-\frac{e^{-x}}{1-x}+e^{-x}\right)$.

E 计数题 by *rsy*

考虑计算每个连通块（轮换）的贡献:

$f(n)=\sum_{k=2}^{n}k^2\binom{n}{k}(k-1)!g(n-k)$，其中 $g(n)$ 为长度为 n 的错排个数.

$$
\begin{aligned}
f(n) &= \sum_{k=2}^{n}k^2\binom{n}{k}(k-1)!g(n-k) \\
&= \sum_{k=2}^{n}k^2\frac{n!}{(n-k)!k!}(k-1)!g(n-k) = n!\sum_{k=2}^{n}k\frac{g(n-k)}{(n-k)!}
\end{aligned}
$$

记 $A(x)=\sum_{k=2}^{\infty}kx^k=\frac{x}{(1-x)^2}-x$，$G(x)=e^{\ln\frac{1}{1-x}-x}=\frac{e^{-x}}{1-x}$，则 $f(n)=n![x^n]A(x)G(x)=n![x^n]\left(\frac{xe^{-x}}{(1-x)^3}-\frac{e^{-x}}{1-x}+e^{-x}\right)$.

可以拆成三部分计算，利用求导不难得到每部分的递推式. 使用分块打表或整式递推加速计算即可.

F 集合划分 by *xiaolilsq*

F 集合划分 by *xiaolilsq*

考虑 1 以及大于 $\sqrt{n}$ 的质数都必须要单独放在一个集合内，而我们通过构造的方式证明其它数字最多额外剩下一个数，其余数都可以放在大小大于 1 的集合中.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

- 如果 $p>2$ 且 $p(p+1)<n$，那么 p 的所有还未被分配的倍数为 $p,2p,\ldots,p^2$，此时我们可以将其全部分到同一个组内.

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

- 如果 $p>2$ 且 $p(p+1)<n$，那么 p 的所有还未被分配的倍数为 $p, 2p, \ldots, p^2$，此时我们可以将其全部分到同一个组内.
- 如果 $p>2$ 且 $p(p+1)\geq n$，那么 p 的倍数中至少有 $2p, 3p, \ldots, (p-1)p$ 和 $p(p+1)$，共有 $p-1$ 个还未被分配的质数，且它们的最小质因子都小于 p，亦或者说它们还可以在后面的过程中被分配。考虑那些不能在后面的过程中分配的数，也就是它们的最小质因子为 p，我们总是能够取这 $p-1$ 个还未被分配的数中的若干个补成 p 的倍数，这样它们总能够在这一次分配过程中分好，而剩下的可以丢到后面的过程去考虑.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

- 如果 $p>2$ 且 $p(p+1)<n$，那么 p 的所有还未被分配的倍数为 $p,2p,\dots,p^2$，此时我们可以将其全部分到同一个组内.
- 如果 $p>2$ 且 $p(p+1)\geq n$，那么 p 的倍数中至少有 $2p,3p,\dots,(p-1)p$ 和 $p(p+1)$，共有 $p-1$ 个还未被分配的质数，且它们的最小质因子都小于 p，亦或者说它们还可以在后面的过程中被分配。考虑那些不能在后面的过程中分配的数，也就是它们的最小质因子为 p，我们总是能够取这 $p-1$ 个还未被分配的数中的若干个补成 p 的倍数，这样它们总能够在这一次分配过程中分好，而剩下的可以丢到后面的过程去考虑.
- 如果 $p=2$，将所有数两个两个分组，最多会额外剩下一个，由此得证.

如果额外剩下一个，考虑通过细微调整的方法使其不会剩下.

F 集合划分 by *xiaolilsq*

如果额外剩下一个，考虑通过细微调整的方法使其不会剩下.

具体而言就是在 $p=3$ 的分配过程时，如果有一个大小为 3 的集合里面被分到的全都是 6 的倍数，那么我们就可以把这些数全部放到 $p=2$ 的分配过程中，这样就不会剩下这样一个多出来的了.

如果额外剩下一个，考虑通过细微调整的方法使其不会剩下.

具体而言就是在 $p = 3$ 的分配过程时，如果有一个大小为 3 的集合里面被分到的全都是 6 的倍数，那么我们就可以把这些数全部放到 $p = 2$ 的分配过程中，这样就不会剩下这样一个多出来的了.

实现后发现在 $n \geq 30$ 左右时总是可以这样调整，所以这样都是最优的。而不能调整的很少，可以通过证明如此构造得到的也是最优情况.

