基础数论

梦熊人才联盟

2024.8

1 乘法逆元

2 exgcd

3 CRT

4 lucas

5 BSGS

6 整除分块

1 乘法逆元

2 exgcd

3 CRT

4 lucas

5 BSGS

6 整除分块

1. 乘法逆元
2. exgcd
3. CRT
4. lucas
5. BSGS
6. 整除分块

费马小定理

乘法逆元：对于 a, p，求出 x 使得 $ax \equiv 1 \pmod p$。

若 p 为素数，$\gcd(a,\ p) = 1$，则 $a^{p-1} \equiv 1 \pmod p$。

用于求乘法逆元：$a^{p-2} \equiv a^{-1} \pmod p$。

费马小定理

设 $A=\{1,2,\dots,p-1\}$，则 $f(x)=ax \bmod p$ 为 A 到自身的一个双射。

证明：若 $x,y\in A, f(x)=f(y)$，则 $a(x-y)\equiv 0 \pmod p$。由 $\gcd(a,p)=1$ 可推出 $x=y$。

所以 $\prod_{i=1}^{p-1} i=\prod_{i=1}^{p-1} a\cdot i$，故 $a^{p-1}\equiv 1 \pmod p$。

乘法逆元

若 p 不为素数，乘法逆元可以通过 exgcd 等方式求得（根据同余方程，逆元存在需满足 $\gcd(a,p)=1$）。

线性求 n 个数的逆元：记 $s_i=\prod_{j=1}^{i}a_j$，求出 s_n 的逆元 v_n，通过 $v_i\times a_i$ 可以求出 s_{i-1} 的逆元 v_{i-1}，于是 $a_i^{-1}=s_{i-1}\cdot v_i$。

1. 乘法逆元
2. exgcd
3. CRT
4. lucas
5. BSGS
6. 整除分块

exgcd

算术基本定理：一个正整数总是可以被写成 $p_1^{k_1}p_2^{k_2}\cdots p_c^{k_c}$。

$\gcd(x,y)=p_1^{\min\{k_1,k_1'\}}p_2^{\min\{k_2,k_2'\}}\cdots p_c^{\min\{k_c,k_c'\}}$。

$\mathrm{lcm}(x,y)=p_1^{\max\{k_1,k_1'\}}p_2^{\max\{k_2,k_2'\}}\cdots p_c^{\max\{k_c,k_c'\}}$。

约数和：$\prod_{i=1}^{c}(\sum_{j=0}^{k_i}p_i^j)$。

P4397 [JLOI2014] 聪明的燕姿

k 组数据，每次给定一个数 S，求出所有约数和等于 S 的那些数。

$k \leq 100, S \leq 2 \times 10^9$。

exgcd

辗转相除法：$\gcd(x, y) = \gcd(y, x \bmod y)$。

考虑证明 (x, y) 和 $(y, x \bmod y)$ 的公约数集合相等：
设 $x = ay + b$。若 $d \mid x, d \mid y$，$b = x - ay, \frac{b}{d} = \frac{x}{d} - a\frac{y}{d}$，故 $\frac{b}{d}$ 为整数。
若 $d \mid y, d \mid (x \bmod y)$，$\frac{x \bmod y}{d} = \frac{x}{d} - a\frac{y}{d}$，故 $\frac{x}{d}$ 为整数。

```
int gcd(int x,int y) {
    if(!y) {
        return x;
    }
    return gcd(y,x%y);
}
```

P1072 [NOIP 2009 提高组] Hankson 的趣味题

给定 a_0, a_1, b_0, b_1，求有多少个 x 满足 $\gcd(x, a_0) = a_1$ 且 $\operatorname{lcm}(x, b_0) = b_1$。

$1 \le T \le 2000$，$1 \le a_0, a_1, b_0, b_1 \le 10^9$。

exgcd

裴蜀定理：设 a, b 为不全为 0 的整数，$d = \gcd(a, b)$。则对于所有 x, y，有 $d \mid ax + by$，且存在 x, y 使得 $ax + by = d$。

