数论入门

施开成

北京大学

August 5, 2024

质数

Definition 1.1 (质数 & 合数)

对于一个 >1 的正整数 n，如果它只有两个因子 $1, n$，则称 n 为质数（素数），否则称 n 为合数。

质数

Definition 1.1 (质数 & 合数)

对于一个 >1 的正整数 n，如果它只有两个因子 $1,n$，则称 n 为质数（素数），否则称 n 为合数。

由素数定理，$1\sim n$ 之间的质数个数是 $O\left(\dfrac{n}{\log n}\right)$ 的。

算术基本定理

Theorem 1.1 (算术基本定理（正整数唯一分解定理）)

对于任意正整数 A，存在唯一一个集合 $\{(p_1, q_1), (p_2, q_2), \cdots, (p_n, q_n)\}$ 满足 $A = \prod_{i=1}^{n} p_i^{q_i}$，其中 p_i 是质数，q_i 是正整数。

算术基本定理

Theorem 1.1 (算术基本定理（正整数唯一分解定理）)

对于任意正整数 A，存在唯一一个集合 $\{(p_1, q_1), (p_2, q_2), \cdots, (p_n, q_n)\}$ 满足 $A = \prod_{i=1}^{n} p_i^{q_i}$，其中 p_i 是质数，q_i 是正整数。

换句话说，每个正整数 A 都对应一个无限长的非负整数序列 $\{a_i\}$，满足 $A = \prod_{i=1}^{\infty} p_i^{a_i}$，其中 p_i 表示第 i 个质数。我们把序列 $\{a_i\}$ 称为 A 的指数序列（名字是我随便取的）。

算术基本定理

Theorem 1.1 (算术基本定理（正整数唯一分解定理）)

对于任意正整数 A，存在唯一一个集合 $\{(p_1,q_1),(p_2,q_2),\cdots,(p_n,q_n)\}$ 满足 $A=\prod_{i=1}^{n} p_i^{q_i}$，其中 p_i 是质数，q_i 是正整数。

换句话说，每个正整数 A 都对应一个无限长的非负整数序列 $\{a_i\}$，满足 $A=\prod_{i=1}^{\infty} p_i^{a_i}$，其中 p_i 表示第 i 个质数。我们把序列 $\{a_i\}$ 称为 A 的指数序列（名字是我随便取的）。

事实上指数序列的定义方式也适用于有理数，甚至根式。

Definition 1.2 (带余除法)

对于被除数 n 和除数 p，存在唯一的整数 q, r 满足 $n = pq + r$，$0 \le r < p$。q 被称作带余除法的商，记作 $q = \left\lfloor \dfrac{n}{p} \right\rfloor$。$r$ 被称为带余除法的余数，记作 $r = n \bmod p$。

Definition 1.2 (带余除法)

对于被除数 n 和除数 p，存在唯一的整数 q, r 满足 $n = pq + r$，$0 \leq r < p$。q 被称作带余除法的商，记作 $q = \left\lfloor \frac{n}{p} \right\rfloor$。$r$ 被称为带余除法的余数，记作 $r = n \bmod p$。

推论：$n \bmod p = n - \left\lfloor \frac{n}{p} \right\rfloor p$。

Definition 1.2 (带余除法)

对于被除数 n 和除数 p，存在唯一的整数 q,r 满足 $n=pq+r$，$0\leq r<p$。q 被称作带余除法的商，记作 $q=\left\lfloor\dfrac{n}{p}\right\rfloor$。$r$ 被称为带余除法的余数，记作 $r=n \bmod p$。

推论：$n \bmod p = n-\left\lfloor\dfrac{n}{p}\right\rfloor p$。

Definition 1.3 (整除)

如果 n 除 p 的余数为 0，则称 p 能整除 n，记作 $p\mid n$。否则称 p 不整除 n，记作 $p\nmid n$。

Definition 1.4 (最大公约数)

两个数 a, b 的最大公约数定义为它们的指数序列每一位取 $\min$ 之后得到的数，记作 $\gcd(a, b)$。

Definition 1.4 (最大公约数)

两个数 a, b 的最大公约数定义为它们的指数序列每一位取 $\min$ 之后得到的数，记作 $\gcd(a, b)$。

Definition 1.5 (最小公倍数)

两个数 a, b 的最小公倍数定义为它们的指数序列每一位取 $\max$ 之后得到的数，记作 $\operatorname{lcm}(a, b)$。

Definition 1.4 (最大公约数)

两个数 a, b 的最大公约数定义为它们的指数序列每一位取 $\min$ 之后得到的数，记作 $\gcd(a, b)$。

Definition 1.5 (最小公倍数)

两个数 a, b 的最小公倍数定义为它们的指数序列每一位取 $\max$ 之后得到的数，记作 $\operatorname{lcm}(a, b)$。

由于 $\min(a, b) + \max(a, b) = a + b$，因此自然有 $\gcd(a, b) \operatorname{lcm}(a, b) = ab$。

Problem（P10548 【THUPC 2024 决赛】朔望（局部））

给定两个有理数，求它们的 lcm。

Problem (P10548 【THUPC 2024 决赛】朔望（局部））

给定两个有理数，求它们的 lcm。

设给定的两个数是 $\frac{p_1}{q_1}$ 和 $\frac{p_2}{q_2}$，且是约分后的形式。则它们的 lcm 为 $\frac{\operatorname{lcm}(p_1, p_2)}{\gcd(q_1, q_2)}$。

埃氏筛

埃氏筛算法可以在 $O(n \log\log n)$ 的时间内预处理出 $1 \sim n$ 中所有的质数。

埃氏筛

埃氏筛算法可以在 $O(n\log\log n)$ 的时间内预处理出 $1\sim n$ 中所有的质数。

具体地，我们从 2 到 n 扫描，如果当前数未被标记则将其加入质数序列中，并把它的所有倍数标记为合数。复杂度 $\sum_{p\le n,p\text{ is prime}}\frac{n}{p}=n\log\log n$。

P7960 [NOIP2021] 报数

设 $p(x)$ 表示 x 的十进制表示中是否含有数字 7，若含有则 $p(x)=1$，否则 $p(x)=0$。则一个正整数 x 不能被报出，当且仅当存在正整数 y 和 z，使得 $x=yz$ 且 $p(y)=1$。

T 组询问，每次给出 x，如果 x 不能报出则输出 -1，否则输出 x 之后要报的下一个数。

$1\le T\le 2\times 10^5, 1\le x\le 10^7$。

显然只要预处理出每个数是否合法即可。

显然只要预处理出每个数是否合法即可。

考虑埃氏筛，我们称所有含有数字 7 的数是“类质数”。从 1 到 n 扫描，如果当前数不是任何“类质数”的倍数，则检查该数本身是否是“类质数”。如果是，则对它的所有倍数进行标记。

预处理复杂度 $O(V\log V)$，但常数较小，可以通过。

P1835 素数密度

给定 L, R，请计算区间 $[L, R]$ 中素数的个数。
$1 \le L \le R < 2^{31}$，$R - L \le 10^6$。

借鉴埃氏筛的思路，扫描每个 $\leq \sqrt{R}$ 的质数，并把它们在 $[L, R]$ 中的倍数标记为合数。此时，$[L, R]$ 中剩余未被标记的数即为质数。

借鉴埃氏筛的思路，扫描每个 $\leq \sqrt{R}$ 的质数，并把它们在 $[L, R]$ 中的倍数标记为合数。此时，$[L, R]$ 中剩余未被标记的数即为质数。

复杂度 $O(\sqrt{R} \log\log R + (R-L)\log\log R)$，如果预处理质数的部分使用线性筛，则为 $O(\sqrt{R} + (R-L)\log\log R)$。

这种方法被称为区间筛。

线性筛

线性筛算法可以在 $O(n)$ 的时间内预处理出 $1 \sim n$ 中所有的质数。

线性筛

线性筛算法可以在 $O(n)$ 的时间内预处理出 $1 \sim n$ 中所有的质数。

令 $\mathrm{low}(n)$ 表示 n 的最小质因子。我们从 2 到 n 扫描，对于 i，我们枚举所有 $\le \mathrm{low}(i)$ 的质数 j，并将 ij 标记为合数。

线性筛

线性筛算法可以在 $O(n)$ 的时间内预处理出 $1 \sim n$ 中所有的质数。

令 $\mathrm{low}(n)$ 表示 n 的最小质因子。我们从 2 到 n 扫描，对于 i，我们枚举所有 $\le \mathrm{low}(i)$ 的质数 j，并将 ij 标记为合数。

可以发现，n 只会在 $\frac{n}{\mathrm{low}(n)}$ 处被标记，因此每个数只会被标记 1 次，总复杂度 $O(n)$。

P3383 【模板】线性筛素数

模板题。

质因子个数

Problem

给定 n，对于 $1 \sim n$ 中的每个 i，求 i 的质因子个数。

质因子个数

Problem

给定 n，对于 $1 \sim n$ 中的每个 i，求 i 的质因子个数。

令 $d(i)$ 表示 i 的质因子个数。则在线性筛的时候，可以直接把质数 p 的 $d(p)$ 设为 1，并在从 i 转移到 pi 时令 $d(pi) = d(i) + 1$。复杂度不变，仍为 $O(n)$。

欧拉 φ 函数

Definition 1.6 (互质)

正整数 a 和 b 互质当且仅当 $\gcd(a, b) = 1$。

Definition 1.7 (欧拉 φ 函数)

令 $\varphi(n)$ 表示 $1 \sim n$ 中与 n 互质的数的个数。

欧拉 φ 函数

Definition 1.6 (互质)

正整数 a 和 b 互质当且仅当 $\gcd(a,b)=1$。

Definition 1.7 (欧拉 φ 函数)

令 $\varphi(n)$ 表示 $1\sim n$ 中与 n 互质的数的个数。

枚举 $1\sim n$ 中的数与 n 的最大公约数，则有 $n=\sum_{d|n}\varphi\left(\frac{n}{d}\right)$。

Theorem 1.2

如果 $\gcd(p,q)=1$，则 $\varphi(pq)=\varphi(p)\varphi(q)$。

Theorem 1.2

如果 $\gcd(p,q)=1$，则 $\varphi(pq)=\varphi(p)\varphi(q)$。

先证明以下引理：对于 $0 \le i,j < pq$，如果 $i \ne j$，则 $(i \bmod p, i \bmod q) \ne (j \bmod p, j \bmod q)$（此处括号表示二元组）。

Theorem 1.2

如果 $\gcd(p,q)=1$，则 $\varphi(pq)=\varphi(p)\varphi(q)$。

先证明以下引理：对于 $0 \le i,j < pq$，如果 $i \ne j$，则 $(i \bmod p, i \bmod q) \ne (j \bmod p, j \bmod q)$（此处括号表示二元组）。

证明：如果后半句成立，则意味着 $p \mid i-j, q \mid i-j$，即 $\operatorname{lcm}(p,q) \mid i-j$。由 $\gcd(p,q)=1$ 可知 $\operatorname{lcm}(p,q)=pq$，因此 $pq \mid i-j$。这与 $i \ne j$ 的前提条件相矛盾。

Theorem 1.2

如果 $\gcd(p,q)=1$，则 $\varphi(pq)=\varphi(p)\varphi(q)$。

先证明以下引理：对于 $0\le i,j<pq$，如果 $i\ne j$，则 $(i \bmod p, i \bmod q)\ne(j \bmod p, j \bmod q)$（此处括号表示二元组）。

证明：如果后半句成立，则意味着 $p\mid i-j, q\mid i-j$，即 $\operatorname{lcm}(p,q)\mid i-j$。由 $\gcd(p,q)=1$ 可知 $\operatorname{lcm}(p,q)=pq$，因此 $pq\mid i-j$。这与 $i\ne j$ 的前提条件相矛盾。

由于不同的 $(i \bmod p, i \bmod q)$ 只有 pq 个，因此 $[0,pq)$ 的每一个整数都对应一个 (i,j)。又因为 $\gcd(a,pq)=1$ 当且仅当 $\gcd(a,p)=1$ 且 $\gcd(a,q)=1$，因此满足 $\gcd(a,pq)=1$ 的 a 与满足 $\gcd(i,p)=1,\gcd(j,q)=1$ 的 (i,j) 一一对应，即方案数为 $\varphi(p)\varphi(q)$。

φ 函数计算公式

对于质数 p^k，$\gcd(a, p^k) = 1$ 当且仅当 a 不是 p 的倍数，因此 $\varphi(p^k) = (p-1)p^{k-1}$。

φ 函数计算公式

对于质数 p^k，$\gcd(a, p^k) = 1$ 当且仅当 a 不是 p 的倍数，因此 $\varphi(p^k) = (p-1)p^{k-1}$。

因此，如果 $n = \prod_{i=1}^{n} p_i^{q_i}$，则有 $\varphi(n) = \prod_{i=1}^{n} (p_i - 1)p_i^{q_i - 1}$。

P10031 「Cfz Round 3」Xor with Gcd

T 次询问，每次询问给定一个整数 n。你需要求出 $\gcd(1,n) \oplus \gcd(2,n) \oplus \cdots \oplus \gcd(n,n)$ 的值。其中 $\oplus$ 表示按位异或。

$1 \le T \le 100$，$1 \le n \le 10^{18}$。

对于 $d \mid n$，满足 $\gcd(i, n) = d$ 的 $1 \sim n$ 中的 i 的个数为 $\varphi\left(\frac{n}{d}\right)$。

对于 $d \mid n$，满足 $\gcd(i,n)=d$ 的 $1 \sim n$ 中的 i 的个数为 $\varphi\left(\frac{n}{d}\right)$。

如果 $\varphi\left(\frac{n}{d}\right)$ 是偶数，则 d 不会对最终的异或和产生任何影响。

对于 $d \mid n$，满足 $\gcd(i,n)=d$ 的 $1 \sim n$ 中的 i 的个数为 $\varphi\left(\frac{n}{d}\right)$。

如果 $\varphi\left(\frac{n}{d}\right)$ 是偶数，则 d 不会对最终的异或和产生任何影响。

考虑 $\varphi(x)$ 什么时候可能是偶数，代入之前的 φ 函数计算公式可知，$\varphi(x)$ 是奇数当且仅当 $x=1$ 或 $x=2$。

对于 $d \mid n$，满足 $\gcd(i,n)=d$ 的 $1\sim n$ 中的 i 的个数为 $\varphi\left(\frac{n}{d}\right)$。

如果 $\varphi\left(\frac{n}{d}\right)$ 是偶数，则 d 不会对最终的异或和产生任何影响。

考虑 $\varphi(x)$ 什么时候可能是偶数，代入之前的 φ 函数计算公式可知，$\varphi(x)$ 是奇数当且仅当 $x=1$ 或 $x=2$。

因此当 n 为奇数时答案为 n，否则答案为 $n\oplus\frac{n}{2}$。

线性筛求 φ 函数

Problem

给定 n，对于 $1 \sim n$ 中的每个 i，求 $\varphi(i)$。

线性筛求 φ 函数

Problem

给定 n，对于 $1 \sim n$ 中的每个 i，求 $\varphi(i)$。

考虑线性筛的过程，如果当前的数 p 是一个质数，则令 $\varphi(p) = p - 1$。对于在 i 处筛去 pi 的过程，如果 $\mathrm{low}(i) = p$，则 $\varphi(pi) = p\varphi(i)$，否则 $\varphi(pi) = (p-1)\varphi(i)$。复杂度不变，仍为 $O(n)$。

大家经常会看到“在模意义下...”这种说法。

给定一个 $n-1$ 次多项式 $A(x)$，求一个在 $\bmod x^n$ 意义下的多项式 $B(x)$，使得 $B(x) \equiv (A(x))^k \pmod{x^n}$。

多项式的系数在 $\bmod 998244353$ 的意义下进行运算。

大家经常会看到“在模意义下...”这种说法。

给定一个 $n-1$ 次多项式 $A(x)$，求一个在 $\bmod x^n$ 意义下的多项式 $B(x)$，使得 $B(x) \equiv (A(x))^k \pmod{x^n}$。

多项式的系数在 $\bmod 998244353$ 的意义下进行运算。

模 p 意义的含义，是忽略一个数的具体值，只关心它在对 p 做带余除法后的余数。模 p 意义下的数可以用 $0 \sim p-1$ 来表示，x 在模 p 意义下的值为 i 意味着 $x = i + kp$。很多时候答案的值会很大，此时就会让你输出答案模一个质数意义下的结果。

大家经常会看到“在模意义下...”这种说法。

给定一个 $n-1$ 次多项式 $A(x)$，求一个在 $\bmod x^n$ 意义下的多项式 $B(x)$，使得 $B(x) \equiv (A(x))^k \pmod{x^n}$。

多项式的系数在 $\bmod 998244353$ 的意义下进行运算。

模 p 意义的含义，是忽略一个数的具体值，只关心它在对 p 做带余除法后的余数。模 p 意义下的数可以用 $0 \sim p-1$ 来表示，x 在模 p 意义下的值为 i 意味着 $x = i + kp$。很多时候答案的值会很大，此时就会让你输出答案模一个质数意义下的结果。

同余记号 $a \equiv b \pmod p$ 的含义是 a 和 b 在模 p 意义下相等，该记号不要求 $0 \leq b < p$。

模意义下数的加减乘法都很容易计算，先运算再取模即可。模意义下的除法需要通过方程定义，之后会提到。

模意义下数的加减乘法都很容易计算，先运算再取模即可。模意义下的除法需要通过方程定义，之后会提到。

模质数意义下的运算具有良好的性质，几乎所有有理数具有的性质在模质数意义下都有：例如，加法乘法的交换律、结合律、分配律，以及除了 0 以外都能被除，等等。

模意义下数的加减乘法都很容易计算，先运算再取模即可。模意义下的除法需要通过方程定义，之后会提到。

模质数意义下的运算具有良好的性质，几乎所有有理数具有的性质在模质数意义下都有：例如，加法乘法的交换律、结合律、分配律，以及除了 0 以外都能被除，等等。

模合数意义下的运算性质会稍弱，除了 0 以外还会有一些元素不能被除。如果问题要求在模合数意义下求值，一定要避免除法。

光速乘

对于 $0 \le a, b < p$，显然有 a 和 p 在模 p 意义下的乘积为 $ab \bmod p$。

但是，假如 a 和 b 都是 long long 范围的整数，那么中间值 $a \times b$ 就会超过 long long 范围，导致溢出。当然可以先将其强制转为 __int128 再相乘，但能不能不使用更高级的整数类型呢？

光速乘

对于 $0 \le a, b < p$，显然有 a 和 p 在模 p 意义下的乘积为 $ab \bmod p$。

但是，假如 a 和 b 都是 long long 范围的整数，那么中间值 $a \times b$ 就会超过 long long 范围，导致溢出。当然可以先将其强制转为 __int128 再相乘，但能不能不使用更高级的整数类型呢？

为解决这个问题，我们有一种被称作“光速乘”的方法：

```
ll times(ll a,ll b,ll c){
    ull t=(long double)a*b/c+0.5;
    ll ans=(ull)a*b-t*c;
    if(ans<0) ans+=c;
    return ans;
}
```

模意义下的除法通过方程定义，无法直接计算。为更好地计算除法，我们需要引出逆元的概念。

模意义下的除法通过方程定义，无法直接计算。为更好地计算除法，我们需要引出逆元的概念。

Definition 2.1 (逆元)

对于 $1 \le a < p$ 的正整数 a，a 在模 p 意义下的逆元是方程 $ax \equiv 1 \pmod{p}$ 的唯一解 x，常记作 $a^{-1} \pmod{p}$。

模意义下的除法通过方程定义，无法直接计算。为更好地计算除法，我们需要引出逆元的概念。

Definition 2.1 (逆元)