F 集合划分 by *xiaolilsq*

F 集合划分 by *xiaolilsq*

考虑 1 以及大于 $\sqrt{n}$ 的质数都必须要单独放在一个集合内，而我们通过构造的方式证明其它数字最多额外剩下一个数，其余数都可以放在大小大于 1 的集合中.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

- 如果 $p>2$ 且 $p(p+1)<n$，那么 p 的所有还未被分配的倍数为 $p,2p,\ldots,p^2$，此时我们可以将其全部分到同一个组内.

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

- 如果 $p>2$ 且 $p(p+1)<n$，那么 p 的所有还未被分配的倍数为 $p,2p,\dots,p^2$，此时我们可以将其全部分到同一个组内.
- 如果 $p>2$ 且 $p(p+1)\geq n$，那么 p 的倍数中至少有 $2p,3p,\dots,(p-1)p$ 和 $p(p+1)$，共有 $p-1$ 个还未被分配的质数，且它们的最小质因子都小于 p，亦或者说它们还可以在后面的过程中被分配。考虑那些不能在后面的过程中分配的数，也就是它们的最小质因子为 p，我们总是能够取这 $p-1$ 个还未被分配的数中的若干个补成 p 的倍数，这样它们总能够在这一次分配过程中分好，而剩下的可以丢到后面的过程去考虑.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

- 如果 $p>2$ 且 $p(p+1)<n$，那么 p 的所有还未被分配的倍数为 $p,2p,\ldots,p^2$，此时我们可以将其全部分到同一个组内.
- 如果 $p>2$ 且 $p(p+1)\geq n$，那么 p 的倍数中至少有 $2p,3p,\ldots,(p-1)p$ 和 $p(p+1)$，共有 $p-1$ 个还未被分配的质数，且它们的最小质因子都小于 p，亦或者说它们还可以在后面的过程中被分配。考虑那些不能在后面的过程中分配的数，也就是它们的最小质因子为 p，我们总是能够取这 $p-1$ 个还未被分配的数中的若干个补成 p 的倍数，这样它们总能够在这一次分配过程中分好，而剩下的可以丢到后面的过程去考虑.
- 如果 $p=2$，将所有数两个两个分组，最多会额外剩下一个，由此得证.

F 集合划分 by *xiaolilsq*

如果额外剩下一个，考虑通过细微调整的方法使其不会剩下.

如果额外剩下一个，考虑通过细微调整的方法使其不会剩下.

具体而言就是在 $p=3$ 的分配过程时，如果有一个大小为 3 的集合里面被分到的全都是 6 的倍数，那么我们就可以把这些数全部放到 $p=2$ 的分配过程中，这样就不会剩下这样一个多出来的了.

如果额外剩下一个，考虑通过细微调整的方法使其不会剩下.

具体而言就是在 $p = 3$ 的分配过程时，如果有一个大小为 3 的集合里面被分到的全都是 6 的倍数，那么我们就可以把这些数全部放到 $p = 2$ 的分配过程中，这样就不会剩下这样一个多出来的了.

实现后发现在 $n \geq 30$ 左右时总是可以这样调整，所以这样都是最优的。而不能调整的很少，可以通过证明如此构造得到的也是最优情况.

F 集合划分 by *xiaolilsq*

F 集合划分 by *xiaolilsq*

考虑 1 以及大于 $\sqrt{n}$ 的质数都必须要单独放在一个集合内，而我们通过构造的方式证明其它数字最多额外剩下一个数，其余数都可以放在大小大于 1 的集合中.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

- 如果 $p > 2$ 且 $p(p+1) < n$，那么 p 的所有还未被分配的倍数为 $p, 2p, \ldots, p^2$，此时我们可以将其全部分到同一个组内.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

- 如果 $p>2$ 且 $p(p+1)<n$，那么 p 的所有还未被分配的倍数为 $p, 2p, \ldots, p^2$，此时我们可以将其全部分到同一个组内.
- 如果 $p>2$ 且 $p(p+1) \geq n$，那么 p 的倍数中至少有 $2p, 3p, \ldots, (p-1)p$ 和 $p(p+1)$，共有 $p-1$ 个还未被分配的质数，且它们的最小质因子都小于 p，亦或者说它们还可以在后面的过程中被分配。考虑那些不能在后面的过程中分配的数，也就是它们的最小质因子为 p，我们总是能够取这 $p-1$ 个还未被分配的数中的若干个补成 p 的倍数，这样它们总能够在这一次分配过程中分好，而剩下的可以丢到后面的过程去考虑.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