对于后一点的证明，考虑方程 $\frac{a}{d}x + \frac{b}{d}y = 1$，我们使用 exgcd 算法直接构造。

exgcd

exgcd 是用来解决一类形如 $ax + by = 1, \gcd(a, b) = 1$ 的二元（x, y 二元）不定方程整数解的算法。

如果解出来这样的 x, y，我们可以发现，我们相当于顺便算出来 $a \bmod b$ 的逆元，和 $b \bmod a$ 的逆元。（等式两边同时对 a,b 取 mod）。
我们仿照欧几里得算法来解决这个问题。

exgcd

不妨假设 $a > b$:

$$ax + by = 1 \implies (a \bmod b)x + \lfloor \frac{a}{b} \rfloor bx + by = 1$$
$$\implies (a \bmod b)x + b(\lfloor \frac{a}{b} \rfloor x + y) = 1$$

令 $u = x, v = \lfloor \frac{a}{b} \rfloor x + y$，那么我们只要解出来 $(a \bmod b)u + bv = 1$ 这个方程，就可以反推出 $ax + by = 1$ 方程的解了。这个形式跟求 gcd 非常类似，一直递归到 $a = 1, b = 0$ 时我们就能确定当前的解为 $x = 1, y = 0$，再不断去反推上一层的解。

exgcd

```
void exgcd(int a,int b,int &x,int &y){
    if(b==0){
        x=1,y=0;return ;
    }
    exgcd(b,a%b,y,x);y-=(a/b)*x;
}
```

我们可以通过归纳来证明，通过 exgcd 求出来的解，一定是 $|x|,|y|$ 最小的那两组解其中之一。我们也不用在 a,b 都是 int 范围下，将答案开 long long，因为每一次递归都会满足该性质。

解出一组 $ax+by=d$ 的特解后，方程的通解可以表示为 $x'=x+k\cdot\frac{b}{d}, y'=y-k\cdot\frac{a}{d}$。

P1516 青蛙的约会

两只青蛙在长 L 米的环上朝同一方向跳跃，初始位置分别离远点距离 x, y 米，两只青蛙一次分别会跳 m, n 米，问它们跳了多少次之后会相遇。
$x, y, m, n, L \leq 2 \times 10^9$。

1 乘法逆元

2 exgcd

3 CRT

4 lucas

5 BSGS

6 整除分块

CRT

- CRT：中国剩余定理，是来解同余方程组的算法。

CRT

- CRT：中国剩余定理，是来解同余方程组的算法。
- 假设现在有 n 个形如：$x \equiv a_i \pmod{m_i}$，并且满足对于任意 $1 \le i, j \le n, i \neq j, \gcd(m_i, m_j) = 1$。

CRT

- CRT：中国剩余定理，是来解同余方程组的算法。
- 假设现在有 n 个形如：$x \equiv a_i \pmod{m_i}$，并且满足对于任意 $1 \le i,j \le n, i \ne j, \gcd(m_i, m_j) = 1$。
- 我们首先考虑到，设 $M = \prod m_i$，那么如果 x 是解的话，$x + M$ 必然也是解。而 $x + t$，$1 \le t < M$ 必然不可能是解。所以最后答案的形式一定是 $x \equiv t \pmod{M}$。这也启示我们，只要找到一个**特解**，就能找到所有的解。

CRT

- CRT：中国剩余定理，是来解同余方程组的算法。
- 假设现在有 n 个形如：$x \equiv a_i \pmod{m_i}$，并且满足对于任意 $1 \leq i,j \leq n, i \neq j, \gcd(m_i, m_j) = 1$。
- 我们首先考虑到，设 $M = \prod m_i$，那么如果 x 是解的话，$x + M$ 必然也是解。而 $x + t$，$1 \leq t < M$ 必然不可能是解。所以最后答案的形式一定是 $x \equiv t \pmod{M}$。这也启示我们，只要找到一个**特解**，就能找到所有的解。
- 设 u_i 表示 M/m_i 在 m_i 意义下的逆元，即 $u_i(M/m_i) \equiv 1 \pmod{m_i}$。那么我们找到一个特解 $x_0 = \sum_{i=1}^{n} a_i u_i (M/m_i)$。

CRT

- CRT：中国剩余定理，是来解同余方程组的算法。
- 假设现在有 n 个形如：$x \equiv a_i \pmod{m_i}$，并且满足对于任意 $1 \leq i,j \leq n, i \neq j, \gcd(m_i, m_j) = 1$。
- 我们首先考虑到，设 $M = \prod m_i$，那么如果 x 是解的话，$x + M$ 必然也是解。而 $x + t$, $1 \leq t < M$ 必然不可能是解。所以最后答案的形式一定是 $x \equiv t \pmod{M}$。这也启示我们，只要找到一个**特解**，就能找到所有的解。
- 设 u_i 表示 M/m_i 在 m_i 意义下的逆元，即 $u_i(M/m_i) \equiv 1 \pmod{m_i}$。那么我们找到一个特解 $x_0 = \sum_{i=1}^{n} a_i u_i (M/m_i)$。
- 注意，在这里，我们把 $u_i, a_i, M/m_i$ 都看成普通的整数，他们的乘法也只不过是普通乘法，最后再对 M 取模。