对于 $1 \le a < p$ 的正整数 a，a 在模 p 意义下的逆元是方程 $ax \equiv 1 \pmod p$ 的唯一解 x，常记作 $a^{-1} \pmod p$。

换句话说，a 的逆元就是指 a 在模意义下的倒数。容易发现 $\left(a^{-1}\right)^{-1} \equiv a \pmod p$，因此逆元关系是相互的。

模意义下的除法通过方程定义，无法直接计算。为更好地计算除法，我们需要引出逆元的概念。

Definition 2.1 (逆元)

对于 $1 \le a < p$ 的正整数 a，a 在模 p 意义下的逆元是方程 $ax \equiv 1 \pmod p$ 的唯一解 x，常记作 $a^{-1} \pmod p$。

换句话说，a 的逆元就是指 a 在模意义下的倒数。容易发现 $\left(a^{-1}\right)^{-1} \equiv a \pmod p$，因此逆元关系是相互的。

对于质数 p，如果 b 在模 p 意义下不为 0，则 $a \div b$ 在模意义下等于 $a \cdot b^{-1}$。

逆元的性质

令 p 是质数。
对于 $a \not\equiv 0, b \not\equiv 0 \pmod{p}$，有 $a^{-1}b^{-1} \equiv (ab)^{-1} \pmod{p}$。

逆元的性质

令 p 是质数。
对于 $a \not\equiv 0, b \not\equiv 0 \pmod p$，有 $a^{-1}b^{-1} \equiv (ab)^{-1} \pmod p$。
对于 $a \not\equiv 0, b \not\equiv 0, a \not\equiv b \pmod p$，有 $a^{-1} \not\equiv b^{-1} \pmod p$。

逆元的性质

令 p 是质数。
对于 $a \not\equiv 0, b \not\equiv 0 \pmod p$，有 $a^{-1}b^{-1} \equiv (ab)^{-1} \pmod p$。
对于 $a \not\equiv 0, b \not\equiv 0, a \not\equiv b \pmod p$，有 $a^{-1} \not\equiv b^{-1} \pmod p$。
对于 $c \not\equiv 0, ac \equiv bc \pmod p$，有 $a \equiv b \pmod p$。

由于逆元关系是相互的，因此我们可以把互为逆元的元素进行配对。对于质数 p，只有 1 和 $p-1$ 的逆元等于自身，$2,3,\cdots,p-2$ 可以恰好分成若干个二元组，满足每组的两个数互为逆元（乘积为 1）。

由于逆元关系是相互的，因此我们可以把互为逆元的元素进行配对。对于质数 p，只有 1 和 $p-1$ 的逆元等于自身，$2,3,\cdots,p-2$ 可以恰好分成若干个二元组，满足每组的两个数互为逆元（乘积为 1）。

Theorem 2.1 (威尔逊定理)

对于质数 p，有 $1\times 2\times\cdots\times(p-1)\equiv -1 \pmod p$。

Theorem 2.2 (费马小定理)

对于质数 p 和任意 $a \not\equiv 0 \pmod p$，有 $a^{p-1} \equiv 1 \pmod p$。

Theorem 2.2 (费马小定理)

对于质数 p 和任意 $a \not\equiv 0 \pmod p$，有 $a^{p-1} \equiv 1 \pmod p$。

Proof

对于 $i \neq j$，有 $ai \neq aj$。

因此 $a \times 1, a \times 2, \cdots, a \times (p-1)$ 在模 p 意义下是互不相同的 $p-1$ 个数，且均不为 0。这意味着模 p 意义下 $a \times 1, a \times 2, \cdots, a \times (p-1)$ 在排序后即为 $1, 2, \cdots, p-1$，可得：

$$1 \times 2 \times \cdots \times (p-1) \equiv (a \times 1) \times (a \times 2) \times \cdots \times (a \times (p-1)) \pmod p$$

两边同除 $1 \times 2 \times \cdots \times (p-1)$ 即得到 $a^{p-1} \equiv 1 \pmod p$。

快速幂求逆元

由费马小定理可知，对于质数 p 和 $a \not\equiv 0 \pmod p$，有 $a^{-1} \equiv a^{p-2} \pmod p$。因此可以使用快速幂计算 a 的逆元。

快速幂求逆元

由费马小定理可知，对于质数 p 和 $a \not\equiv 0 \pmod p$，有 $a^{-1} \equiv a^{p-2} \pmod p$。
因此可以使用快速幂计算 a 的逆元。
如果 p 是合数，对于满足 $\gcd(a, p) = 1$ 的数 a，有 $a^{-1} \equiv a^{\varphi(p)-1} \pmod p$。
该公式是由欧拉定理得到的，欧拉定理可以视为费马小定理在合数时的推广。

Theorem 2.3 (欧拉定理)

对于正整数 $p \geq 2$ 和任意 a 满足 $\gcd(a,p)=1$，有 $a^{\varphi(p)} \equiv 1 \pmod p$。

Theorem 2.3 (欧拉定理)

对于正整数 $p \geq 2$ 和任意 a 满足 $\gcd(a, p) = 1$，有 $a^{\varphi(p)} \equiv 1 \pmod p$。

Proof

考虑用费马小定理类似的方式进行证明。

列出所有 $1 \leq i < p$ 且满足 $\gcd(i, p) = 1$ 的正整数 i，设它们分别为 $t_1, t_2, \cdots, t_{\varphi(p)}$。则 $a \cdot t_1, a \cdot t_2, \cdots, a \cdot t_{\varphi(p)}$ 在模 p 意义下是 $t_1, t_2, \cdots, t_{\varphi(p)}$ 的一个重排。

因此 $t_1, t_2, \cdots, t_{\varphi(p)}$ 的乘积与 $a \cdot t_1, a \cdot t_2, \cdots, a \cdot t_{\varphi(p)}$ 的乘积相同，同除 $t_1 \times t_2 \times \cdots \times t_{\varphi(p)}$ 即得 $a^{\varphi(p)} \equiv 1 \pmod p$。

线性预处理逆元

下述方法可以 $O(n)$ 预处理出 $1 \sim n$ 中所有数在模 p 意义下的逆元：

线性预处理逆元

下述方法可以 $O(n)$ 预处理出 $1 \sim n$ 中所有数在模 p 意义下的逆元：
首先 1 的逆元是 1，然后我们考虑按顺序求出 $2, 3, \cdots, n$ 的逆元。

线性预处理逆元

下述方法可以 $O(n)$ 预处理出 $1 \sim n$ 中所有数在模 p 意义下的逆元：
首先 1 的逆元是 1，然后我们考虑按顺序求出 $2, 3, \cdots, n$ 的逆元。
对于 $2 \le i \le n$，让 p 对 i 作带余除法，得到 $p = qi + r$（$r < i$）。

线性预处理逆元

下述方法可以 $O(n)$ 预处理出 $1 \sim n$ 中所有数在模 p 意义下的逆元：
首先 1 的逆元是 1，然后我们考虑按顺序求出 $2, 3, \cdots, n$ 的逆元。
对于 $2 \le i \le n$，让 p 对 i 作带余除法，得到 $p = qi + r$（$r < i$）。
将等式代入模 p 意义下得到 $-qi \equiv r \pmod p$，两边同除 ir 得到 $-\frac{q}{r} \equiv \frac{1}{i}$。

线性预处理逆元

下述方法可以 $O(n)$ 预处理出 $1 \sim n$ 中所有数在模 p 意义下的逆元：
首先 1 的逆元是 1，然后我们考虑按顺序求出 $2, 3, \cdots, n$ 的逆元。
对于 $2 \le i \le n$，让 p 对 i 作带余除法，得到 $p = qi + r$（$r < i$）。
将等式代入模 p 意义下得到 $-qi \equiv r \pmod p$，两边同除 ir 得到 $-\frac{q}{r} \equiv \frac{1}{i}$。
由于 $r < i$，因此 r 的逆元已知，递推计算即可：

```
inv[i]=1ll*(p-p/i)*inv[mod%i]%mod;
```

P3811 【模板】模意义下的乘法逆元

给定 n, p 求 $1 \sim n$ 中所有整数在模 p 意义下的乘法逆元。
这里 a 模 p 的乘法逆元定义为 $ax \equiv 1 \pmod p$ 的解。
$1 \le n \le 3 \times 10^6$， $n < p < 20000528$。
输入保证 p 为质数。

预处理阶乘逆元

在计算组合数时我们常常需要预处理出 $1 \sim n$ 的阶乘和阶乘逆元，我们可以先递推计算 $1 \sim n$ 的阶乘，然后用快速幂求出 $n!$ 的逆元，再递推计算 $n-1 \sim 1$ 的阶乘逆元。

预处理阶乘逆元

在计算组合数时我们常常需要预处理出 $1 \sim n$ 的阶乘和阶乘逆元，我们可以先递推计算 $1 \sim n$ 的阶乘，然后用快速幂求出 $n!$ 的逆元，再递推计算 $n-1 \sim 1$ 的阶乘逆元。

```
fac[0]=1;
for(int i=1;i<=n;++i) fac[i]=1ll*fac[i-1]*i%p;
ifac[n]=qpow(fac[n],mod-2);
for(int i=n;i>=1;--i) ifac[i-1]=1ll*ifac[i]*i%p;
```

离线求逆元

注意到上一页的阶乘逆元也可用于求出 $1 \sim n$ 的逆元。

离线求逆元

注意到上一页的阶乘逆元也可用于求出 $1 \sim n$ 的逆元。

用类似的思想，我们在 $O(n + \log p)$ 的时间内求出序列 $a_1, a_2, \cdots, a_n$ 中每一个数的逆元。

离线求逆元

注意到上一页的阶乘逆元也可用于求出 $1 \sim n$ 的逆元。

用类似的思想，我们在 $O(n + \log p)$ 的时间内求出序列 $a_1, a_2, \cdots, a_n$ 中每一个数的逆元。

预处理前缀积和前缀积逆元，即可 $O(n)$ 算出每个数的逆元。

P5431【模板】模意义下的乘法逆元 2

给定 n 个正整数 a_i ，求它们在模 p 意义下的乘法逆元。
答案对 p 取模。
$1 \leq n \leq 5 \times 10^6$，$2 \leq k < p \leq 10^9$，$1 \leq a_i < p$，保证 p 为质数。

Problem (等差数列的互质性)

给定 a, b, c，构造 x 使得 $\gcd(ax+b, c) = 1$ 或判断无解。

Problem (等差数列的互质性)

给定 a, b, c，构造 x 使得 $\gcd(ax+b, c) = 1$ 或判断无解。

Solution (等差数列的互质性)

当 $\gcd(a, b, c) \neq 1$ 时，必然不存在解。否则，直接令 $x = \dfrac{c}{\gcd\left((ab)^{\infty}, c\right)}$ 即可。

Corollary 2.1

令 a, b 满足 $\gcd(a, b) = 1$，对于任何 c，都存在 x 使得 $\gcd(ax + b, c)$。

Corollary 2.1

令 a, b 满足 $\gcd(a, b) = 1$，对于任何 c，都存在 x 使得 $\gcd(ax + b, c)$。

Problem (因子提取)

给定 a, b，在 $O(\log \max(a, b))$ 的时间内求出 $\gcd(a^\infty, b)$。
$a, b \leq 10^{18}$。

Corollary 2.1

令 a, b 满足 $\gcd(a, b) = 1$，对于任何 c，都存在 x 使得 $\gcd(ax + b, c)$。

Problem (因子提取)

给定 a, b，在 $O(\log \max(a, b))$ 的时间内求出 $\gcd(a^\infty, b)$。
$a, b \leq 10^{18}$。

Solution (因子提取)

先计算 $a^{\log_2(b)} \bmod b$，然后求 $\gcd\left(a^{\log_2(b)} \bmod b, b\right)$。

Problem (模合数的归一化)

给定 a, p，找到一个数 x 满足 $ax \equiv \gcd(a, p) \pmod{p}$，且 $\gcd(x, p) = 1$。

Problem (模合数的归一化)

给定 a, p，找到一个数 x 满足 $ax \equiv \gcd(a,p) \pmod{p}$，且 $\gcd(x,p)=1$。

Solution (模合数的归一化)

直接求逆就可以得到一个满足 $\gcd\left(x, \frac{p}{\gcd(a,p)}\right)=1$ 的解 x，但这不一定满足 $\gcd(x,p)=1$。

Problem (模合数的归一化)

给定 a, p，找到一个数 x 满足 $ax \equiv \gcd(a, p) \pmod{p}$，且 $\gcd(x, p) = 1$。

Solution (模合数的归一化)

直接求逆就可以得到一个满足 $\gcd\left(x, \frac{p}{\gcd(a,p)}\right) = 1$ 的解 x，但这不一定满足 $\gcd(x, p) = 1$。

考虑先求出任意一个满足 $ax \equiv \gcd(a, p) \pmod{p}$ 的解 x_0，容易发现 $x_0 + t\frac{p}{\gcd(a,p)}$ 都是一个解。因此，我们可以借助之前的结论，求出一个 t 使得 $\gcd\left(x_0 + t\frac{p}{\gcd(a,p)}, p\right) = 1$。由于 $\gcd\left(x_0, \frac{p}{\gcd(a,p)}, p\right) = 1$，故必然存在解。

裴蜀定理

Theorem 3.1 (裴蜀定理)

对于正整数 a, b，定义集合 $S = \{ai + bj | i, j \in \mathbb{Z}\}$。则 S 中的最小正整数等于 $\gcd(a, b)$。

裴蜀定理

Theorem 3.1 (裴蜀定理)

对于正整数 a, b，定义集合 $S = \{ai + bj | i, j \in \mathbb{Z}\}$。则 S 中的最小正整数等于 $\gcd(a, b)$。

证明：设 S 中的最小正整数为 d，其具有表示 $d = ai + bj$。则 $\gcd(a, b)$ 显然是 d 的因子，即 $\gcd(a, b) \le d$。之后我们会在扩展欧几里得算法中构造一组系数 i', j' 使得 $\gcd(a, b) = ai' + bj'$，这意味着 $\gcd(a, b) \in S$，因此 $\gcd(a, b) = d$。

P4549 【模板】裴蜀定理

给定一个包含 n 个元素的整数序列 A，记作 $A_1, A_2, A_3, \ldots, A_n$。

求另一个包含 n 个元素的待定整数序列 X，记 $S = \sum_{i=1}^{n} A_i \times X_i$，使得 $S > 0$ 且 S 尽可能的小。

$1 \leq n \leq 20$，$|A_i| \leq 10^5$，且 A 序列不全为 0。

根据裴蜀定理，$\{ai+bj|i,j\in\mathbb{Z}\}=\{\gcd(a,b)i|i\in\mathbb{Z}\}$。

根据裴蜀定理，$\{ai+bj|i,j\in\mathbb{Z}\}=\{\gcd(a,b)i|i\in\mathbb{Z}\}$。
因此，
$$\{ai+bj+ck|i,j,k\in\mathbb{Z}\}=\{\gcd(a,b)i+cj|i,j\in\mathbb{Z}\}=\{\gcd(a,b,c)i|i\in\mathbb{Z}\}。$$
故本题答案等于 $\gcd(|A_1|,|A_2|,\cdots,|A_n|)$。

辗转相除法

考虑以下计算 $\gcd(a,b)$ 的算法：

$$\gcd(a,b)=\begin{cases}a & b=0\\ \gcd(b, a \bmod b) & b\neq 0\end{cases}$$

辗转相除法

考虑以下计算 $\gcd(a, b)$ 的算法：

$$\gcd(a, b) = \begin{cases} a & b = 0 \\ \gcd(b, a \bmod b) & b \neq 0 \end{cases}$$

以上算法被称为欧几里得算法，或辗转相除法。由于 $\gcd(a, b) = \gcd(a, b - a)$，因此上述算法的正确性是显然的。

辗转相除法

考虑以下计算 $\gcd(a,b)$ 的算法：

$$\gcd(a,b)=\begin{cases} a & b=0 \\ \gcd(b, a \bmod b) & b\neq 0 \end{cases}$$

以上算法被称为欧几里得算法，或辗转相除法。由于 $\gcd(a,b)=\gcd(a,b-a)$，因此上述算法的正确性是显然的。

考虑上述算法的复杂度，当 $a<b$ 时，$\gcd(a,b)$ 会在一次递归后转化为 $\gcd(b,a)$；当 $a\geq b$ 时，有 $a \bmod b \leq \min(b, a-b) \leq \frac{a}{2}$。因此，两次操作后 ab 至少会减小至原来的 $\frac{1}{2}$，故复杂度为 $O(\log n)$。

扩展欧几里得算法

假设我们要构造一组 x, y，使得 $xa + yb = \gcd(a, b)$。我们考虑用和欧几里得算法相同的方式进行递归，先递归计算一组解 x', y' 满足 $x'b + y'(a \bmod b) = \gcd(a, b)$，再通过 x', y' 得到 x, y。递归的终止状态是 $b = 0$，此时很容易构造出解 $x = 1, y = 0$。

扩展欧几里得算法

假设我们要构造一组 x, y，使得 $xa + yb = \gcd(a, b)$。我们考虑用和欧几里得算法相同的方式进行递归，先递归计算一组解 x', y' 满足 $x'b + y'(a \bmod b) = \gcd(a, b)$，再通过 x', y' 得到 x, y。递归的终止状态是 $b = 0$，此时很容易构造出解 $x = 1, y = 0$。

对 $x'b + y'(a \bmod b) = \gcd(a, b)$ 变形，使其变为我们想要的形式：

$$\begin{aligned} x'b + y'(a \bmod b) &= \gcd(a, b) \\ x'b + y'\left(a - \left\lfloor \frac{a}{b} \right\rfloor b\right) &= \gcd(a, b) \\ y'a + \left(x' - \left\lfloor \frac{a}{b} \right\rfloor y'\right) b &= \gcd(a, b) \end{aligned}$$

即 $x = y', y = x' - \lfloor \frac{a}{b} \rfloor y'$。

扩展欧几里得算法

假设我们要构造一组 x, y，使得 $xa + yb = \gcd(a, b)$。我们考虑用和欧几里得算法相同的方式进行递归，先递归计算一组解 x', y' 满足 $x'b + y'(a \bmod b) = \gcd(a, b)$，再通过 x', y' 得到 x, y。递归的终止状态是 $b = 0$，此时很容易构造出解 $x = 1, y = 0$。

对 $x'b + y'(a \bmod b) = \gcd(a, b)$ 变形，使其变为我们想要的形式：

$$\begin{aligned} x'b + y'(a \bmod b) &= \gcd(a, b) \\ x'b + y'\left(a - \left\lfloor \frac{a}{b} \right\rfloor b\right) &= \gcd(a, b) \\ y'a + \left(x' - \left\lfloor \frac{a}{b} \right\rfloor y'\right) b &= \gcd(a, b) \end{aligned}$$

即 $x = y', y = x' - \lfloor \frac{a}{b} \rfloor y'$。

以上算法被称为扩展欧几里得算法，也叫 exgcd。

Theorem 3.2

当 $\gcd(a, b) = 1$ 时，使用扩展欧几里得算法求出的解 x, y 满足 $|x| \le b, |y| \le a$。

Theorem 3.2

当 $\gcd(a, b) = 1$ 时，使用扩展欧几里得算法求出的解 x, y 满足 $|x| \le b, |y| \le a$。

证明：可以使用归纳法，假设 $|x'| \le a \bmod b, |y'| \le b$，则 $|x| = |y'| \le b$，$|y| = |x' - \lfloor \frac{a}{b} \rfloor y'| \le |x'| + \lfloor \frac{a}{b} \rfloor |y'| \le a \bmod b + \lfloor \frac{a}{b} \rfloor b = a$。