- 如果 $p>2$ 且 $p(p+1)<n$，那么 p 的所有还未被分配的倍数为 $p,2p,\dots,p^2$，此时我们可以将其全部分到同一个组内.
- 如果 $p>2$ 且 $p(p+1)\geq n$，那么 p 的倍数中至少有 $2p,3p,\dots,(p-1)p$ 和 $p(p+1)$，共有 $p-1$ 个还未被分配的质数，且它们的最小质因子都小于 p，亦或者说它们还可以在后面的过程中被分配。考虑那些不能在后面的过程中分配的数，也就是它们的最小质因子为 p，我们总是能够取这 $p-1$ 个还未被分配的数中的若干个补成 p 的倍数，这样它们总能够在这一次分配过程中分好，而剩下的可以丢到后面的过程去考虑.
- 如果 $p=2$，将所有数两个两个分组，最多会额外剩下一个，由此得证.

如果额外剩下一个，考虑通过细微调整的方法使其不会剩下.

F 集合划分 by *xiaolilsq*

如果额外剩下一个，考虑通过细微调整的方法使其不会剩下.

具体而言就是在 $p=3$ 的分配过程时，如果有一个大小为 3 的集合里面被分到的全都是 6 的倍数，那么我们就可以把这些数全部放到 $p=2$ 的分配过程中，这样就不会剩下这样一个多出来的了.

F 集合划分 by *xiaolilsq*

如果额外剩下一个，考虑通过细微调整的方法使其不会剩下.

具体而言就是在 $p=3$ 的分配过程时，如果有一个大小为 3 的集合里面被分到的全都是 6 的倍数，那么我们就可以把这些数全部放到 $p=2$ 的分配过程中，这样就不会剩下这样一个多出来的了.

实现后发现在 $n\geq 30$ 左右时总是可以这样调整，所以这样都是最优的。而不能调整的很少，可以通过证明如此构造得到的也是最优情况.

F 集合划分 by *xiaolilsq*

F 集合划分 by *xiaolilsq*

考虑 1 以及大于 $\sqrt{n}$ 的质数都必须要单独放在一个集合内，而我们通过构造的方式证明其它数字最多额外剩下一个数，其余数都可以放在大小大于 1 的集合中.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

- 如果 $p>2$ 且 $p(p+1)<n$，那么 p 的所有还未被分配的倍数为 $p, 2p, \ldots, p^2$，此时我们可以将其全部分到同一个组内.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

- 如果 $p>2$ 且 $p(p+1)<n$，那么 p 的所有还未被分配的倍数为 $p,2p,\ldots,p^2$，此时我们可以将其全部分到同一个组内.
- 如果 $p>2$ 且 $p(p+1)\geq n$，那么 p 的倍数中至少有 $2p,3p,\ldots,(p-1)p$ 和 $p(p+1)$，共有 $p-1$ 个还未被分配的质数，且它们的最小质因子都小于 p，亦或者说它们还可以在后面的过程中被分配。考虑那些不能在后面的过程中分配的数，也就是它们的最小质因子为 p，我们总是能够取这 $p-1$ 个还未被分配的数中的若干个补成 p 的倍数，这样它们总能够在这一次分配过程中分好，而剩下的可以丢到后面的过程去考虑.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

- 如果 $p>2$ 且 $p(p+1)<n$，那么 p 的所有还未被分配的倍数为 $p,2p,\ldots,p^2$，此时我们可以将其全部分到同一个组内.
- 如果 $p>2$ 且 $p(p+1)\geq n$，那么 p 的倍数中至少有 $2p,3p,\ldots,(p-1)p$ 和 $p(p+1)$，共有 $p-1$ 个还未被分配的质数，且它们的最小质因子都小于 p，亦或者说它们还可以在后面的过程中被分配。考虑那些不能在后面的过程中分配的数，也就是它们的最小质因子为 p，我们总是能够取这 $p-1$ 个还未被分配的数中的若干个补成 p 的倍数，这样它们总能够在这一次分配过程中分好，而剩下的可以丢到后面的过程去考虑.
- 如果 $p=2$，将所有数两个两个分组，最多会额外剩下一个，由此得证.

如果额外剩下一个，考虑通过细微调整的方法使其不会剩下.

F 集合划分 by *xiaolilsq*

如果额外剩下一个，考虑通过细微调整的方法使其不会剩下.

具体而言就是在 $p=3$ 的分配过程时，如果有一个大小为 3 的集合里面被分到的全都是 6 的倍数，那么我们就可以把这些数全部放到 $p=2$ 的分配过程中，这样就不会剩下这样一个多出来的了.

如果额外剩下一个，考虑通过细微调整的方法使其不会剩下.

具体而言就是在 $p=3$ 的分配过程时，如果有一个大小为 3 的集合里面被分到的全都是 6 的倍数，那么我们就可以把这些数全部放到 $p=2$ 的分配过程中，这样就不会剩下这样一个多出来的了.

实现后发现在 $n\geq 30$ 左右时总是可以这样调整，所以这样都是最优的。而不能调整的很少，可以通过证明如此构造得到的也是最优情况.

F 集合划分 by *xiaolilsq*

F 集合划分 by *xiaolilsq*

考虑 1 以及大于 $\sqrt{n}$ 的质数都必须要单独放在一个集合内，而我们通过构造的方式证明其它数字最多额外剩下一个数，其余数都可以放在大小大于 1 的集合中.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

- 如果 $p > 2$ 且 $p(p+1) < n$，那么 p 的所有还未被分配的倍数为 $p, 2p, \ldots, p^2$，此时我们可以将其全部分到同一个组内.

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

- 如果 $p>2$ 且 $p(p+1)<n$，那么 p 的所有还未被分配的倍数为 $p,2p,\ldots,p^2$，此时我们可以将其全部分到同一个组内.
- 如果 $p>2$ 且 $p(p+1)\geq n$，那么 p 的倍数中至少有 $2p,3p,\ldots,(p-1)p$ 和 $p(p+1)$，共有 $p-1$ 个还未被分配的质数，且它们的最小质因子都小于 p，亦或者说它们还可以在后面的过程中被分配。考虑那些不能在后面的过程中分配的数，也就是它们的最小质因子为 p，我们总是能够取这 $p-1$ 个还未被分配的数中的若干个补成 p 的倍数，这样它们总能够在这一次分配过程中分好，而剩下的可以丢到后面的过程去考虑.

F 集合划分 by *xiaolilsq*

具体而言，从大到小一次考虑每个小于等于 $\sqrt{n}$ 的质数 p，然后考虑 p 所有还未被分配的倍数被分到哪些集合.

- 如果 $p>2$ 且 $p(p+1)<n$，那么 p 的所有还未被分配的倍数为 $p, 2p, \ldots, p^2$，此时我们可以将其全部分到同一个组内.
- 如果 $p>2$ 且 $p(p+1)\geq n$，那么 p 的倍数中至少有 $2p, 3p, \ldots, (p-1)p$ 和 $p(p+1)$，共有 $p-1$ 个还未被分配的质数，且它们的最小质因子都小于 p，亦或者说它们还可以在后面的过程中被分配。考虑那些不能在后面的过程中分配的数，也就是它们的最小质因子为 p，我们总是能够取这 $p-1$ 个还未被分配的数中的若干个补成 p 的倍数，这样它们总能够在这一次分配过程中分好，而剩下的可以丢到后面的过程去考虑.
- 如果 $p=2$，将所有数两个两个分组，最多会额外剩下一个，由此得证.

如果额外剩下一个，考虑通过细微调整的方法使其不会剩下.

F 集合划分 by *xiaolilsq*

如果额外剩下一个，考虑通过细微调整的方法使其不会剩下.

具体而言就是在 $p=3$ 的分配过程时，如果有一个大小为 3 的集合里面被分到的全都是 6 的倍数，那么我们就可以把这些数全部放到 $p=2$ 的分配过程中，这样就不会剩下这样一个多出来的了.

F 集合划分 by *xiaolilsq*

如果额外剩下一个，考虑通过细微调整的方法使其不会剩下.

具体而言就是在 $p=3$ 的分配过程时，如果有一个大小为 3 的集合里面被分到的全都是 6 的倍数，那么我们就可以把这些数全部放到 $p=2$ 的分配过程中，这样就不会剩下这样一个多出来的了.

实现后发现在 $n\geq 30$ 左右时总是可以这样调整，所以这样都是最优的。而不能调整的很少，可以通过证明如此构造得到的也是最优情况.

感谢倾听!