Theorem 3.2

当 $\gcd(a, b) = 1$ 时，使用扩展欧几里得算法求出的解 x, y 满足 $|x| \le b, |y| \le a$。

证明：可以使用归纳法，假设 $|x'| \le a \bmod b, |y'| \le b$，则 $|x| = |y'| \le b$，$|y| = |x' - \lfloor \frac{a}{b} \rfloor y'| \le |x'| + \lfloor \frac{a}{b} \rfloor |y'| \le a \bmod b + \lfloor \frac{a}{b} \rfloor b = a$。

因此，如果我们在调用 exgcd 函数前，先将 a, b 分别除以 $\gcd(a, b)$，则无需在 exgcd 函数中使用更高一级整数。

二元一次不定方程

任意的二元一次不定方程都可以使用 exgcd 算法求解。

二元一次不定方程

任意的二元一次不定方程都可以使用 exgcd 算法求解。

具体地，对于方程 $ax+by=c$，先计算 $d=\gcd(a,b)$，并判断 c 是否是 d 的倍数，如果不是则无解。否则，令 $a'=\frac{a}{d}, b'=\frac{b}{d}, c'=\frac{c}{d}$，将方程转化为 $a'x+b'y=c'$。

二元一次不定方程

任意的二元一次不定方程都可以使用 exgcd 算法求解。

具体地，对于方程 $ax+by=c$，先计算 $d=\gcd(a,b)$，并判断 c 是否是 d 的倍数，如果不是则无解。否则，令 $a'=\frac{a}{d}, b'=\frac{b}{d}, c'=\frac{c}{d}$，将方程转化为 $a'x+b'y=c'$。

之后，使用 exgcd 算法求出一组解 x_0, y_0 满足 $a'x_0+b'y_0=1$，那么 $c'x_0$ 和 $c'y_0$ 即为原方程的一组解。

二元一次不定方程

任意的二元一次不定方程都可以使用 exgcd 算法求解。

具体地，对于方程 $ax+by=c$，先计算 $d=\gcd(a,b)$，并判断 c 是否是 d 的倍数，如果不是则无解。否则，令 $a'=\frac{a}{d}, b'=\frac{b}{d}, c'=\frac{c}{d}$，将方程转化为 $a'x+b'y=c'$。

之后，使用 exgcd 算法求出一组解 x_0, y_0 满足 $a'x_0+b'y_0=1$，那么 $c'x_0$ 和 $c'y_0$ 即为原方程的一组解。

容易发现，假如 (x,y) 是一组解，那么 $(x-b', y+a')$ 也是一组解。可以利用这个性质缩小解的绝对值。

exgcd 求逆元

对于正整数 a, p 满足 $\gcd(a, p) = 1$，$0 \le a < p$。我们可以使用 exgcd 算法求出一个 x 满足 $ax \equiv 1 \pmod p$。

exgcd 求逆元

对于正整数 a, p 满足 $\gcd(a,p)=1$，$0 \le a < p$。我们可以使用 exgcd 算法求出一个 x 满足 $ax \equiv 1 \pmod p$。

把上式从模意义下的等式转为常规等式，得到 $ax + kp = 1$，对其使用 exgcd 算法即可。

如果是求逆元，则额外要求 $0 \le x < p$，那么把 x 对 p 取模即可。

P4777 【模板】扩展中国剩余定理（EXCRT）

给定 n 组非负整数 a_i, b_i ，求解关于 x 的方程组的最小非负整数解。

$$\begin{cases} x \equiv b_1 \pmod{a_1} \\ x \equiv b_2 \pmod{a_2} \\ \dots \\ x \equiv b_n \pmod{a_n} \end{cases}$$

$1 \le n \le 10^5$，$1 \le b_i, a_i \le 10^{12}$，保证所有 a_i 的最小公倍数不超过 10^{18}。

题解

对于方程 $x \equiv b_1 \pmod{a_1}$ 和 $x \equiv b_2 \pmod{a_2}$，如果它们有公共解 x，那么 y 是它们的公共解当且仅当 $\operatorname{lcm}(a_1, a_2) \mid x - y$。因此我们可以对它们进行合并，它们要么无公共解，要么等价于一个方程 $x \equiv c \pmod{\operatorname{lcm}(a_1, a_2)}$。

题解

对于方程 $x \equiv b_1 \pmod{a_1}$ 和 $x \equiv b_2 \pmod{a_2}$，如果它们有公共解 x，那么 y 是它们的公共解当且仅当 $\operatorname{lcm}(a_1, a_2) \mid x - y$。因此我们可以对它们进行合并，它们要么无公共解，要么等价于一个方程 $x \equiv c \pmod{\operatorname{lcm}(a_1, a_2)}$。

尝试求解合并后的方程形式。先把模意义等式转为普通等式，即

$$\begin{cases} x = b_1 + k_1 a_1 \\ x = b_2 + k_2 a_2 \end{cases}。$$

暂时忽略 x 得到 $b_1 + k_1 a_1 = b_2 + k_2 a_2$，注意此时的未知数是 k_1, k_2。

暂时忽略 x 得到 $b_1 + k_1 a_1 = b_2 + k_2 a_2$，注意此时的未知数是 k_1, k_2。

对上式移项得到 $k_1 a_1 - k_2 a_2 = b_2 - b_1$，使用 exgcd 即可求出一组 k_1, k_2。代入 $x = b_1 + k_1 a_1$ 即可得到合并后的方程 $x \equiv b_1 + k_1 a_1 \pmod{\operatorname{lcm}(a_1, a_2)}$。

暂时忽略 x 得到 $b_1 + k_1 a_1 = b_2 + k_2 a_2$，注意此时的未知数是 k_1, k_2。

对上式移项得到 $k_1 a_1 - k_2 a_2 = b_2 - b_1$，使用 exgcd 即可求出一组 k_1, k_2。代入 $x = b_1 + k_1 a_1$ 即可得到合并后的方程 $x \equiv b_1 + k_1 a_1 \pmod{\operatorname{lcm}(a_1, a_2)}$。

因此可以把所有方程合并为一个方程或判定无解，单个方程的最小非负整数解很容易得到。

等差数列环模型

我们可以把模 m 意义下的 m 个数看成 m 个点，如果相差 1 的两个数之间有边，那么它们就构成了一个环。

等差数列环模型

我们可以把模 m 意义下的 m 个数看成 m 个点，如果相差 1 的两个数之间有边，那么它们就构成了一个环。

有时候题目中会给出 k，并且频繁地出现加 k 操作，此时我们可以在两个相差 k 的数之间连边，一共会形成 $\gcd(k, m)$ 个环。

等差数列环模型

我们可以把模 m 意义下的 m 个数看成 m 个点，如果相差 1 的两个数之间有边，那么它们就构成了一个环。

有时候题目中会给出 k，并且频繁地出现加 k 操作，此时我们可以在两个相差 k 的数之间连边，一共会形成 $\gcd(k,m)$ 个环。

注意这里的环不需要真的建出来，先在 $[0,k)$ 中枚举环的起点，然后不断加 k 就可以了。也可以通过取模和求逆直接定位一个数所在环的编号和位置。

等差数列环模型

我们可以把模 m 意义下的 m 个数看成 m 个点，如果相差 1 的两个数之间有边，那么它们就构成了一个环。

有时候题目中会给出 k，并且频繁地出现加 k 操作，此时我们可以在两个相差 k 的数之间连边，一共会形成 $\gcd(k, m)$ 个环。

注意这里的环不需要真的建出来，先在 $[0, k)$ 中枚举环的起点，然后不断加 k 就可以了。也可以通过取模和求逆直接定位一个数所在环的编号和位置。

该模型被我称为等差数列环模型（名字是随便取的）。

CF819D Mister B and Astronomers

给定 $T, a_0 \sim a_{n-1}$。构造数列 b，$b_0 = 0, b_i = (b_{i-1} + a_{i \bmod n}) \bmod T$。对 $0 \le i < n$，求有多少种数在 b 中的第一次出现的位置模 n 等于 i。

$n \le 2 \times 10^5, T, a_i \le 10^9$。

令 $c=\sum_{i=0}^{n-1} a_i$，则有 $b_{i+n} \equiv b_i + c \pmod{T}$。

令 $c=\sum_{i=0}^{n-1} a_i$，则有 $b_{i+n} \equiv b_i + c \pmod{T}$。

因此可以把 c 当作边长构建等差数列环模型，i 的答案即为从 b_i 开始，每次 $+c$，在遇到其它 b_j 之前一共经过了多少个点。

CF1575C Cyclic Sum

给定 $a_0 \sim a_{n-1}, m$ 和质数 k。环形序列 b 由 m 个 a 拼接而成。求 b 有几个区间的和是 k 的倍数。

$n, m, k, a_i \leq 2 \times 10^5$。

枚举区间的左端点，由于左端点为 i 的答案与左端点为 $i+n$ 的答案相同，故只需要考虑左端点为 $0 \sim n-1$ 的情况即可。

枚举区间的左端点，由于左端点为 i 的答案与左端点为 $i+n$ 的答案相同，故只需要考虑左端点为 $0 \sim n-1$ 的情况即可。

令 $c \equiv \sum_{i=0}^{n-1} a_i \pmod{k}$。当左端点为 0 时，区间 $[0, i+n]$ 的和恰好比 $[0, i]$ 的和多 c。因此，如果以 c 为边长构建等差数列环模型，所有以 0 为左端点的区间长度可以通过 n 次区间加得到。

枚举区间的左端点，由于左端点为 i 的答案与左端点为 $i+n$ 的答案相同，故只需要考虑左端点为 $0\sim n-1$ 的情况即可。

令 $c\equiv\sum_{i=0}^{n-1}a_i\pmod{k}$。当左端点为 0 时，区间 $[0,i+n]$ 的和恰好比 $[0,i]$ 的和多 c。因此，如果以 c 为边长构建等差数列环模型，所有以 0 为左端点的区间长度可以通过 n 次区间加得到。

如果维护了所有的区间和，那么从左端点为 i 转移到左端点为 $i+1$ 只需要删除一个区间再加入一个区间，这是容易维护的。

中国剩余定理

对于以下方程，如果有 $a_1, a_2, \cdots a_n$ 两两互质，则必然存在模 $\prod_{i=1}^{n} a_i$ 意义下的唯一解 x：

$$\begin{cases} x \equiv b_1 \pmod{a_1} \\ x \equiv b_2 \pmod{a_2} \\ \dots \\ x \equiv b_n \pmod{a_n} \end{cases}$$

中国剩余定理形式化地给出了这个解 x。

具体地，先构造 B_i 满足 $B_i \bmod a_i = b_i$，$B_i \bmod a_j = 0$（$i \neq j$），再令 $x = B_1 + B_2 + \cdots + B_n$。

具体地，先构造 B_i 满足 $B_i \bmod a_i = b_i$，$B_i \bmod a_j = 0$（$i \neq j$），再令 $x = B_1 + B_2 + \cdots + B_n$。

考虑构造单个 B_i，由于 $B_i \bmod a_j = 0$（$i \neq j$），因此可以设 $B_i = k \prod_{j \neq i} a_j$。此时唯一的约束是 $k \prod_{j \neq i} a_j \equiv b_i \pmod{a_i}$，那么只要令 $k = b_i \prod_{j \neq i} a_j^{-1} \bmod a_i$ 即可。

具体地，先构造 B_i 满足 $B_i \bmod a_i = b_i$，$B_i \bmod a_j = 0$（$i \neq j$），再令 $x = B_1 + B_2 + \cdots + B_n$。

考虑构造单个 B_i，由于 $B_i \bmod a_j = 0$（$i \neq j$），因此可以设 $B_i = k \prod_{j \neq i} a_j$。此时唯一的约束是 $k \prod_{j \neq i} a_j \equiv b_i \pmod{a_i}$，那么只要令 $k = b_i \prod_{j \neq i} a_j^{-1} \bmod a_i$ 即可。

把上面求出的东西代入，得到 $x = \sum_{i=1}^{n} \left(b_i \prod_{j \neq i} a_j^{-1} \bmod a_i \right) \prod_{j \neq i} a_j$。这就是中国剩余定理的内容。

虽然中国剩余定理的求根公式几乎完全没用，但这个定理本身揭示了很重要的一点：

虽然中国剩余定理的求根公式几乎完全没用，但这个定理本身揭示了很重要的一点：

对于 $n=\prod_{i=1}^{k} p_i^{\alpha_i}$，一个模 n 意义下的数可以看成一个 k 维向量，向量的第 i 位是一个模 $p_i^{\alpha_i}$ 意义下的数。

原数的加法就对应向量的加法，原数的乘法对应向量的按位相乘。

P5330 [SNOI2019] 数论

给出正整数 P, Q, T，大小为 n 的整数集 A 和大小为 m 的整数集 B，请你求出：

$$\sum_{i=0}^{T-1} [(i \bmod P) \in A \land (i \bmod Q) \in B]$$

换言之，就是问有多少个小于 T 的非负整数 x 满足：x 除以 P 的余数属于 A 且 x 除以 Q 的余数属于 B。

对于所有数据，$1 \le n, m \le 10^6, 1 \le P, Q \le 10^6, 1 \le T \le 10^{18}$。

首先先把 A, B 中的数按照模 $\gcd(P, Q)$ 的余数分类，A 和 B 中模 $\gcd(P, Q)$ 不同的数显然不会相互影响。对于同一类的 A, B，我们可以进行一些处理，减去模 $\gcd(P, Q)$ 的余数然后将 a, b 同除 $\gcd(P, Q)$，使问题转化为 $\gcd(P, Q) = 1$ 时的原问题。

首先先把 A, B 中的数按照模 $\gcd(P, Q)$ 的余数分类，A 和 B 中模 $\gcd(P, Q)$ 不同的数显然不会相互影响。对于同一类的 A, B，我们可以进行一些处理，减去模 $\gcd(P, Q)$ 的余数然后将 a, b 同除 $\gcd(P, Q)$，使问题转化为 $\gcd(P, Q) = 1$ 时的原问题。

对于同一类的 A, B，设 A 中的元素为 a，B 中的元素为 b。由中国剩余定理可知，a, b 对应的数 x 满足 $x \equiv ak_1 + bk_2 \pmod{\operatorname{lcm}(n, m)}$，其中 $k_1 = (b^{-1} \bmod a)b, k_2 = (a^{-1} \bmod b)a$。

首先先把 A, B 中的数按照模 $\gcd(P, Q)$ 的余数分类，A 和 B 中模 $\gcd(P, Q)$ 不同的数显然不会相互影响。对于同一类的 A, B，我们可以进行一些处理，减去模 $\gcd(P, Q)$ 的余数然后将 a, b 同除 $\gcd(P, Q)$，使问题转化为 $\gcd(P, Q) = 1$ 时的原问题。

对于同一类的 A, B，设 A 中的元素为 a，B 中的元素为 b。由中国剩余定理可知，a, b 对应的数 x 满足 $x \equiv ak_1 + bk_2 \pmod{\operatorname{lcm}(n, m)}$，其中 $k_1 = (b^{-1} \bmod a)b, k_2 = (a^{-1} \bmod b)a$。

对 A 中的所有元素乘 k_1，B 中的所有元素乘 k_2，问题变为：存在多少 $[0, T]$ 之间的整数 x，满足存在 $a \in A, b \in B$ 使得 $x \equiv a + b \pmod{nm}$。这可以通过枚举 A 中的数，并在 B 里二分来得到。

SOJ1726 鸽子做加法（核心）

给定两个循环节长度互质的二进制纯循环小数 $0.\dot{a}_0 a_1 \dots \dot{a}_{n-1}$ 和 $0.\dot{b}_0 b_1 \dots \dot{b}_{m-1}$，求它们的按位异或（显然也是一个纯循环小数，属于有理数）在模 998244353 意义下的值。

$1 \le n, m \le 10^6$，$\gcd(n, m) = 1$。

题解

Lemma 3.1

对于 p 进制下的纯循环小数 $0.\dot{a}_0a_1\ldots\dot{a}_{n-1}$，其分数形式的值为 $\dfrac{\sum\limits_{i=0}^{n-1}a_ip^{n-1-i}}{p^n-1}$。

题解

Lemma 3.1

对于 p 进制下的纯循环小数 $0.\dot{a}_0a_1\ldots\dot{a}_{n-1}$，其分数形式的值为 $\dfrac{\sum\limits_{i=0}^{n-1}a_ip^{n-1-i}}{p^n-1}$。

证明：设其值为 x，则有 $p^nx=\sum\limits_{i=0}^{n-1}a_ip^{n-1-i}+x$，解方程即得到上式。

先通过恒等式 $a+b=a\oplus b+a\&b$ 将求按位异或转化为求按位与。此时，如果输入的两个循环节中分别有 p 个 1 和 q 个 1，则答案的循环节中含有 pq 个 1。

先通过恒等式 $a+b=a\oplus b+a\&b$ 将求按位异或转化为求按位与。此时，如果输入的两个循环节中分别有 p 个 1 和 q 个 1，则答案的循环节中含有 pq 个 1。

对于任意满足 $a_i=1$ 和 $b_j=1$ 的一组 i,j，它们唯一对应了答案循环节中的一个 1。设这个 1 的位置是 x，则有 $x\equiv i\pmod n$，$x\equiv j\pmod m$。由中国剩余定理可知，$x=(ik_1+jk_2)\bmod nm$，其中 $k_1=(m^{-1}\bmod n)m,k_2=(n^{-1}\bmod m)n$。

先通过恒等式 $a+b=a\oplus b+a\&b$ 将求按位异或转化为求按位与。此时，如果输入的两个循环节中分别有 p 个 1 和 q 个 1，则答案的循环节中含有 pq 个 1。

对于任意满足 $a_i=1$ 和 $b_j=1$ 的一组 i,j，它们唯一对应了答案循环节中的一个 1。设这个 1 的位置是 x，则有 $x\equiv i \pmod n$，$x\equiv j \pmod m$。由中国剩余定理可知，$x=(ik_1+jk_2) \bmod nm$，其中 $k_1=(m^{-1} \bmod n)m, k_2=(n^{-1} \bmod m)n$。

对于所有满足 $a_i=1$ 的 i，我们把 $ik_1 \bmod nm$ 加入序列 a'；对于所有满足 $b_i=1$ 的 i，我们把 $ik_2 \bmod nm$ 加入序列 b'。此时，答案等于

$$\frac{\sum_{i=0}^{p-1}\sum_{j=0}^{q-1} 2^{nm-1-(a'_i+b'_j \bmod nm)}}{2^{nm}-1}。$$

这等价于计算 $\sum_{i=0}^{p-1}\sum_{j=0}^{q-1}\left(\frac{1}{2}\right)^{a_i'+b_j' \bmod nm}$。

这等价于计算 $\sum_{i=0}^{p-1}\sum_{j=0}^{q-1}\left(\frac{1}{2}\right)^{a'_i+b'_j \bmod nm}$。

注意到 $a'_i+b'_j \bmod nm$ 等于 $a'_i+b'_j$ 或 $a'_i+b'_j-nm$。因此，我们可以将序列 a' 和 b' 分别排序，之后按顺序枚举 a' 中的元素，满足 $a'_i+b'_j \bmod nm = a'_i+b'_j$ 的恰好是 b' 的一段前缀，则 $\sum_{j=0}^{t}\left(\frac{1}{2}\right)^{a'_i+b'_j}=\left(\frac{1}{2}\right)^{a'_i}\sum_{j=0}^{t}\left(\frac{1}{2}\right)^{b'_j}$，可以通过预处理前缀和来快速计算。另一部分同样可以借助预处理快速计算。

这等价于计算 $\sum_{i=0}^{p-1}\sum_{j=0}^{q-1}\left(\frac{1}{2}\right)^{a_i'+b_j' \bmod nm}$。

注意到 $a_i'+b_j' \bmod nm$ 等于 $a_i'+b_j'$ 或 $a_i'+b_j'-nm$。因此，我们可以将序列 a' 和 b' 分别排序，之后按顺序枚举 a' 中的元素，满足 $a_i'+b_j' \bmod nm = a_i'+b_j'$ 的恰好是 b' 的一段前缀，则 $\sum_{j=0}^{t}\left(\frac{1}{2}\right)^{a_i'+b_j'} = \left(\frac{1}{2}\right)^{a_i'}\sum_{j=0}^{t}\left(\frac{1}{2}\right)^{b_j'}$，可以通过预处理前缀和来快速计算。另一部分同样可以借助预处理快速计算。

$a_i'+b_j' \bmod nm = a_i'+b_j'$ 和 $a_i'+b_j' \bmod nm = a_i'+b_j'-nm$ 的分界点可以通过双指针得到，总复杂度 $O(n\log n)$，瓶颈在于排序。

试除法：$O(\sqrt{n})$ 判定一个数是否是质数。

试除法：$O(\sqrt{n})$ 判定一个数是否是质数。

线性筛：$O(V)$ 预处理，对于 $\leq V$ 的数可以 $O(1)$ 判定其素性，否则对于 $\leq V^2$ 的数可以 $O\left(\frac{\sqrt{n}}{\log n}\right)$ 判定。

我们可以尝试使用费马小定理来检验素数，即，对于给定的待检验数 p，随机一个数 a，计算 $a^{p-1} \bmod p$。如果 $a^{p-1} \not\equiv 1 \pmod p$，则 p 不是质数。这种检验质数的方式也被称为费马判定。

我们可以尝试使用费马小定理来检验素数，即，对于给定的待检验数 p，随机一个数 a，计算 $a^{p-1} \bmod p$。如果 $a^{p-1} \not\equiv 1 \pmod p$，则 p 不是质数。这种检验质数的方式也被称为费马判定。

但很遗憾，这样的算法不可行。存在一类合数 n，其对于任意满足 $\gcd(a, n) = 1$ 的整数 a 都有 $a^{n-1} \equiv 1 \pmod n$ 成立。这类数被称为卡迈克尔数，上述算法几乎无法检验出它们。

我们可以尝试使用费马小定理来检验素数，即，对于给定的待检验数 p，随机一个数 a，计算 $a^{p-1} \bmod p$。如果 $a^{p-1} \not\equiv 1 \pmod p$，则 p 不是质数。这种检验质数的方式也被称为费马判定。

但很遗憾，这样的算法不可行。存在一类合数 n，其对于任意满足 $\gcd(a, n) = 1$ 的整数 a 都有 $a^{n-1} \equiv 1 \pmod n$ 成立。这类数被称为卡迈克尔数，上述算法几乎无法检验出它们。

因此我们要对费马判定进行加强，使其能更好地判别质数。

Lemma 4.1

对于任意质数 p，如果 $x^2 \equiv 1 \pmod p$ 成立，则 $x \equiv 1$ 或 $x \equiv -1$。

Lemma 4.1

对于任意质数 p，如果 $x^2 \equiv 1 \pmod{p}$ 成立，则 $x \equiv 1$ 或 $x \equiv -1$。

证明：由于 $(x-1)(x+1) \equiv 0 \pmod{p}$，故显然不存在其它解。

Lemma 4.1

对于任意质数 p，如果 $x^2 \equiv 1 \pmod p$ 成立，则 $x \equiv 1$ 或 $x \equiv -1$。

证明：由于 $(x-1)(x+1) \equiv 0 \pmod p$，故显然不存在其它解。

我们可以借助这一引理加强费马判定，当 $a^{p-1} \equiv 1 \pmod p$ 成立时，如果 $p-1$ 是偶数，则计算 $a^{\frac{p-1}{2}}$。如果 $a^{\frac{p-1}{2}} \not\equiv \pm 1$ 则 p 不是质数，否则如果 $a^{\frac{p-1}{2}} \equiv 1$ 且 $\dfrac{p-1}{2}$ 仍为偶数，则递归进行判定。

Miller-Rabin 质数判定法

1. 取底数 a，设 p 为待判定整数，$p-1=2^c \cdot s$ (s 为奇数)。

Miller-Rabin 质数判定法

1. 取底数 a，设 p 为待判定整数，$p-1=2^c \cdot s$ (s 为奇数)。
2. 令 $v \leftarrow a^s \bmod p, j \leftarrow 1$。

Miller-Rabin 质数判定法

1. 取底数 a，设 p 为待判定整数，$p-1=2^c\cdot s$ (s 为奇数)。
2. 令 $v\leftarrow a^s \bmod p, j\leftarrow 1$。
3. 令 $u\leftarrow v, v\leftarrow v^2 \bmod p$。

Miller-Rabin 质数判定法

1. 取底数 a，设 p 为待判定整数，$p-1=2^c\cdot s$ (s 为奇数)。
2. 令 $v\leftarrow a^s \bmod p, j\leftarrow 1$。
3. 令 $u\leftarrow v, v\leftarrow v^2 \bmod p$。
4. 若 $u\neq 1$ 且 $v=1$，则返回 p 为合数。

Miller-Rabin 质数判定法

1. 取底数 a，设 p 为待判定整数，$p-1=2^c\cdot s$ (s 为奇数)。
2. 令 $v\leftarrow a^s \bmod p, j\leftarrow 1$。
3. 令 $u\leftarrow v, v\leftarrow v^2 \bmod p$。
4. 若 $u\neq 1$ 且 $v=1$，则返回 p 为合数。
5. 若 $j<c$，则令 $j\leftarrow j+1$，回到步骤 3。

Miller-Rabin 质数判定法

1. 取底数 a，设 p 为待判定整数，$p-1=2^c\cdot s$ (s 为奇数)。
2. 令 $v\leftarrow a^s \bmod p, j\leftarrow 1$。
3. 令 $u\leftarrow v, v\leftarrow v^2 \bmod p$。
4. 若 $u\neq 1$ 且 $v=1$，则返回 p 为合数。
5. 若 $j<c$，则令 $j\leftarrow j+1$，回到步骤 3。
6. 若 $v\neq 1$，则返回 p 为合数。

Miller-Rabin 质数判定法

1. 取底数 a，设 p 为待判定整数，$p-1=2^c\cdot s$ (s 为奇数)。
2. 令 $v \leftarrow a^s \bmod p, j \leftarrow 1$。
3. 令 $u \leftarrow v, v \leftarrow v^2 \bmod p$。
4. 若 $u \neq 1$ 且 $v=1$，则返回 p 为合数。
5. 若 $j<c$，则令 $j \leftarrow j+1$，回到步骤 3。
6. 若 $v \neq 1$，则返回 p 为合数。
7. 认为 p 通过以 a 为底的强伪素数测试。

于是，在该算法中，a 的选择是至关重要的。

于是，在该算法中，a 的选择是至关重要的。

事实上，对于某些合数 n 和底 a，n 仍然能通过以 a 为底的强伪素数测试，此时我们称 n 为以 a 为底的强伪素数。

于是，在该算法中，a 的选择是至关重要的。

事实上，对于某些合数 n 和底 a，n 仍然能通过以 a 为底的强伪素数测试，此时我们称 n 为以 a 为底的强伪素数。

Theorem 4.1

对于一个合数 n，它至多只能通过模 n 下 $\frac{1}{4}n$ 个底的强伪素数测试。

于是，在该算法中，a 的选择是至关重要的。

事实上，对于某些合数 n 和底 a，n 仍然能通过以 a 为底的强伪素数测试，此时我们称 n 为以 a 为底的强伪素数。

Theorem 4.1

对于一个合数 n，它至多只能通过模 n 下 $\frac{1}{4}n$ 个底的强伪素数测试。

因此，随机 k 个底数，判断错误的概率就不超过 4^{-k} 了。

上述算法被称为 Miller-Rabin 算法。在通常使用时，由于被检验的 n 不是很大，我们可以利用一些先前总结过的底数列表：

上述算法被称为 Miller-Rabin 算法。在通常使用时，由于被检验的 n 不是很大，我们可以利用一些先前总结过的底数列表：

底数列表	最小的强伪素数
2	2047
$2, 3$	1373653
$2, 7, 61$	$4\,759\,123\,141$
$2, 3, 5, 7, 11$	$2\,152\,302\,898\,747$
$2, 3, 5, 7, 11, 13, 17[, 19]$	$341\,550\,071\,728\,321$
$2, 3, 5, 7, 11, 13, 17, 19, 23[, 29, 31]$	$3\,825\,123\,056\,546\,413\,051$
$2, 325, 9375, 28178, 450775, 9780504, 1795265022$	$> 2^{64}$
$2, 3, 5, 7, 11, 13, 82, 373$	
$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37$	$318\,665\,857\,834\,031\,151\,167\,461$

上述算法被称为 Miller-Rabin 算法。在通常使用时，由于被检验的 n 不是很大，我们可以利用一些先前总结过的底数列表：

底数列表	最小的强伪素数
2	2047
$2, 3$	1373653
$2, 7, 61$	$4\,759\,123\,141$
$2, 3, 5, 7, 11$	$2\,152\,302\,898\,747$
$2, 3, 5, 7, 11, 13, 17[, 19]$	$341\,550\,071\,728\,321$
$2, 3, 5, 7, 11, 13, 17, 19, 23[, 29, 31]$	$3\,825\,123\,056\,546\,413\,051$
$2, 325, 9375, 28178, 450775, 9780504, 1795265022$	$> 2^{64}$
$2, 3, 5, 7, 11, 13, 82, 373$	
$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37$	$318\,665\,857\,834\,031\,151\,167\,461$

表中红色的数为合数，使用时应当注意；灰色的数表示添加与否不影响结果。

上述算法被称为 Miller-Rabin 算法。在通常使用时，由于被检验的 n 不是很大，我们可以利用一些先前总结过的底数列表：

底数列表	最小的强伪素数
2	2047
$2, 3$	1373653
$2, 7, 61$	$4\,759\,123\,141$
$2, 3, 5, 7, 11$	$2\,152\,302\,898\,747$
$2, 3, 5, 7, 11, 13, 17[, 19]$	$341\,550\,071\,728\,321$
$2, 3, 5, 7, 11, 13, 17, 19, 23[, 29, 31]$	$3\,825\,123\,056\,546\,413\,051$
$2, 325, 9375, 28178, 450775, 9780504, 1795265022$	$> 2^{64}$
$2, 3, 5, 7, 11, 13, 82, 373$	
$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37$	$318\,665\,857\,834\,031\,151\,167\,461$

表中红色的数为合数，使用时应当注意；灰色的数表示添加与否不影响结果。更多相关信息可见 `http://miller-rabin.appspot.com/`。

朴素算法

使用线性筛 $O(V)$ 预处理 $1 \sim V$ 之间的质数，则可以 $O\left(\frac{\sqrt{n}}{\log n}\right)$ 对一个 $\leq V^2$ 的数进行质因数分解。

朴素算法

使用线性筛 $O(V)$ 预处理 $1 \sim V$ 之间的质数，则可以 $O\left(\frac{\sqrt{n}}{\log n}\right)$ 对一个 $\leq V^2$ 的数进行质因数分解。

在不进行预处理的情况下，也可以 $O(\sqrt{n})$ 对一个数进行分解。

朴素算法

使用线性筛 $O(V)$ 预处理 $1 \sim V$ 之间的质数，则可以 $O\left(\frac{\sqrt{n}}{\log n}\right)$ 对一个 $\leq V^2$ 的数进行质因数分解。

在不进行预处理的情况下，也可以 $O(\sqrt{n})$ 对一个数进行分解。

计算 $\varphi(n)$，$\mu(n)$ 等值所需的时间与对 n 进行质因数分解所需的时间基本相同。

生日悖论

Lemma 4.2 (生日悖论)

23 个人的房间中，有两个人生日相同的概率超过 $\frac{1}{2}$。

生日悖论

Lemma 4.2 (生日悖论)

23 个人的房间中，有两个人生日相同的概率超过 $\frac{1}{2}$。

生日悖论揭示的内容实际上是：假如有 n 个随机数，那么“n 个数中存在相同的数”事实上表示“所有 $\frac{n(n-1)}{2}$ 对数中有至少一对数相同”。换句话说，“存在两个数相同”实际上对应了 $O(n^2)$ 个随机冲突事件，而不是 $O(n)$ 个。

生日悖论

Lemma 4.2 (生日悖论)

23 个人的房间中，有两个人生日相同的概率超过 $\dfrac{1}{2}$。

生日悖论揭示的内容实际上是：假如有 n 个随机数，那么“n 个数中存在相同的数”事实上表示“所有 $\dfrac{n(n-1)}{2}$ 对数中有至少一对数相同”。换句话说，“存在两个数相同”实际上对应了 $O(n^2)$ 个随机冲突事件，而不是 $O(n)$ 个。

因此，我们可以感性地认为，$O(\sqrt{n})$ 个 $[1,n]$ 间的随机整数中有 $O(1)$ 的概率存在相同的数。

ρ 形序列

对于一个值域大小有限的函数 f 和起始值 x，序列 $x, f(x), f(f(x)), \cdots$ 具有 ρ 形结构。即，从某一位开始，该序列进入循环。特别地，如果 f 是可逆函数，则该序列具有周期。

ρ 形序列

对于一个值域大小有限的函数 f 和起始值 x，序列 $x, f(x), f(f(x)), \cdots$ 具有 ρ 形结构。即，从某一位开始，该序列进入循环。特别地，如果 f 是可逆函数，则该序列具有周期。

如果该序列是周期序列，那它的周期是很好找的。只要从 $f(x)$ 开始，找到第一个与 x 相等的数即可。

ρ 形序列

对于一个值域大小有限的函数 f 和起始值 x，序列 $x, f(x), f(f(x)), \cdots$ 具有 ρ 形结构。即，从某一位开始，该序列进入循环。特别地，如果 f 是可逆函数，则该序列具有周期。

如果该序列是周期序列，那它的周期是很好找的。只要从 $f(x)$ 开始，找到第一个与 x 相等的数即可。

如果该 ρ 形序列不是周期序列，则可以通过 Floyd 判圈法来找到该序列的周期。即，初始时 $y_1 = x, y_2 = x$，之后每一步令 $y_1 \leftarrow f(y_1), y_2 \leftarrow f(f(y_2))$，直到 $y_1 = y_2$ 时停止。

ρ 形序列

对于一个值域大小有限的函数 f 和起始值 x，序列 $x, f(x), f(f(x)), \cdots$ 具有 ρ 形结构。即，从某一位开始，该序列进入循环。特别地，如果 f 是可逆函数，则该序列具有周期。

如果该序列是周期序列，那它的周期是很好找的。只要从 $f(x)$ 开始，找到第一个与 x 相等的数即可。

如果该 ρ 形序列不是周期序列，则可以通过 Floyd 判圈法来找到该序列的周期。即，初始时 $y_1 = x, y_2 = x$，之后每一步令 $y_1 \leftarrow f(y_1), y_2 \leftarrow f(f(y_2))$，直到 $y_1 = y_2$ 时停止。

假如该函数是一个较为随机的函数，则该序列的前 $O(\sqrt{n})$ 项大概率会有相同的元素，即该序列的周期大概率为 $O(\sqrt{n})$。

Pollard-Rho 算法

假设我们要分解一个合数 n，我们考虑找到它的一个非平凡因子 m，然后递归分解 m 和 $\frac{n}{m}$。我们首先要选取一个较为随机的一元函数 f，通常的方法是选择一个常数 $c \in [3, 100]$，然后令 $f(x) = x^2 + c$。

Pollard-Rho 算法

假设我们要分解一个合数 n，我们考虑找到它的一个非平凡因子 m，然后递归分解 m 和 $\frac{n}{m}$。我们首先要选取一个较为随机的一元函数 f，通常的方法是选择一个常数 $c\in[3,100]$，然后令 $f(x)=x^2+c$。

考虑序列 $x, f(x), f(f(x)), \cdots$，根据之前的结论，该序列的周期是 $O(\sqrt{n})$ 的。令 p 表示 n 的最小质因子，那么该序列在模 p 意义下的周期应当是 $O\left(\sqrt{p}\right)$ 的。而且，在大多数情况下，这个周期是不等于其在模 n 意义下的周期的。

Pollard-Rho 算法

假设我们要分解一个合数 n，我们考虑找到它的一个非平凡因子 m，然后递归分解 m 和 $\frac{n}{m}$。我们首先要选取一个较为随机的一元函数 f，通常的方法是选择一个常数 $c\in[3,100]$，然后令 $f(x)=x^2+c$。

考虑序列 $x,f(x),f(f(x)),\cdots$，根据之前的结论，该序列的周期是 $O(\sqrt{n})$ 的。令 p 表示 n 的最小质因子，那么该序列在模 p 意义下的周期应当是 $O\left(\sqrt{p}\right)$ 的。而且，在大多数情况下，这个周期是不等于其在模 n 意义下的周期的。

因此，考虑走过一个模 p 意义下的周期，则序列中有两数 x,y 满足 $p\mid x-y$，但 $n\nmid x-y$。于是 $\gcd(x-y,n)$ 为 n 的非平凡因子，我们就找到了一个因子。

考虑 Pollard-Rho 算法的时间复杂度。首先，由于周期的期望大小为 $O\left(\sqrt{p}\right) = O\left(n^{\frac{1}{4}}\right)$。而每次求最大公约数需要花费 $O\left(\log n\right)$ 的时间，因此朴素实现的期望时间复杂度将为 $O\left(n^{\frac{1}{4}} \log n\right)$。

考虑 Pollard-Rho 算法的时间复杂度。首先，由于周期的期望大小为 $O\left(\sqrt{p}\right)=O\left(n^{\frac{1}{4}}\right)$。而每次求最大公约数需要花费 $O\left(\log n\right)$ 的时间，因此朴素实现的期望时间复杂度将为 $O\left(n^{\frac{1}{4}}\log n\right)$。

注意使用 Floyd 判圈法时不要在一个圈里待太久，因为圈长有可能非常长导致复杂度退化。可以设定一个阈值，如果步数超过阈值仍未找出非平凡因子则更换 f。

考虑 Pollard-Rho 算法的时间复杂度。首先，由于周期的期望大小为 $O(\sqrt{p}) = O\left(n^{\frac{1}{4}}\right)$。而每次求最大公约数需要花费 $O(\log n)$ 的时间，因此朴素实现的期望时间复杂度将为 $O\left(n^{\frac{1}{4}} \log n\right)$。

注意使用 Floyd 判圈法时不要在一个圈里待太久，因为圈长有可能非常长导致复杂度退化。可以设定一个阈值，如果步数超过阈值仍未找出非平凡因子则更换 f。

注意到在 Pollard-Rho 算法中，大部分的 $x-y$ 都是和 N 互素的，因此我们可以选择一个常数 M，然后将 $x-y$ 每 M 项连乘起来再与 N 作 $\gcd$，这样求 $\gcd$ 所花费的 $\log$ 的时间就可以忽略不计了。

Pollard $p-1$ 算法

大数分解还有许多算法，下面仅举一例 Pollard $p-1$ 算法。

Pollard $p-1$ 算法

大数分解还有许多算法，下面仅举一例 Pollard $p-1$ 算法。

Pollard $p-1$ 算法要求有素因数 p 使得 $p-1$ 的最大素因子比较小。它的流程如下：

Pollard $p-1$ 算法

大数分解还有许多算法，下面仅举一例 Pollard $p-1$ 算法。

Pollard $p-1$ 算法要求有素因数 p 使得 $p-1$ 的最大素因子比较小。它的流程如下：

1. 取定正整数 B。

Pollard $p-1$ 算法

大数分解还有许多算法，下面仅举一例 Pollard $p-1$ 算法。

Pollard $p-1$ 算法要求有素因数 p 使得 $p-1$ 的最大素因子比较小。它的流程如下：

1. 取定正整数 B。
2. 令 $M=\prod_{p\leq B, p\in\mathbb{P}} p^{\lfloor \log_p B\rfloor}$。

Pollard $p-1$ 算法

大数分解还有许多算法，下面仅举一例 Pollard $p-1$ 算法。

Pollard $p-1$ 算法要求有素因数 p 使得 $p-1$ 的最大素因子比较小。它的流程如下：

1. 取定正整数 B。
2. 令 $M=\prod\limits_{p\le B, p\in\mathbb{P}} p^{\lfloor \log_p B\rfloor}$。
3. 随机取正整数 $a\ge 2$，不妨设 $(a,N)=1$。

Pollard $p-1$ 算法

大数分解还有许多算法，下面仅举一例 Pollard $p-1$ 算法。

Pollard $p-1$ 算法要求有素因数 p 使得 $p-1$ 的最大素因子比较小。它的流程如下：

1. 取定正整数 B。
2. 令 $M = \prod_{p \le B, p \in \mathbb{P}} p^{\lfloor \log_p B \rfloor}$。
3. 随机取正整数 $a \ge 2$，不妨设 $(a, N) = 1$。
4. 计算 $g = \gcd\left(a^M - 1, N\right)$。

Pollard $p-1$ 算法

大数分解还有许多算法，下面仅举一例 Pollard $p-1$ 算法。

Pollard $p-1$ 算法要求有素因数 p 使得 $p-1$ 的最大素因子比较小。它的流程如下：

1. 取定正整数 B。
2. 令 $M = \prod_{p \le B, p \in \mathbb{P}} p^{\lfloor \log_p B \rfloor}$。
3. 随机取正整数 $a \ge 2$，不妨设 $(a, N) = 1$。
4. 计算 $g = \gcd\left(a^M - 1, N\right)$。
5. 若 $1 < g < N$，则 g 为 N 的非平凡因子。

Pollard $p-1$ 算法

大数分解还有许多算法，下面仅举一例 Pollard $p-1$ 算法。

Pollard $p-1$ 算法要求有素因数 p 使得 $p-1$ 的最大素因子比较小。它的流程如下：

1. 取定正整数 B。
2. 令 $M=\prod_{p\leq B, p\in\mathbb{P}} p^{\lfloor \log_p B\rfloor}$。
3. 随机取正整数 $a\geq 2$，不妨设 $(a,N)=1$。
4. 计算 $g=\gcd\left(a^M-1,N\right)$。
5. 若 $1<g<N$，则 g 为 N 的非平凡因子。
6. 若 $g=1$，说明 B 太小了，适当调大 B 的值。

Pollard $p-1$ 算法

大数分解还有许多算法，下面仅举一例 Pollard $p-1$ 算法。

Pollard $p-1$ 算法要求有素因数 p 使得 $p-1$ 的最大素因子比较小。它的流程如下：

1. 取定正整数 B。
2. 令 $M = \prod_{p \le B, p \in \mathbb{P}} p^{\lfloor \log_p B \rfloor}$。
3. 随机取正整数 $a \ge 2$，不妨设 $(a, N) = 1$。
4. 计算 $g = \gcd\left(a^M - 1, N\right)$。
5. 若 $1 < g < N$，则 g 为 N 的非平凡因子。
6. 若 $g = 1$，说明 B 太小了，适当调大 B 的值。
7. 若 $g = N$，说明 B 太大了，适当调小 B 的值。

阶

Definition 5.1 (阶)

对于正整数 a, p，$\gcd(a, p) = 1$，定义 a 在模 p 意义下的阶为最小的正整数 t 满足 $a^t \bmod p = 1$。

阶

Definition 5.1 (阶)

对于正整数 a, p，$\gcd(a,p)=1$，定义 a 在模 p 意义下的阶为最小的正整数 t 满足 $a^t \bmod p = 1$。

a 在模 p 意义下的阶记作 $\mathrm{ord}_p(a)$。对于整数 k，$a^k \equiv 1 \pmod p$ 当且仅当 $\mathrm{ord}_p(a) \mid k$。

阶的计算

由欧拉定理可知，$a^{\varphi}(p) \equiv 1 \pmod{p}$。因此，一定有 $\operatorname{ord}_p(a) \mid \varphi(p)$ 成立。

阶的计算

由欧拉定理可知，$a^{\varphi}(p) \equiv 1 \pmod p$。因此，一定有 $\mathrm{ord}_p(a) \mid \varphi(p)$ 成立。计算 $\mathrm{ord}_p(a)$ 时，初始令 $x = \varphi(p)$，之后依次枚举 $\varphi(p)$ 的质因子 p_i，如果 $a^{\frac{x}{p_i}} \equiv 1 \pmod p$，则令 $x \leftarrow \frac{x}{p_i}$。最终的 x 即为 $\mathrm{ord}_p(a)$。

阶的计算

由欧拉定理可知，$a^{\varphi}(p) \equiv 1 \pmod{p}$。因此，一定有 $\mathrm{ord}_p(a) \mid \varphi(p)$ 成立。计算 $\mathrm{ord}_p(a)$ 时，初始令 $x = \varphi(p)$，之后依次枚举 $\varphi(p)$ 的质因子 p_i，如果 $a^{\frac{x}{p_i}} \equiv 1 \pmod{p}$，则令 $x \leftarrow \frac{x}{p_i}$。最终的 x 即为 $\mathrm{ord}_p(a)$。
瓶颈在于质因数分解，总复杂度 $O(\sqrt{n})$。

降幂

由于 $a^{\mathrm{ord}_p(a)} \equiv 1 \pmod p$，因此可以通过 $a^b \equiv a^{b \bmod \mathrm{ord}_p(a)} \pmod p$ 来降低指数。

降幂

由于 $a^{\mathrm{ord}_p(a)} \equiv 1 \pmod p$，因此可以通过 $a^b \equiv a^{b \bmod \mathrm{ord}_p(a)} \pmod p$ 来降低指数。

但这只对 $\gcd(a,p)=1$ 的情况有效，我们无法直接对 $\gcd(a,p) \neq 1$ 的情况降低指数。

降幂

由于 $a^{\mathrm{ord}_p(a)} \equiv 1 \pmod p$，因此可以通过 $a^b \equiv a^{b \bmod \mathrm{ord}_p(a)} \pmod p$ 来降低指数。

但这只对 $\gcd(a,p)=1$ 的情况有效，我们无法直接对 $\gcd(a,p) \neq 1$ 的情况降低指数。

为解决这个问题，我们可以对上述结论进行推广：

当 $\gcd(a^b,p)=\gcd(a^\infty,p)$ 时，有 $a^{b+\mathrm{ord}_p(a)} \equiv a^b \pmod p$。

例题

给定质数 p 和整数 a, b，判断是否存在非负整数 t 使得 $a^t \equiv b \pmod{p}$。

我们注意到 $a^t \equiv b \pmod p$ 的一个必要条件是 $\mathrm{ord}_p(b) \mid \mathrm{ord}_p(a)$。我们猜测它也是一个充分条件，下面给出证明。

我们注意到 $a^t \equiv b \pmod p$ 的一个必要条件是 $\mathrm{ord}_p(b) \mid \mathrm{ord}_p(a)$。我们猜测它也是一个充分条件，下面给出证明。

Lemma 5.1 (拉格朗日定理（数论）)

对于质数 p 和 n 次多项式 f，f 在模 p 意义下至多有 n 个根。

我们注意到 $a^t \equiv b \pmod p$ 的一个必要条件是 $\mathrm{ord}_p(b) \mid \mathrm{ord}_p(a)$。我们猜测它也是一个充分条件，下面给出证明。

Lemma 5.1 (拉格朗日定理（数论）)

对于质数 p 和 n 次多项式 f，f 在模 p 意义下至多有 n 个根。

证明：设根为 $\alpha_1, \alpha_2, \cdots, \alpha_{n+1}$，则该多项式对 $(x-\alpha_1)(x-\alpha_2)\cdots(x-\alpha_n)$ 取模后的结果有至少 n 个根且不为 0，使用数学归纳法证明即可。

因此，对于整数 a，如果 $\mathrm{ord}_p(a) = d$，则 $a^0, a^1, \cdots, a^{d-1}$ 在模 p 意义下互不相同。而方程 $x^d \equiv 1 \pmod p$ 至多有 d 个根，因此这 d 个根恰好是 $a^0, a^1, \cdots, a^{d-1}$。

因此，对于整数 a，如果 $\text{ord}_p(a) = d$，则 $a^0, a^1, \cdots, a^{d-1}$ 在模 p 意义下互不相同。而方程 $x^d \equiv 1 \pmod p$ 至多有 d 个根，因此这 d 个根恰好是 $a^0, a^1, \cdots, a^{d-1}$。

因此，如果 $\text{ord}_p(b) \mid \text{ord}_p(a)$ 成立，则 b 是 $x^d \equiv 1 \pmod p$ 的一个根，也就意味着存在非负整数 t 使得 $a^t \equiv b \pmod p$。

ABC335G Discrete Logarithm Problems

给定 n 个正整数 $a_1, a_2, \cdots, a_n$ 和一个素数 p。求满足以下条件的二元组 (i, j) 的个数。

- $1 \le i, j \le n$。
- 存在正整数 k，使得 $a_i^k \equiv a_j \pmod p$。

$2 \le n \le 2 \times 10^5$，$1 \le a_i < p$，$2 \le p \le 10^{13}$，$p$ 是质数。

根据之前的结论，存在正整数 k 使得 $a_i^k \equiv a_j \pmod p$ 当且仅当 $\mathrm{ord}_p(a_j) \mid \mathrm{ord}_p(a_i)$。

根据之前的结论，存在正整数 k 使得 $a_i^k \equiv a_j \pmod p$ 当且仅当 $\mathrm{ord}_p(a_j) \mid \mathrm{ord}_p(a_i)$。

可以对所有的 $\mathrm{ord}_p(a_i)$ 去重并计数，然后暴力枚举二元组。暴力枚举部分的复杂度为 $O\left(d(p)^2\right)$，可以接受。

P4139 上帝与集合的正确用法

定义 $a_0 = 1, a_n = 2^{a_{n-1}}$，可以证明 $b_n = a_n \bmod p$ 在某一项后都是同一个值，求这个值。

$T \le 10^3$，$p \le 10^7$。

题解

令 $f(x)$ 表示 $2^{2^{2^{\cdots}}} \bmod x$，则答案等于 $f(p)$。

题解

令 $f(x)$ 表示 $2^{2^{2^{\cdots}}} \bmod x$，则答案等于 $f(p)$。

根据欧拉定理及之前的扩展降幂结论，有 $f(p) = 2^{f(\varphi(p))+k\varphi(p)} \bmod p$，其中 $f(\varphi(p)) + k\varphi(p) \geq \log_2(p)$，递归计算即可。

原根

Definition 5.2 (原根)

对于自然数 p 和整数 a，如果 $\gcd(a, p) = 1$ 且 $\mathrm{ord}_p(a) = \varphi(p)$，则称 a 为模 p 意义下的原根。

原根

Definition 5.2 (原根)

对于自然数 p 和整数 a，如果 $\gcd(a,p)=1$ 且 $\mathrm{ord}_p(a)=\varphi(p)$，则称 a 为模 p 意义下的原根。

Theorem 5.1 (质数的原根存在定理)

所有质数都存在原根。

原根

Definition 5.2 (原根)

对于自然数 p 和整数 a，如果 $\gcd(a,p)=1$ 且 $\mathrm{ord}_p(a)=\varphi(p)$，则称 a 为模 p 意义下的原根。

Theorem 5.1 (质数的原根存在定理)

所有质数都存在原根。

Theorem 5.2 (质数幂的原根存在定理)

所有奇质数的幂都存在原根。

Lemma 5.2

对于质数 p 和与 p 互质的整数 a, b，如果 $\gcd(\mathrm{ord}_p(a), \mathrm{ord}_p(b)) = 1$，则 $\mathrm{ord}_p(ab) = \mathrm{ord}_p(a)\,\mathrm{ord}_p(b)$。

Lemma 5.2

对于质数 p 和与 p 互质的整数 a, b，如果 $\gcd(\mathrm{ord}_p(a), \mathrm{ord}_p(b)) = 1$，则 $\mathrm{ord}_p(ab) = \mathrm{ord}_p(a)\,\mathrm{ord}_p(b)$。

证明：显然 $(ab)^{\mathrm{ord}_p(a)\,\mathrm{ord}_p(b)} \equiv 1 \pmod p$，而对于任意质数 q 满足 $q \mid \mathrm{ord}_p(a)\,\mathrm{ord}_p(b)$，都有 $(ab)^{\frac{\mathrm{ord}_p(a)\,\mathrm{ord}_p(b)}{q}} \not\equiv 1 \pmod p$。

Lemma 5.3

对于质数 p 和与 p 互质的整数 a, b，一定存在一个与 p 互质的数 c，满足 $\mathrm{ord}_p(c) = \mathrm{lcm}(\mathrm{ord}_p(a), \mathrm{ord}_p(b))$。

Lemma 5.3

对于质数 p 和与 p 互质的整数 a,b，一定存在一个与 p 互质的数 c，满足 $\mathrm{ord}_p(c)=\mathrm{lcm}(\mathrm{ord}_p(a),\mathrm{ord}_p(b))$。

证明：对于每个质数 q，如果 $\gcd(q^\infty,\mathrm{ord}_p(a))>\gcd(q^\infty,\mathrm{ord}_p(b))$，则令 $b\leftarrow b^{\gcd(q^\infty,\mathrm{ord}_p(b))}$，否则令 $a\leftarrow a^{\gcd(q^\infty,\mathrm{ord}_p(a))}$。

Lemma 5.3

对于质数 p 和与 p 互质的整数 a,b，一定存在一个与 p 互质的数 c，满足 $\operatorname{ord}_p(c)=\operatorname{lcm}(\operatorname{ord}_p(a),\operatorname{ord}_p(b))$。

证明：对于每个质数 q，如果 $\gcd(q^\infty,\operatorname{ord}_p(a))>\gcd(q^\infty,\operatorname{ord}_p(b))$，则令 $b\leftarrow b^{\gcd(q^\infty,\operatorname{ord}_p(b))}$，否则令 $a\leftarrow a^{\gcd(q^\infty,\operatorname{ord}_p(a))}$。

令最终得到的数为 a',b'，则有 $\gcd(\operatorname{ord}_p(a'),\operatorname{ord}_p(b'))=1$ 且 $\operatorname{ord}_p(a')\operatorname{ord}_p(b')=\operatorname{lcm}(\operatorname{ord}_p(a),\operatorname{ord}_p(b))$。结合上一页的引理可知，$a'b'$ 即为满足要求的 c。

Proof (质数的原根存在定理)

1. 初始令 $x = 1$。

Proof (质数的原根存在定理)

1. 初始令 $x = 1$。
2. 如果当前 $\operatorname{ord}_p(x) = p - 1$，则 x 即为原根。

Proof (质数的原根存在定理)

1. 初始令 $x = 1$。
2. 如果当前 $\mathrm{ord}_p(x) = p - 1$，则 x 即为原根。
3. 否则当前必然有 $\mathrm{ord}_p(x) < p - 1$，令 $t = \mathrm{ord}_p(x)$，考虑方程 $x^t \equiv 1 \pmod p$，由拉格朗日定理它至多有 t 个根，因此 $1 \sim p - 1$ 中必然存在一个数 y 不满足 $y^t \equiv 1 \pmod p$。

Proof (质数的原根存在定理)

1. 初始令 $x = 1$。
2. 如果当前 $\mathrm{ord}_p(x) = p - 1$，则 x 即为原根。
3. 否则当前必然有 $\mathrm{ord}_p(x) < p - 1$，令 $t = \mathrm{ord}_p(x)$，考虑方程 $x^t \equiv 1 \pmod p$，由拉格朗日定理它至多有 t 个根，因此 $1 \sim p - 1$ 中必然存在一个数 y 不满足 $y^t \equiv 1 \pmod p$。
4. 根据上一页的引理，可以找到 x' 满足 $\mathrm{ord}_p(x') = \mathrm{lcm}(\mathrm{ord}_p(x), \mathrm{ord}_p(y))$。由于 $\mathrm{ord}_p(y) \nmid t$，故 $\mathrm{ord}_p(x')$ 一定大于 $\mathrm{ord}_p(x)$。令 $x \leftarrow x'$，回到步骤 2。

Proof (质数的原根存在定理)

1. 初始令 $x=1$。
2. 如果当前 $\operatorname{ord}_p(x)=p-1$，则 x 即为原根。
3. 否则当前必然有 $\operatorname{ord}_p(x)<p-1$，令 $t=\operatorname{ord}_p(x)$，考虑方程 $x^t\equiv 1 \pmod p$，由拉格朗日定理它至多有 t 个根，因此 $1\sim p-1$ 中必然存在一个数 y 不满足 $y^t\equiv 1 \pmod p$。
4. 根据上一页的引理，可以找到 x' 满足 $\operatorname{ord}_p(x')=\operatorname{lcm}(\operatorname{ord}_p(x),\operatorname{ord}_p(y))$。由于 $\operatorname{ord}_p(y)\nmid t$，故 $\operatorname{ord}_p(x')$ 一定大于 $\operatorname{ord}_p(x)$。令 $x\leftarrow x'$，回到步骤 2。

不难发现上述过程一定会终止，故质数一定存在原根。

Lemma 5.4

对于任意奇质数 p，一定存在 p 的原根 g，满足 $g^{p-1} \not\equiv 1 \pmod{p^2}$。

Lemma 5.4

对于任意奇质数 p，一定存在 p 的原根 g，满足 $g^{p-1} \not\equiv 1 \pmod{p^2}$。

证明：考虑任意 p 的原根 g，如果 g 满足条件则直接取 g 即可。

Lemma 5.4

对于任意奇质数 p，一定存在 p 的原根 g，满足 $g^{p-1} \not\equiv 1 \pmod{p^2}$。

证明：考虑任意 p 的原根 g，如果 g 满足条件则直接取 g 即可。

否则有 $(p+g)^{p-1} \equiv \sum_{i=0}^{p-1} \binom{p-1}{i} p^i g^{p-1-i} \equiv g^{p-1} + pg^{p-2} \equiv 1 + g^{p-2}p \not\equiv 1 \pmod{p^2}$。即 $g+p$ 满足条件。

Lemma 5.4

对于任意奇质数 p，一定存在 p 的原根 g，满足 $g^{p-1} \not\equiv 1 \pmod{p^2}$。

证明：考虑任意 p 的原根 g，如果 g 满足条件则直接取 g 即可。

否则有 $(p+g)^{p-1} \equiv \sum_{i=0}^{p-1} \binom{p-1}{i} p^i g^{p-1-i} \equiv g^{p-1} + pg^{p-2} \equiv 1 + g^{p-2}p \not\equiv 1 \pmod{p^2}$。即 $g+p$ 满足条件。

令满足条件的数为 g'，有 $g'^{p-1} \equiv 1 + kp \pmod{p^2}$（$p \nmid k$）。考虑 $g'^{a(p-1)} \bmod p^2$，有 $g'^{a(p-1)} \equiv (1+kp)^a \equiv 1 + akp \pmod{p^2}$，因此 $\mathrm{ord}_{p^2}(g') = p(p-1) = \varphi(p^2)$。

Proof (质数幂的原根存在定理)

由上一页的引理可知，存在 g 满足 $\mathrm{ord}_{p^2}(g) = \varphi(p^2)$。把 $k = 2$ 时 g 是 p^k 的原根作为初始条件，并对 k 使用数学归纳法。

Proof (质数幂的原根存在定理)

由上一页的引理可知，存在 g 满足 $\mathrm{ord}_{p^2}(g)=\varphi(p^2)$。把 $k=2$ 时 g 是 p^k 的原根作为初始条件，并对 k 使用数学归纳法。

上述的 g 满足 $g^{(p-1)p^{k-2}}\equiv 1+tp^{k-1}\pmod{p^k}$，其中 $p\nmid t$。

Proof (质数幂的原根存在定理)

由上一页的引理可知，存在 g 满足 $\mathrm{ord}_{p^2}(g) = \varphi(p^2)$。把 $k = 2$ 时 g 是 p^k 的原根作为初始条件，并对 k 使用数学归纳法。

上述的 g 满足 $g^{(p-1)p^{k-2}} \equiv 1 + tp^{k-1} \pmod{p^k}$，其中 $p \nmid t$。

$g^{(p-1)p^{k-2}} \equiv 1 + tp^{k-1} + t'p^k \pmod{p^{k+1}}$，对等式两侧求 p 次方得 $g^{(p-1)p^{k-1}} \equiv 1 + tp^k \pmod{p^{k+1}}$。

Proof (质数幂的原根存在定理)

由上一页的引理可知，存在 g 满足 $\mathrm{ord}_{p^2}(g)=\varphi(p^2)$。把 $k=2$ 时 g 是 p^k 的原根作为初始条件，并对 k 使用数学归纳法。

上述的 g 满足 $g^{(p-1)p^{k-2}}\equiv 1+tp^{k-1}\pmod{p^k}$，其中 $p\nmid t$。

$g^{(p-1)p^{k-2}}\equiv 1+tp^{k-1}+t'p^k\pmod{p^{k+1}}$，对等式两侧求 p 次方得 $g^{(p-1)p^{k-1}}\equiv 1+tp^k\pmod{p^{k+1}}$。

参照引理 5.4 后的部分，可以得到 $\mathrm{ord}_{p^{k+1}}(g)=\varphi(p^{k+1})$。

Proof (质数幂的原根存在定理)

由上一页的引理可知，存在 g 满足 $\mathrm{ord}_{p^2}(g) = \varphi(p^2)$。把 $k=2$ 时 g 是 p^k 的原根作为初始条件，并对 k 使用数学归纳法。

上述的 g 满足 $g^{(p-1)p^{k-2}} \equiv 1 + tp^{k-1} \pmod{p^k}$，其中 $p \nmid t$。

$g^{(p-1)p^{k-2}} \equiv 1 + tp^{k-1} + t'p^k \pmod{p^{k+1}}$，对等式两侧求 p 次方得 $g^{(p-1)p^{k-1}} \equiv 1 + tp^k \pmod{p^{k+1}}$。

参照引理 5.4 后的部分，可以得到 $\mathrm{ord}_{p^{k+1}}(g) = \varphi(p^{k+1})$。

由上述证明可知，如果 g 是 p^2 的原根，则它一定是 p^i 的原根（$i \geq 2$）。$\dfrac{\varphi(p^i)}{\varphi(p^2)} = p^{i-2}$ 也从侧面验证了这一性质。

求原根

枚举每个满足 $\gcd(a,p)=1$ 的 a，并求出 a 在模 p 意义下的阶，直到找到一个 a 满足 $\mathrm{ord}_p(a)=p-1$。

求原根

枚举每个满足 $\gcd(a, p) = 1$ 的 a，并求出 a 在模 p 意义下的阶，直到找到一个 a 满足 $\mathrm{ord}_p(a) = p - 1$。

首先原根有 $\varphi(\varphi(p))$ 个，因此最小的原根不会很大，复杂度可以接受。具体复杂度不太重要，一方面是数论相关的内容不适合用传统的方式描述复杂度，另一方面是一般的题目中至多只需要求一次原根。

P6091 【模板】原根

模板题。

P5605 小 A 与两位神仙

给定质数幂 m，n 次询问。每次询问给出两个正整数 x, y，保证 $\gcd(x, m) = \gcd(y, m) = 1$，询问是否存在非负整数 a 满足 $x^a \equiv y \pmod m$。

$1 \le n \le 2 \times 10^4$，$3 \le m \le 10^18$，$1 \le x, y < m$。

由于模质数幂意义下存在原根，因此可以求出 x, y 的离散对数 x', y'，存在非负整数 a 满足 $x^a \equiv y \pmod{m}$ 等价于存在 a 满足 $ax' \equiv y' \pmod{\varphi(m)}$。

由于模质数幂意义下存在原根，因此可以求出 x, y 的离散对数 x', y'，存在非负整数 a 满足 $x^a \equiv y \pmod{m}$ 等价于存在 a 满足 $ax' \equiv y' \pmod{\varphi(m)}$。

由模意义的性质可知，这等价于 $\gcd(x', m) \mid \gcd(y', m)$，即 $\operatorname{ord}_m(y) \mid \operatorname{ord}_m(x)$，故求阶即可。

2 的幂的伪原根

当 $k \geq 3$ 时，2^k 不存在原根，即无法找到 g 满足 $\mathrm{ord}_{2^k}(g) = 2^{k-1}$。但我们仍然能找到 g 满足 $\mathrm{ord}_{2^k}(g) = 2^{k-2}$，此时我们称 g 为 2^k 的伪原根（名字也是我随便取的）。

2 的幂的伪原根

当 $k \geq 3$ 时，2^k 不存在原根，即无法找到 g 满足 $\operatorname{ord}_{2^k}(g) = 2^{k-1}$。但我们仍然能找到 g 满足 $\operatorname{ord}_{2^k}(g) = 2^{k-2}$，此时我们称 g 为 2^k 的伪原根（名字也是我随便取的）。

仍然考虑数学归纳法，当 $k = 3$ 时，存在 $g = 5$ 满足 $g^{2^{k-3}} \equiv 1 + 2^{k-1} \pmod{2^k}$。接下来证明该等式在 $k = k + 1$ 时也成立。

2 的幂的伪原根

当 $k \geq 3$ 时，2^k 不存在原根，即无法找到 g 满足 $\operatorname{ord}_{2^k}(g) = 2^{k-1}$。但我们仍然能找到 g 满足 $\operatorname{ord}_{2^k}(g) = 2^{k-2}$，此时我们称 g 为 2^k 的伪原根（名字也是我随便取的）。

仍然考虑数学归纳法，当 $k = 3$ 时，存在 $g = 5$ 满足 $g^{2^{k-3}} \equiv 1 + 2^{k-1} \pmod{2^k}$。接下来证明该等式在 $k = k + 1$ 时也成立。

有 $g^{2^{k-3}} \equiv 1 + 2^{k-1} + t2^k \pmod{2^{k+1}}$，对它两边平方得到 $g^{2^{k-2}} \equiv 1 + 2^k \pmod{2^{k+1}}$。因此，对于任意 $k \geq 3$ 有 $\operatorname{ord}_{2^k}(g) = 2^{k-2}$。

2 的幂的伪原根

当 $k\geq 3$ 时，2^k 不存在原根，即无法找到 g 满足 $\mathrm{ord}_{2^k}(g)=2^{k-1}$。但我们仍然能找到 g 满足 $\mathrm{ord}_{2^k}(g)=2^{k-2}$，此时我们称 g 为 2^k 的伪原根（名字也是我随便取的）。

仍然考虑数学归纳法，当 $k=3$ 时，存在 $g=5$ 满足 $g^{2^{k-3}}\equiv 1+2^{k-1}\pmod{2^k}$。接下来证明该等式在 $k=k+1$ 时也成立。

有 $g^{2^{k-3}}\equiv 1+2^{k-1}+t2^k\pmod{2^{k+1}}$，对它两边平方得到 $g^{2^{k-2}}\equiv 1+2^k\pmod{2^{k+1}}$。因此，对于任意 $k\geq 3$ 有 $\mathrm{ord}_{2^k}(g)=2^{k-2}$。

此外，由于 $x^2+1\equiv 0\pmod 8$ 无解，故 $k\geq 3$ 时不存在 t 满足 $g^t\equiv -1\pmod{2^k}$。因此，$(-1)^a g^b$（$a\in[0,2),b\in[0,2^{k-2})$）可以遍历模 2^k 意义下的所有奇数，起到类似原根的效果。实际应用的时候，可以直接令 $g=5$。

BSGS

Problem

给定质数 p 和底数 a，q 次询问，每次询问给出 k，求 $a^k \bmod p$ 的值。
$0 \le a < p \le 2 \times 10^9$，要求单次查询 $O(1)$。

BSGS

Problem

给定质数 p 和底数 a，q 次询问，每次询问给出 k，求 $a^k \bmod p$ 的值。
$0 \le a < p \le 2 \times 10^9$，要求单次查询 $O(1)$。

Solution

令 $B = \lfloor \sqrt{p} \rfloor$，预处理 $a^0 \bmod p, a^1 \bmod p, a^2 \bmod p, \cdots$，$a^B \bmod p, a^{2B} \bmod p, a^{3B} \bmod p, \cdots$。
则可以 $O(\sqrt{p})$ 预处理，$O(1)$ 查询。

P3846 [TJOI2007] 可爱的质数/【模板】BSGS

给定一个质数 p，以及一个整数 b，一个整数 n，现在要求你计算一个最小的非负整数 l，满足 $b^l \equiv n \pmod{p}$，或判断无解。

对于所有的测试点，保证 $2 \leq b, n < p < 2^{31}$。

令 $d = \mathrm{ord}_p(b)$，则问题相当于在 $b^0, b^1, \cdots, b^{d-1}$ 中找到与 n 同余的数。

令 $d = \mathrm{ord}_p(b)$，则问题相当于在 $b^0, b^1, \cdots, b^{d-1}$ 中找到与 n 同余的数。

令 $S = \lceil \sqrt{d} \rceil$，则 $b^0, b^1, \cdots, b^{d-1}$ 中的每个数都可以写成 b^{iS+j} 的形式，其中 $0 \le i, j < S$。原问题即为找到一组 i, j 使得 $b^{iS} = nb^{-j}$。

令 $d = \text{ord}_p(b)$，则问题相当于在 $b^0, b^1, \cdots, b^{d-1}$ 中找到与 n 同余的数。

令 $S = \lceil\sqrt{d}\rceil$，则 $b^0, b^1, \cdots, b^{d-1}$ 中的每个数都可以写成 b^{iS+j} 的形式，其中 $0 \le i, j < S$。原问题即为找到一组 i, j 使得 $b^{iS} = nb^{-j}$。

将所有的 nb^{-j} 加入哈希表，并在遍历 i 时查询 b^{iS} 是否在哈希表中，这样可以得到 $iS + j$ 最小的解。复杂度 $O(\sqrt{p})$。

令 $d = \mathrm{ord}_p(b)$，则问题相当于在 $b^0, b^1, \cdots, b^{d-1}$ 中找到与 n 同余的数。

令 $S = \lceil \sqrt{d} \rceil$，则 $b^0, b^1, \cdots, b^{d-1}$ 中的每个数都可以写成 b^{iS+j} 的形式，其中 $0 \le i, j < S$。原问题即为找到一组 i, j 使得 $b^{iS} = nb^{-j}$。

将所有的 nb^{-j} 加入哈希表，并在遍历 i 时查询 b^{iS} 是否在哈希表中，这样可以得到 $iS + j$ 最小的解。复杂度 $O(\sqrt{p})$。

当模数不变时，如果要求 T 次离散对数，可以通过更改块大小，将复杂度从 $O(T\sqrt{p})$ 变为 $O(\sqrt{Tp})$。

离散对数

对于质数 p 和其原根 g，任意满足 $\gcd(a,p)=1$ 的整数 a 在模 p 意义下都是 g 的幂。因此，我们把最小的满足 $g^t \equiv a \pmod{p}$ 的非负整数 t 称为 a 在模 p 意义下以 g 为底的离散对数。也可记作 $\log_g(a)$。

P4195 【模板】扩展 BSGS/exBSGS

多组数据，每组数据给定 a, p, b，求满足 $a^x\ b \pmod p$ 的最小自然数 x 。
$1 \le a, p, b\ 10^9$，$\sum \sqrt{p} \le 5 \times 10^6$。

此题没有保证 p 是质数，因此可能会出现无法求逆的情况，不能直接套用之前的做法。

此题没有保证 p 是质数，因此可能会出现无法求逆的情况，不能直接套用之前的做法。

序列 $a^0, a^1, a^2, \cdots$ 可以被分成两段，前一段满足 $\gcd(a^i, p)$ 递增，后一段满足 $\gcd(a^i, p)$ 始终相等。前一段的长度不超过 $\log p$，可以暴力枚举。而后一段满足 $\gcd\left(a, \dfrac{p}{\gcd(a^i, p)}\right) = 1$，可以放到模 $\dfrac{p}{\gcd(a^i, p)}$ 意义下考虑，此时可以正常求逆，使用 BSGS 算法即可。

HDU 6632 discrete logarithm problem

https://vjudge.net/problem/HDU-6632

T 组数据，每组数据给定 a, b, p，求最小的正整数 x 满足 $a^x \equiv b \pmod{p}$。

保证 p 是质数，且 $p-1$ 不包含 2 和 3 之外的质因子。

$T \le 200, 65537 \le p \le 10^{18}, 2 \le a, b \le p-1$。

Pohlig–Hellman algorithm

如果 $p-1$ 的质因子都很小，则我们有另一种求解离散对数的算法。

Pohlig–Hellman algorithm

如果 $p-1$ 的质因子都很小，则我们有另一种求解离散对数的算法。
首先计算 a, b 的阶以判断是否有解。令 $\operatorname{ord}_p(a) = d$，对 d 进行质因子分解，得到 $d = \prod_{i=1}^{k} p_i^{\alpha_i}$。

Pohlig-Hellman algorithm

如果 $p-1$ 的质因子都很小，则我们有另一种求解离散对数的算法。

首先计算 a, b 的阶以判断是否有解。令 $\mathrm{ord}_p(a) = d$，对 d 进行质因子分解，得到 $d = \prod_{i=1}^{k} p_i^{\alpha_i}$。

我们可以分别求出 $x \bmod p_i^{\alpha_i}$ 的值，再用中国剩余定理合并。

Pohlig–Hellman algorithm

如果 $p-1$ 的质因子都很小，则我们有另一种求解离散对数的算法。

首先计算 a,b 的阶以判断是否有解。令 $\mathrm{ord}_p(a)=d$，对 d 进行质因子分解，得到 $d=\prod_{i=1}^{k} p_i^{\alpha_i}$。

我们可以分别求出 $x \bmod p_i^{\alpha_i}$ 的值，再用中国剩余定理合并。

对于单个 p_i，可以将原方程变为 $\left(a^{\frac{d}{p_i}}\right)^x \equiv b^{\frac{d}{p_i}} \pmod p$，以求出 $x \bmod p_i$ 的值。再将原方程变为 $\left(a^{\frac{d}{p_i^2}}\right)^x \equiv b^{\frac{d}{p_i^2}} \pmod p$，以求出 $x \bmod p_i^2$ 的值。按此方法不断递增指数，直到求出 $x \bmod p_i^{\alpha_i}$。

单次求解 $x \bmod p_i$ 或提升指数的复杂度为 $O(p_i)$，因此总复杂度为 $O(\sum \alpha_i p_i)$。如果用 BSGS 优化，则复杂度变为 $O(\sum \alpha_i \sqrt{p_i})$。

二次剩余

Definition 6.1 (二次剩余)

对于正整数 p 和整数 a，如果 $\gcd(a, p) = 1$，且同余方程 $x^2 \equiv a \pmod p$ 有解，则称 a 为 p 的二次剩余。

二次剩余

Definition 6.1 (二次剩余)

对于正整数 p 和整数 a，如果 $\gcd(a, p) = 1$，且同余方程 $x^2 \equiv a \pmod p$ 有解，则称 a 为 p 的二次剩余。

Theorem 6.1

设 p 是奇质数，a 是 $[1, p-1]$ 之间的整数，则同余方程 $x^2 \equiv a$ 或者无解，或者有两个不同余的解。

二次剩余

Definition 6.1 (二次剩余)

对于正整数 p 和整数 a，如果 $\gcd(a,p)=1$，且同余方程 $x^2 \equiv a \pmod{p}$ 有解，则称 a 为 p 的二次剩余。

Theorem 6.1

设 p 是奇质数，a 是 $[1,p-1]$ 之间的整数，则同余方程 $x^2 \equiv a$ 或者无解，或者有两个不同余的解。

证明：如果有一个解 x，则 $-x$ 也是解，且 x 与 $-x$ 不同余。对于两个解 x_1, x_2，根据平方差公式有 $(x_1-x_2)(x_1+x_2) \equiv 0 \pmod{p}$。由于 x_1, x_2 不相等，因此 $x_2 = -x_1$。

Corollary 6.1

设 p 是奇质数，则 $1, 2, \cdots, p-1$ 中恰有 $\frac{p-1}{2}$ 个二次剩余。

Corollary 6.1

设 p 是奇质数，则 $1, 2, \cdots, p-1$ 中恰有 $\frac{p-1}{2}$ 个二次剩余。

证明：由于同余方程 $x^2 \equiv a$ 或者无解，或者有两个不同余的解，那么 $1, 2, \cdots, p-1$ 必然是两两配对的，也就意味着恰有 $\frac{p-1}{2}$ 个二次剩余。

Corollary 6.1

设 p 是奇质数，则 $1, 2, \cdots, p-1$ 中恰有 $\frac{p-1}{2}$ 个二次剩余。

证明：由于同余方程 $x^2 \equiv a$ 或者无解，或者有两个不同余的解，那么 $1, 2, \cdots, p-1$ 必然是两两配对的，也就意味着恰有 $\frac{p-1}{2}$ 个二次剩余。

另一种证明：取 p 的一个原根 g，则只有 $g^0, g^2, g^4, \cdots, g^{p-3}$ 是二次剩余。

Theorem 6.2 (欧拉判别法)

对于奇质数 p 和整数 a，如果 $\gcd(a,p)=1$，则 a 是二次剩余当且仅当 $a^{\frac{p-1}{2}} \equiv 1 \pmod p$，否则 $a^{\frac{p-1}{2}} \equiv -1 \pmod p$。

Theorem 6.2 (欧拉判别法)

对于奇质数 p 和整数 a，如果 $\gcd(a,p)=1$，则 a 是二次剩余当且仅当 $a^{\frac{p-1}{2}} \equiv 1 \pmod p$，否则 $a^{\frac{p-1}{2}} \equiv -1 \pmod p$。

Proof

当 a 是二次剩余时，$a^{\frac{p-1}{2}} \equiv (x^2)^{\frac{p-1}{2}} \equiv x^{p-1} \equiv 1$。当 a 不是二次剩余时，可以将 $1, 2, \cdots, p-1$ 分成 $\dfrac{p-1}{2}$ 对，使得每一对的乘积为 a。由威尔逊定理可知，$a^{\frac{p-1}{2}} \equiv (p-1)! \equiv -1 \pmod p$。

Cipolla 算法

Problem

求解方程 $x^2 \equiv c \pmod{p}$。

Cipolla 算法

Problem

求解方程 $x^2 \equiv c \pmod p$。

首先用欧拉判别法，如果 c 不是 p 的二次剩余则无解。

Cipolla 算法

Problem

求解方程 $x^2 \equiv c \pmod p$。

首先用欧拉判别法，如果 c 不是 p 的二次剩余则无解。

如果有解，则不断随机 a，直到 $a^2 - c$ 不是 p 的二次剩余。单次随机的成功率为 50%，故最坏也只需要随 $\log$ 次。

Cipolla 算法

Problem

求解方程 $x^2 \equiv c \pmod{p}$。

首先用欧拉判别法，如果 c 不是 p 的二次剩余则无解。

如果有解，则不断随机 a，直到 $a^2 - c$ 不是 p 的二次剩余。单次随机的成功率为 50%，故最坏也只需要随 $\log$ 次。

添加虚数 w，令其满足 $w^2 \equiv a^2 - c$。我们声称方程的解为 $x \equiv (a+w)^{\frac{p+1}{2}}$，证明见后文。

Lemma 6.1

$$(a+w)^p \equiv a^p + w^p$$

Lemma 6.1

$$(a+w)^p \equiv a^p + w^p$$

证明：由二项式定理，$(a+w)^p = \sum_{i=0}^{p} \binom{p}{i} a^i w^{a-i}$。注意到 $\binom{p}{i} = \frac{p!}{i!(p-i)!}$，因此 $\binom{p}{i}$ 不含 p 因子当且仅当 $i=0$ 或 $i=p$，即 $(a+w)^p \equiv a^p + w^p$。

Lemma 6.2

$$w^p \equiv -w$$

Lemma 6.2

$$w^p \equiv -w$$

证明：由于 $a^2 - c$ 不是二次剩余，因此 $(a^2 - c)^{\frac{p-1}{2}} \equiv -1$。
$w^p \equiv w \times w^{p-1} \equiv w \times (a^2 - c)^{\frac{p-1}{2}} \equiv -w$。

Lemma 6.2

$$w^p \equiv -w$$

证明：由于 $a^2 - c$ 不是二次剩余，因此 $(a^2-c)^{\frac{p-1}{2}} \equiv -1$。
$w^p \equiv w \times w^{p-1} \equiv w \times (a^2-c)^{\frac{p-1}{2}} \equiv -w$。
因此，
$(a+w)^{p+1} \equiv (a+w)^p(a+w) \equiv (a^p+w^p)(a+w) \equiv (a-w)(a+w) \equiv a^2-w^2 \equiv c$，
即 $x \equiv (a+w)^{\frac{p+1}{2}}$。

Lemma 6.2

$$w^p \equiv -w$$

证明：由于 $a^2 - c$ 不是二次剩余，因此 $(a^2-c)^{\frac{p-1}{2}} \equiv -1$。
$w^p \equiv w \times w^{p-1} \equiv w \times (a^2-c)^{\frac{p-1}{2}} \equiv -w$。
因此，
$(a+w)^{p+1} \equiv (a+w)^p(a+w) \equiv (a^p+w^p)(a+w) \equiv (a-w)(a+w) \equiv a^2-w^2 \equiv c$，
即 $x \equiv (a+w)^{\frac{p+1}{2}}$。

关于 $(a+w)^{\frac{p+1}{2}}$ 虚部为 0 的证明：假设 $(a+w)^{\frac{p+1}{2}} \equiv A+Bw$，则 $(A+Bw)^2 \equiv A^2+B^2(a^2-c)+2ABw$。因此有 $A \equiv 0$ 或 $B \equiv 0$。如果 $A \equiv 0$，则 $B^2(a^2-c)$ 不是 p 的二次剩余，不可能等于 c，因此有 $B \equiv 0$。

P5491 【模板】二次剩余

模板题。

P5668 【模板】N 次剩余

你需要解方程 $x^n \equiv k \pmod{m}$，并输出方程的所有解，其中 $x \in [0, m-1]$。共 T 组数据。

$1 \le T \le 100$，$1 \le n \le 10^9$，$0 \le k < m \le 10^9$。

设 m 的唯一分解形式为 $m = \prod_{i=1}^{s} p_i^{q_i}$，保证方程 $x^n \equiv k \pmod{p_i^{q_i}}$ 在 $[0, p_i^{q_i})$ 中的解数 $\le 10^6$。

首先用中国剩余定理将 m 拆成质数幂的乘积。令 $m=\prod_{i=1}^{k}p_i^{\alpha_i}$，如果对于 $1\le i\le k$ 都有 $x^n\equiv k\pmod{p_i^{\alpha_i}}$ 成立，则一定有 $x^n\equiv k\pmod{m}$ 成立。

首先用中国剩余定理将 m 拆成质数幂的乘积。令 $m=\prod_{i=1}^{k} p_i^{\alpha_i}$，如果对于 $1\le i\le k$ 都有 $x^n\equiv k \pmod{p_i^{\alpha_i}}$ 成立，则一定有 $x^n\equiv k \pmod{m}$ 成立。

因此我们可以对于每个质数幂分别解方程，再使用中国剩余定理合并，整体解的个数应当是每个质数幂方程解的个数的乘积。

对于方程 $x^n \equiv c \pmod{p^k}$，先处理 $\gcd(c, p^k) \neq 1$ 的情况。

对于方程 $x^n \equiv c \pmod{p^k}$，先处理 $\gcd(c, p^k) \neq 1$ 的情况。
首先，如果 $c \equiv 0 \pmod{p^k}$，则方程成立当且仅当 $p^{\lceil \frac{k}{n} \rceil} \mid x$。

对于方程 $x^n \equiv c \pmod{p^k}$，先处理 $\gcd(c, p^k) \neq 1$ 的情况。

首先，如果 $c \equiv 0 \pmod{p^k}$，则方程成立当且仅当 $p^{\lceil \frac{k}{n} \rceil} \mid x$。

否则令 $c = p^s t$，其中 $p \nmid t$。如果 $n \nmid s$ 则无解，否则令 $x = p^{\frac{s}{n}} x'$，解方程 $x'^n \equiv t \pmod{p^{k-s}}$ 即可。

对于方程 $x^n \equiv c \pmod{p^k}$，先处理 $\gcd(c, p^k) \neq 1$ 的情况。

首先，如果 $c \equiv 0 \pmod{p^k}$，则方程成立当且仅当 $p^{\lceil \frac{k}{n} \rceil} \mid x$。

否则令 $c = p^s t$，其中 $p \nmid t$。如果 $n \nmid s$ 则无解，否则令 $x = p^{\frac{s}{n}} x'$，解方程 $x'^n \equiv t \pmod{p^{k-s}}$ 即可。

对于 $\gcd(c, p^k) = 1$ 的情况，如果 p 是奇质数，则 p^k 存在原根。可以先找到一个原根，然后对 c 求离散对数，将原问题转为离散对数方程 $n \log(x) \equiv \log(c) \pmod{\varphi(p^k)}$。

那么现在唯一的问题就是 $p=2$ 的情况。我们假定有 $k\geq 3$，$k\leq 2$ 的情况可以通过暴力枚举解决。

那么现在唯一的问题就是 $p=2$ 的情况。我们假定有 $k\geq 3$，$k\leq 2$ 的情况可以通过暴力枚举解决。

之前提到模 2^k 意义下的每个奇数都可以表示成 $(-1)^a5^b$ 的形式，因此，我们可以把 x 和 c 都表示成这个形式，然后对 -1 和 5 两维分别解方程。

那么现在唯一的问题就是 $p = 2$ 的情况。我们假定有 $k \geq 3$，$k \leq 2$ 的情况可以通过暴力枚举解决。

之前提到模 2^k 意义下的每个奇数都可以表示成 $(-1)^a 5^b$ 的形式，因此，我们可以把 x 和 c 都表示成这个形式，然后对 -1 和 5 两维分别解方程。

令 $x = (-1)^{x_1} 5^{x_2}, c = (-1)^{c_1} 5^{c_2}$，则方程组变为 $nx_1 \equiv c_1 \pmod 2$，$nx_2 \equiv c_2 \pmod{2^{k-2}}$。解出所有的 x_1, x_2 并合并即可。

P3403 跳楼机

给定 n, x, y, z，求有多少个 k 满足：

- $k \in [1, n]$。
- 存在非负整数 a, b, c，使得 $ax + by + cz = k$。

$1 \leq x, y, z \leq 10^5, h \leq 2^{63} - 1$。

考虑背包，令 f_i 表示 x, y, z 能否组合出 i。注意到如果 $f_i = 1$，则 f_{i+x} 一定等于 1。

考虑背包，令 f_i 表示 x, y, z 能否组合出 i。注意到如果 $f_i = 1$，则 f_{i+x} 一定等于 1。

因此可以精简 dp 状态，首先设 $x \le y \le z$，然后对于 $i \in [0, x)$，令 g_i 表示最小的 t 满足 $t \equiv i \pmod x$ 且 $f_t = 1$。

考虑背包，令 f_i 表示 x, y, z 能否组合出 i。注意到如果 $f_i = 1$，则 f_{i+x} 一定等于 1。

因此可以精简 dp 状态，首先设 $x \le y \le z$，然后对于 $i \in [0, x)$，令 g_i 表示最小的 t 满足 $t \equiv i \pmod x$ 且 $f_t = 1$。

现在 x 已经直接被放到了 dp 状态里了，那么还有 y 和 z，需要用转移的形式来体现。容易发现 y 的转移即为 $g_{(i+y) \bmod x} = \min(g_{(i+y) \bmod x}, g_i + y)$，$z$ 的转移同理。

考虑背包，令 f_i 表示 x, y, z 能否组合出 i。注意到如果 $f_i = 1$，则 f_{i+x} 一定等于 1。

因此可以精简 dp 状态，首先设 $x \le y \le z$，然后对于 $i \in [0, x)$，令 g_i 表示最小的 t 满足 $t \equiv i \pmod x$ 且 $f_t = 1$。

现在 x 已经直接被放到了 dp 状态里了，那么还有 y 和 z，需要用转移的形式来体现。容易发现 y 的转移即为 $g_{(i+y) \bmod x} = \min(g_{(i+y) \bmod x}, g_i + y)$，$z$ 的转移同理。

我们发现 y 和 z 的转移是可以分开进行的，因此可以先进行 y 的转移，再进行 z 的转移。接下来考虑如何完成 y 的转移。

显然我们可以把转移看成有向边，对于所有 i，从 i 向 $(i+y) \bmod x$ 连一条长度为 y 的边。然后给定初始最短路数组，求最终的最短路即可。

显然我们可以把转移看成有向边，对于所有 i，从 i 向 $(i+y) \bmod x$ 连一条长度为 y 的边。然后给定初始最短路数组，求最终的最短路即可。

但这需要 $O(x\log x)$ 的时间，太慢了。我们考虑以 y 为边长建立等差数列环模型，现在 dp 数组构成了若干个环，每个元素的 dp 值可以由它前一个元素的 dp 值 $+y$ 转移过来。

显然我们可以把转移看成有向边，对于所有 i，从 i 向 $(i+y) \bmod x$ 连一条长度为 y 的边。然后给定初始最短路数组，求最终的最短路即可。

但这需要 $O(x \log x)$ 的时间，太慢了。我们考虑以 y 为边长建立等差数列环模型，现在 dp 数组构成了若干个环，每个元素的 dp 值可以由它前一个元素的 dp 值 $+y$ 转移过来。

对于单个环，我们可以找到初始 dp 值最小的点，然后从它开始向后更新 dp 值，绕一圈后停止。这样的复杂度是 $O(x)$ 的。

显然我们可以把转移看成有向边，对于所有 i，从 i 向 $(i+y) \bmod x$ 连一条长度为 y 的边。然后给定初始最短路数组，求最终的最短路即可。

但这需要 $O(x \log x)$ 的时间，太慢了。我们考虑以 y 为边长建立等差数列环模型，现在 dp 数组构成了若干个环，每个元素的 dp 值可以由它前一个元素的 dp 值 $+y$ 转移过来。

对于单个环，我们可以找到初始 dp 值最小的点，然后从它开始向后更新 dp 值，绕一圈后停止。这样的复杂度是 $O(x)$ 的。

我们也可以不找初始 dp 值最小的点，而是从任意一个点开始，向后绕两圈来更新，这样显然也是正确的。

QOJ8225 [PKUWC 2024] 最小值之和（dp 转移部分）

初始对于每个 $1 \le i < n, 0 \le j \le m$，给定所有 $dp[i][i+1][j]$ 的值，值均属于 $\{0,1\}$。并保证，如果 $dp[i][i+1][j]=1$，则 $dp[i][i+1][j-1]$ 一定等于 1。还对于 $1 \le i \le n, 0 \le j \le m$ 给定了所有 $dp[i][i][j]$ 的值，满足 $dp[i][i][j]=[j=0]$。

对于 $1 \le l < l+1 < r \le n$，$0 \le k \le m$，$dp[l][r][k]$ 等于 1 当且仅当：

- 存在 $l \le i \le r$，$0 \le j$ 满足 $dp[l][i][k+j(r-l)]$ 和 $dp[i+1][r][k+j(r-l)]$ 均为 1。

请求出完整的 dp 数组。

$1 \le n \le 80, 0 \le m \le 10^8$。

P8457 「SWTR-8」幂塔方程

求解方程 $x^x \equiv D \pmod{n}$。

保证 n 的最大质因子不超过 10^5，且 D 与 n 互质。

你需要保证得到的解 x 为 $[0, 2^{125}]$ 范围内的整数。若该范围内无解，输出 -1；若存在多解，输出任意一个。

T 组测试数据。

$1 \le T \le 4 \times 10^4$，$2 \le n \le 10^{18}$，$1 \le D < n$，$D \perp n$，$2 \le p_1 < p_2 < \cdots < p_k \le 10^5$。

令 p 是一个质数，考虑先对 $n=p$ 的情况构造解。

令 p 是一个质数，考虑先对 $n = p$ 的情况构造解。
不难发现只要联立方程 $x \equiv D \pmod{p}$，$x \equiv 1 \pmod{p-1}$ 即可。

令 p 是一个质数，考虑先对 $n = p$ 的情况构造解。

不难发现只要联立方程 $x \equiv D \pmod p$，$x \equiv 1 \pmod{p-1}$ 即可。

假设现在已经有了 $n = p^k$ 的解，考虑将它变为 $n = p^{k+1}$ 的解。注意到令 $x \leftarrow x + t(p-1)p^k$ 后它仍是 $n = p^k$ 的解，因此可以设 $x' = x + t(p-1)p^k$，并要求 $x'^{x'} \equiv D \pmod{p^{k+1}}$。

令 p 是一个质数，考虑先对 $n = p$ 的情况构造解。

不难发现只要联立方程 $x \equiv D \pmod p$，$x \equiv 1 \pmod{p-1}$ 即可。

假设现在已经有了 $n = p^k$ 的解，考虑将它变为 $n = p^{k+1}$ 的解。注意到令 $x \leftarrow x + t(p-1)p^k$ 后它仍是 $n = p^k$ 的解，因此可以设 $x' = x + t(p-1)p^k$，并要求 $x'^{x'} \equiv D \pmod{p^{k+1}}$。

使用欧拉定理将其改写为 $\left(x + t(p-1)p^k\right)^x \equiv D \pmod{p^{k+1}}$，再使用二项式定理展开得到 $x^x + x \cdot x^{x-1}t(p-1)p^k \equiv D \pmod{p^{k+1}}$，稍作改写得 $x \cdot x^{x-1}t(p-1) \equiv \dfrac{D - x^x}{p^k} \pmod p$。由于 D 与 n 互质，故 x 与 n 互质，因此该方程中的 t 有模 p 意义下的唯一解。

令 p 是一个质数，考虑先对 $n=p$ 的情况构造解。

不难发现只要联立方程 $x \equiv D \pmod p$，$x \equiv 1 \pmod{p-1}$ 即可。

假设现在已经有了 $n=p^k$ 的解，考虑将它变为 $n=p^{k+1}$ 的解。注意到令 $x \leftarrow x+t(p-1)p^k$ 后它仍是 $n=p^k$ 的解，因此可以设 $x'=x+t(p-1)p^k$，并要求 $x'^{x'} \equiv D \pmod{p^{k+1}}$。

使用欧拉定理将其改写为 $\left(x+t(p-1)p^k\right)^x \equiv D \pmod{p^{k+1}}$，再使用二项式定理展开得到 $x^x + x \cdot x^{x-1} t(p-1)p^k \equiv D \pmod{p^{k+1}}$，稍作改写得 $x \cdot x^{x-1} t(p-1) \equiv \dfrac{D-x^x}{p^k} \pmod p$。由于 D 与 n 互质，故 x 与 n 互质，因此该方程中的 t 有模 p 意义下的唯一解。

使用上述方法不断升幂即可，最终得到的解 x 不超过 $(p-1)p^k$。

对于一般的情况，假设当前已经求出了 x 满足 $x^x \equiv D \pmod{M}$，我们需要在模数中增加一个新质数 p，然后求出模 Mp 的解。

对于一般的情况，假设当前已经求出了 x 满足 $x^x \equiv D \pmod M$，我们需要在模数中增加一个新质数 p，然后求出模 Mp 的解。

不难想到我们可以令 $x' = x + tM\varphi(M)$，然后要求 $x'^{x'} \equiv D \pmod{pM}$。但这个方程几乎无从下手，我们略微增加限制，令 $x' = x + tM\varphi(pM)$，并使用欧拉定理将方程变为 $(x + tM\varphi(pM))^x \equiv D \pmod p$。这里的模数是 p 是因为该方程在模 M 意义下必然成立。

对于一般的情况，假设当前已经求出了 x 满足 $x^x \equiv D \pmod{M}$，我们需要在模数中增加一个新质数 p，然后求出模 Mp 的解。

不难想到我们可以令 $x' = x + tM\varphi(M)$，然后要求 $x'^{x'} \equiv D \pmod{pM}$。但这个方程几乎无从下手，我们略微增加限制，令 $x' = x + tM\varphi(pM)$，并使用欧拉定理将方程变为 $(x + tM\varphi(pM))^x \equiv D \pmod{p}$。这里的模数是 p 是因为该方程在模 M 意义下必然成立。

求解该方程的难点在于求出 D 的 x 次剩余，但当 $\gcd(x, p-1) \neq 1$ 时，D 不一定存在 x 次剩余，因此我们希望让 x 与 D 互质。具体地，我们可以先令 $x \leftarrow x + t'M\varphi(M)$ 使得新的 x 与 $p-1$ 互质。

对于一般的情况，假设当前已经求出了 x 满足 $x^x \equiv D \pmod{M}$，我们需要在模数中增加一个新质数 p，然后求出模 Mp 的解。

不难想到我们可以令 $x' = x + tM\varphi(M)$，然后要求 $x'^{x'} \equiv D \pmod{pM}$。但这个方程几乎无从下手，我们略微增加限制，令 $x' = x + tM\varphi(pM)$，并使用欧拉定理将方程变为 $(x + tM\varphi(pM))^x \equiv D \pmod{p}$。这里的模数是 p 是因为该方程在模 M 意义下必然成立。

求解该方程的难点在于求出 D 的 x 次剩余，但当 $\gcd(x, p-1) \neq 1$ 时，D 不一定存在 x 次剩余，因此我们希望让 x 与 D 互质。具体地，我们可以先令 $x \leftarrow x + t'M\varphi(M)$ 使得新的 x 与 $p-1$ 互质。

由性质 2.? 可知，只要原始的 x 与 $M\varphi(M)$ 互质，就必然可以让新的 x 与 $p-1$ 互质，此时有 $t \equiv \dfrac{D^{x^{-1} \bmod p-1} - x}{M\varphi(pM)} \pmod{p}$。同时，最终得到的 x' 也必然会与 $Mp\varphi(Mp)$ 互质，这恰好以归纳法的性质保证了前置条件成立。

另一种情况是对模数增加一个已有质数，将模 Mp^k 的解变为模 Mp^{k+1} 的解。

另一种情况是对模数增加一个已有质数，将模 Mp^k 的解变为模 Mp^{k+1} 的解。

这种情况与从 p^k 上升到 p^{k+1} 的情况类似，对于满足 $x^x \equiv D \pmod{Mp^k}$ 的 x，令 $x' \leftarrow x + tMp^k\varphi(Mp^k)$，然后对方程 $x'^{x'} \equiv D \pmod{p^{k+1}}$ 使用欧拉定理和二项式定理如法炮制即可。

另一种情况是对模数增加一个已有质数，将模 Mp^k 的解变为模 Mp^{k+1} 的解。

这种情况与从 p^k 上升到 p^{k+1} 的情况类似，对于满足 $x^x \equiv D \pmod{Mp^k}$ 的 x，令 $x' \leftarrow x + tMp^k\varphi(Mp^k)$，然后对方程 $x'^{x'} \equiv D \pmod{p^{k+1}}$ 使用欧拉定理和二项式定理如法炮制即可。

由于 $n\varphi(n)$ 是解的循环节，因此最终答案一定不超过 $n\varphi(n)$。

扩展欧几里得算法

Theorem 8.1

如果 a 是正整数，x 是实数，则 $\left\lfloor \frac{x}{a} \right\rfloor = \left\lfloor \frac{\lfloor x \rfloor}{a} \right\rfloor$。

Theorem 8.1

如果 a 是正整数，x 是实数，则 $\left\lfloor \frac{x}{a} \right\rfloor = \left\lfloor \frac{\lfloor x \rfloor}{a} \right\rfloor$。

Corollary 8.1

如果 b, c 是正整数，a 是整数，则 $\left\lfloor \frac{a}{bc} \right\rfloor = \left\lfloor \frac{\lfloor \frac{a}{b} \rfloor}{c} \right\rfloor$。

整除分块

Theorem 8.2

集合 $\left\{\left\lfloor\frac{n}{i}\right\rfloor \middle| i \in \mathbb{Z}, 1 \leq i \leq n\right\}$ 的大小是 $O(\sqrt{n})$ 的。

整除分块

Theorem 8.2

集合 $\left\{\left\lfloor \frac{n}{i} \right\rfloor \middle| i \in \mathbb{Z}, 1 \le i \le n\right\}$ 的大小是 $O(\sqrt{n})$ 的。

证明：对于 $i \le \sqrt{(n)}$，$\frac{n}{i}$ 只有 $O(\sqrt{n})$ 个；对于 $i > \sqrt{n}$，$\frac{n}{i} \le \sqrt{n}$，因此 $\left\lfloor \frac{n}{i} \right\rfloor$ 也只有 $O(\sqrt{n})$ 个。

整除分块

Theorem 8.2

集合 $\left\{\left\lfloor \frac{n}{i} \right\rfloor \middle| i \in \mathbb{Z}, 1 \le i \le n\right\}$ 的大小是 $O(\sqrt{n})$ 的。

证明：对于 $i \le \sqrt(n)$，$\frac{n}{i}$ 只有 $O(\sqrt{n})$ 个；对于 $i > \sqrt{n}$，$\frac{n}{i} \le \sqrt{n}$，因此 $\left\lfloor \frac{n}{i} \right\rfloor$ 也只有 $O(\sqrt{n})$ 个。

我们把利用这一性质来解决问题的方法称作整除分块。

例题

给定正整数 n，求 $\sum_{i=1}^{n} i\left\lfloor\frac{n}{i}\right\rfloor$。

$n \le 10^{12}$。

题解

显然我们只需要枚举每一种可能的 $\left\lfloor\frac{n}{i}\right\rfloor$，并确定其对应的 i 的范围 $[l,r]$ 即可。

题解

显然我们只需要枚举每一种可能的 $\left\lfloor \frac{n}{i} \right\rfloor$，并确定其对应的 i 的范围 $[l, r]$ 即可。

具体的，我们先找出所有的 i 满足 $\left\lfloor \frac{n}{i} \right\rfloor \neq \left\lfloor \frac{n}{i+1} \right\rfloor$，并它们从小到大排序，得到序列 $a_1, a_2, \cdots, a_m$。该序列即为 $1, 2, \cdots, \sqrt{n}, \left\lfloor \frac{n}{\lfloor \sqrt{n} \rfloor} \right\rfloor, \left\lfloor \frac{n}{\lfloor \sqrt{n} \rfloor - 1} \right\rfloor, \cdots, \left\lfloor \frac{n}{1} \right\rfloor$（如果 $\sqrt{n} = \left\lfloor \frac{n}{\lfloor \sqrt{n} \rfloor} \right\rfloor$ 则合并这两项）。

题解

显然我们只需要枚举每一种可能的 $\left\lfloor\frac{n}{i}\right\rfloor$，并确定其对应的 i 的范围 $[l, r]$ 即可。

具体的，我们先找出所有的 i 满足 $\left\lfloor\frac{n}{i}\right\rfloor \neq \left\lfloor\frac{n}{i+1}\right\rfloor$，并它们从小到大排序，得到序列 $a_1, a_2, \cdots, a_m$。该序列即为 $1, 2, \cdots, \sqrt{n}, \left\lfloor\frac{n}{\lfloor\sqrt{n}\rfloor}\right\rfloor, \left\lfloor\frac{n}{\lfloor\sqrt{n}\rfloor-1}\right\rfloor, \cdots, \left\lfloor\frac{n}{1}\right\rfloor$（如果 $\sqrt{n} = \left\lfloor\frac{n}{\lfloor\sqrt{n}\rfloor}\right\rfloor$ 则合并这两项）。

此外，对于 a_i，有 $\left\lfloor\frac{n}{a_i}\right\rfloor = a_{m+1-i}$。因此，上述算法总共只需要 $\sqrt{n} + O(1)$ 次 64 位整数除法，无论是运行效率还是代码理解难度都好于网上的部分整除分块题解。

P3935 Calculating

若 x 分解质因数结果为 $x = p_1^{k_1} p_2^{k_2} \cdots p_n^{k_n}$，令 $f(x) = (k_1+1)(k_2+1)\cdots(k_n+1)$，求 $\sum_{i=l}^{r} f(i)$ 对 998 244 353 取模的结果。

$1 \le l \le 10^{14}, 1 \le r \le 1.6 \times 10^{14}$。

题解

设 $g(n) = \sum_{i=1}^{n} f(i)$，则答案即为 $g(r) - g(l-1)$。

题解

设 $g(n) = \sum_{i=1}^{n} f(i)$，则答案即为 $g(r) - g(l-1)$。

注意到 $f(x)$ 等于 x 的约数个数，因此可以换一个方式计算 $g(x)$。枚举每个因子 i，则 i 的倍数有 $\left\lfloor \frac{n}{i} \right\rfloor$ 个。因此，$g(n) = \sum_{i=1}^{n} \left\lfloor \frac{n}{i} \right\rfloor$，整除分块计算即可。

ARC068E Snuke Line

有一趟列车有 $M+1$ 个车站，从 0 到 M 编号。有 N 种商品，第 i 种只在编号 $[l_i, r_i]$ 的车站出售。一辆列车有一个预设好的系数 d，从 0 出发，只会在 d 的倍数车站停车。对于 d 从 1 到 M 的列车，求最多能买到多少种商品。

$1 \le n \le 3 \times 10^5, 1 \le M \le 10^5, 1 \le l_i \le r_i \le M$。

题解

首先，第 i 种商品能在 d 号列车上买到当且仅当 $\left\lfloor \frac{l_i - 1}{d} \right\rfloor < \left\lfloor \frac{r_i}{d} \right\rfloor$。

题解

首先，第 i 种商品能在 d 号列车上买到当且仅当 $\left\lfloor \frac{l_i - 1}{d} \right\rfloor < \left\lfloor \frac{r_i}{d} \right\rfloor$。

对于每一个 i，对 $l_i - 1$ 和 r_i 分别整除分块。$\left\lfloor \frac{l_i - 1}{d} \right\rfloor$ 和 $\left\lfloor \frac{r_i}{d} \right\rfloor$ 的值分别把 d 划分成了 $O(\sqrt{n})$ 个区间，二者合并之后还是 $O(\sqrt{n})$ 个区间。每个区间可能会使得一段连续的 d 对应的答案增加一，用差分维护即可。

积性函数

Definition 8.1 (数论函数)

定义域为正整数的函数被称为数论函数。

积性函数

Definition 8.1 (数论函数)

定义域为正整数的函数被称为数论函数。

Definition 8.2 (积性函数)

如果数论函数 $f(n)$ 对于任意 $p,q\in\mathbb{N}^+,\gcd(p,q)=1$ 都有 $f(pq)=f(p)f(q)$ 成立，则称 f 为积性函数。

例如，$f(n)=n^k$，$f(n)=\varphi(n)$，$f(n)=\sum\limits_{d|n} d^k$ 都是积性函数。

积性函数

Definition 8.1 (数论函数)

定义域为正整数的函数被称为数论函数。

Definition 8.2 (积性函数)

如果数论函数 $f(n)$ 对于任意 $p, q \in \mathbb{N}^{+}, \gcd(p, q) = 1$ 都有 $f(pq) = f(p)f(q)$ 成立，则称 f 为积性函数。

例如，$f(n) = n^k$，$f(n) = \varphi(n)$，$f(n) = \sum_{d|n} d^k$ 都是积性函数。

对于积性函数，我们只需要知道它在质数幂处的值即可确定整个函数，具体实现可以借助线性筛算法。

狄利克雷前缀和

Problem (狄利克雷前缀和)

给定数论函数 $f(x)$ 在 $1 \sim n$ 处的值。求函数 $F(x) = \sum_{i|x} f(i)$ 在 $1 \sim n$ 处的值。$n \leq 2 \times 10^7$。

狄利克雷前缀和

Problem (狄利克雷前缀和)

给定数论函数 $f(x)$ 在 $1 \sim n$ 处的值。求函数 $F(x) = \sum_{i|x} f(i)$ 在 $1 \sim n$ 处的值。

$n \leq 2 \times 10^7$。

朴素算法的复杂度是 $O(n \log n)$ 的，我们希望更优。

狄利克雷前缀和

Problem (狄利克雷前缀和)

给定数论函数 $f(x)$ 在 $1 \sim n$ 处的值。求函数 $F(x) = \sum_{i|x} f(i)$ 在 $1 \sim n$ 处的值。

$n \leq 2 \times 10^7$。

朴素算法的复杂度是 $O(n \log n)$ 的，我们希望更优。

考虑对每一个质数分别求前缀和，即依次考虑每一个质数 p_i，令新的 $f(x)$ 等于 $\sum_{p^i|x} f\left(\frac{x}{p^i}\right)$。该算法的正确性仍然是对的，但复杂度降至 $O(n \log \log n)$。

莫比乌斯反演

Problem (莫比乌斯反演)

给定数论函数 $F(x)$ 在 $1 \sim n$ 处的值。有关系 $F(x) = \sum_{i|x} f(i)$ 成立，求函数 $f(x)$ 在 $1 \sim n$ 处的值。

$n \leq 2 \times 10^7$。

莫比乌斯反演

Problem (莫比乌斯反演)

给定数论函数 $F(x)$ 在 $1 \sim n$ 处的值。有关系 $F(x) = \sum_{i|x} f(i)$ 成立，求函数 $f(x)$ 在 $1 \sim n$ 处的值。
$n \leq 2 \times 10^7$。

显然莫比乌斯反演是狄利克雷前缀和的逆变换，因此考虑上一页中算法的逆变换。依次考虑每一个质数 p_i，令新的 $F(x)$ 等于 $F(x) - [p_i \mid x]F(\frac{x}{p_i})$。复杂度为 $O(n \log\log n)$。

莫比乌斯函数

把上一页的算法展开，可以得到 $f(x)=\sum_{i|x}F(i)\mu(\frac{x}{i})$。其中 $\mu(x)$ 是莫比乌斯函数，定义如下：

$$\mu(n)=\begin{cases}1 & n=1\\ 0 & n\ \text{有平方因子}\\ (-1)^p & n\ \text{是}p\ \text{个不同质数的乘积}\end{cases}$$

莫比乌斯函数

把上一页的算法展开，可以得到 $f(x)=\sum_{i|x}F(i)\mu(\frac{x}{i})$。其中 $\mu(x)$ 是莫比乌斯函数，定义如下：

$$\mu(n)=\begin{cases}1 & n=1\\ 0 & n\text{ 有平方因子}\\ (-1)^p & n\text{ 是}p\text{ 个不同质数的乘积}\end{cases}$$

容易发现 $\mu(x)$ 也是积性函数。

莫比乌斯函数

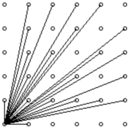
把上一页的算法展开，可以得到 $f(x)=\sum_{i|x}F(i)\mu(\frac{x}{i})$。其中 $\mu(x)$ 是莫比乌斯函数，定义如下：

$$\mu(n)=\begin{cases}1 & n=1\\ 0 & n\text{ 有平方因子}\\ (-1)^p & n\text{ 是}p\text{ 个不同质数的乘积}\end{cases}$$

容易发现 $\mu(x)$ 也是积性函数。
常见的莫比乌斯反演：$\varphi(n)=\sum_{d|n}d\mu(\frac{n}{d})$，$[n=1]=\sum_{d|n}\mu(d)$。

P2158 [SDOI2008] 仪仗队

作为体育委员，C 君负责这次运动会仪仗队的训练。仪仗队是由学生组成的 $N \times N$ 的方阵，为了保证队伍在行进中整齐划一，C 君会跟在仪仗队的左后方，根据其视线所及的学生人数来判断队伍是否整齐（如下图）。



现在，C 君希望你告诉他队伍整齐时能看到的学生人数。

$1 \leq N \leq 40000$。

题解

忽略最左边一行和最下方一列，则答案为 $\sum_{i}^{n-1}\sum_{j}^{n-1}[\gcd(i,j)=1]$。

题解

忽略最左边一行和最下方一列，则答案为 $\sum\limits_{i}^{n-1}\sum\limits_{j}^{n-1}[\gcd(i,j)=1]$。

$$\sum_{i}^{n}\sum_{j}^{m}[\gcd(i,j)=1]$$

$$=\sum_{i}^{n}\sum_{j}^{m}\sum_{t|\gcd(i,j)}\mu(t)$$

$$=\sum_{i}^{n}\sum_{j}^{m}\sum_{t|i,t|j}\mu(t)$$

$$=\sum_{t=1}^{\min(n,m)}\left\lfloor\frac{n}{t}\right\rfloor\left\lfloor\frac{m}{t}\right\rfloor\mu t$$

SP26017 GCDMAT - GCD OF MATRIX

给定 n, m，再在每组数据中给定不大于 n 的整数 i_1, j_1 和不大于 m 的整数 i_2, j_2，求出 $\sum_{i=i_1}^{i_2}\sum_{j=j_1}^{j_2}\gcd(i,j)$ 的值。T 组数据。

$1 \le n, m \le 5 \times 10^4$，$1 \le i_1, j_1 \le n$，$1 \le i_2, j_2 \le m$，$1 \le T \le 500$。

题解

显然可以转化为若干次 $\sum_{i}^{n}\sum_{j}^{m}\gcd(i,j)$ 的查询。

题解

显然可以转化为若干次 $\sum_i^n \sum_j^m \gcd(i,j)$ 的查询。

$$\sum_i^n \sum_j^m \gcd(i,j)$$

$$=\sum_i^n \sum_j^m \sum_{t|\gcd(i,j)} \varphi(t)$$

$$=\sum_i^n \sum_j^m \sum_{t|i,t|j} \varphi(t)$$

$$=\sum^{\min(n,m)} \left\lfloor \frac{n}{t} \right\rfloor \left\lfloor \frac{m}{t} \right\rfloor \varphi(t)$$

P3327 [SDOI2015] 约数个数和

设 $d(x)$ 为 x 的约数个数，给定 n, m，求

$$\sum_{i=1}^{n}\sum_{j=1}^{m} d(ij)$$

多测，T 组数据。
$1 \le T, n, m \le 50000$。

题解

首先需要了解 d 函数的一个特殊性质：

$$d(ij) = \sum_{x|i} \sum_{y|j} [\gcd(x, y) = 1]$$

题解

首先需要了解 d 函数的一个特殊性质：

$$d(ij) = \sum_{x|i} \sum_{y|j} [\gcd(x, y) = 1]$$

$$\sum_{i}^{n} \sum_{j}^{m} d(ij)$$

$$= \sum_{i}^{n} \sum_{j}^{m} \sum_{x|i} \sum_{y|j} [\gcd(x, y) = 1]$$

$$= \sum_{x=1}^{n} \sum_{y=1}^{m} \left\lfloor \frac{n}{x} \right\rfloor \left\lfloor \frac{m}{y} \right\rfloor [\gcd(x, y) = 1]$$

$$
\begin{aligned}
&= \sum_{x=1}^{n} \sum_{y=1}^{m} \left\lfloor \frac{n}{x} \right\rfloor \left\lfloor \frac{m}{y} \right\rfloor [\gcd(x, y) = 1] \\
&= \sum_{t=1}^{\min(n,m)} \mu(t) \sum_{x=1}^{\frac{n}{t}} \sum_{y=1}^{\frac{m}{t}} \left\lfloor \frac{n}{tx} \right\rfloor \left\lfloor \frac{n}{ty} \right\rfloor \\
&= \sum_{t=1}^{\min(n,m)} \mu(t) \left(\sum_{x=1}^{\frac{n}{t}} \left\lfloor \frac{\left(\frac{n}{t}\right)}{x} \right\rfloor \right) \left(\sum_{y=1}^{\frac{m}{t}} \left\lfloor \frac{\left(\frac{m}{t}\right)}{y} \right\rfloor \right)
\end{aligned}
$$

$$=\sum_{x=1}^{n}\sum_{y=1}^{m}\left\lfloor\frac{n}{x}\right\rfloor\left\lfloor\frac{m}{y}\right\rfloor[\gcd(x,y)=1]$$

$$=\sum_{t=1}^{\min(n,m)}\mu(t)\sum_{x=1}^{\frac{n}{t}}\sum_{y=1}^{\frac{m}{t}}\left\lfloor\frac{n}{tx}\right\rfloor\left\lfloor\frac{n}{ty}\right\rfloor$$

$$=\sum_{t=1}^{\min(n,m)}\mu(t)\left(\sum_{x=1}^{\frac{n}{t}}\left\lfloor\frac{\left(\frac{n}{t}\right)}{x}\right\rfloor\right)\left(\sum_{y=1}^{\frac{m}{t}}\left\lfloor\frac{\left(\frac{m}{t}\right)}{y}\right\rfloor\right)$$

对所有的 x 预处理 $\sum_{i=1}^{x}\left\lfloor\frac{x}{i}\right\rfloor$，使用整除分块即可 $O(\sqrt{n})$ 完成单次查询。

前置结论的证明

Theorem 8.3

$$d(ij) = \sum_{x|i} \sum_{y|j} [\gcd(x, y) = 1]$$

前置结论的证明

Theorem 8.3

$$d(ij) = \sum_{x|i} \sum_{y|j} [\gcd(x,y) = 1]$$

对于 $k \mid ij$，考虑每一个质数 p，如果 i 中有因子 p^a，j 中有因子 p^b，k 中有因子 p^c：

- 当 $c \le a$ 时，在 x 中添加因子 p^c。
- 当 $c > a$ 时，在 y 中添加因子 p^{c-a}。

前置结论的证明

Theorem 8.3

$$d(ij)=\sum_{x|i}\sum_{y|j}[\gcd(x,y)=1]$$

对于 $k\mid ij$，考虑每一个质数 p，如果 i 中有因子 p^a，j 中有因子 p^b，k 中有因子 p^c：

- 当 $c\le a$ 时，在 x 中添加因子 p^c。
- 当 $c>a$ 时，在 y 中添加因子 p^{c-a}。

显然这样构造出的 x,y 二元组互不相同，且有 $x\mid i, y\mid j, \gcd(x,y)=1$ 成立。而给定互质的 x,y，也反推出 k 的值，因此原式成立。