CRT

特解：$x_0 = (\sum_{i=1}^{n} a_i u_i (M/m_i)) \bmod M$。

正确性：当 $j \neq i$ 时，$(M/m_j) \equiv 0 \pmod{m_i}$，故 $a_j u_j (M/m_j) \equiv 0 \pmod{m_i}$。

$$x_0 \equiv \sum_{i=1}^{n} a_i u_i (M/m_i) \equiv a_i u_i (M/m_i) \equiv a_i \pmod{m_i}$$

[TJOI2009] 猜数字

题意：现有两组数字，每组 k 个。
第一组中的数字分别用 $a_1, a_2, \cdots, a_k$ 表示，第二组中的数字分别用 $b_1, b_2, \cdots, b_k$ 表示。
其中第二组中的数字是两两互素的。求最小的 $n \in \mathbb{N}$，满足对于 $\forall i \in [1, k]$，有 $b_i | (n - a_i)$。
$1 \le k \le 10$，$|a_i| \le 10^9$，$1 \le b_i \le 6 \times 10^3$，$\prod_{i=1}^{k} b_i \le 10^{18}$。

[TJOI2009] 猜数字

实际上就是让我们去解 $n \equiv a_i \pmod{b_i}$ 这个同余方程组。
具体的，我们使用 exgcd 去解满足 $u_i(M/m_i) \equiv 1 \pmod{m_i}$ 的 u_i。
特别需要注意，在解出 u_i 后，所有乘法与加法都是在 $\bmod M$ 意义下而不是 $\bmod m_i$ 意义下。

excrt

扩展 CRT，如果不保证 $\forall i \neq j, \gcd(m_i, m_j) = 1$ 的话，我们可能得不到一个较为优美，可以被称为“定理”的形式。
但我们仍可以在 exgcd 的基础上，得到一个算法。
我们考虑合并两个同余式：$x \equiv a_0 \pmod{b_0}$，$x \equiv a_1 \pmod{b_1}$。
我们假设 $x = kb_0 + a_0$, 那么我们将问题转化成一个关于 k 的同余方程：$kb_0 + a_0 \equiv a_1 \pmod{b_1}$。
回忆一下，$ax \equiv b \pmod{c}$ 的同余方程的解法是，先考察 $b \mid \gcd(a, c)$，然后将 a,b,c 全部除掉 $\gcd(a, c)$，然后 exgcd 求逆元。

excrt

$k \equiv k_0 \pmod{\frac{b_1}{\gcd(b_0,b_1)}}$，不妨设 $k = k_0 + t\frac{b_1}{\gcd(b_0,b_1)}$。
那么 $x = b_0(k_0 + t\frac{b_1}{\gcd(b_0,b_1)}) + a_0 = b_0 k_0 + a_0 + t\frac{b_0 b_1}{\gcd(b_0,b_1)}$。
可以发现此时 $x \equiv b_0 k_0 + a_0 \pmod{\mathrm{lcm}(b_0, b_1)}$ 还是非常深刻的。

屠龙勇士

题目大意：n 条龙血量分别为 a_i，恢复能力为 p_i，死亡后会掉落一把攻击力为 t_i 的剑，初始有 m 把剑，攻击力分别为 b_i。
从第一条龙开始打，每次挑一把攻击力小于龙血量的攻击力最大的剑，如果没有则用攻击力最小的那把剑。我们设用攻击力为 c_i 的剑打的第 i 条龙。
求一个最小的天数 x，使得对于每一个 i，$p_i | xc_i$。
$1 \leq n, m \leq 10^5$，满足 $a_i \leq p_i$, $\mathrm{lcm}(p_i) \leq 10^{12}$。

屠龙勇士

- 首先，c_i 可以用一个 muliset 来维护出来

屠龙勇士

- 首先，c_i 可以用一个 muliset 来维护出来
- 接下来，我们考虑实际上只需要解出这个同余方程组，其中第 i 条方程为：$c_i x \equiv a_i \pmod{p_i}$。

屠龙勇士

- 首先，c_i 可以用一个 muliset 来维护出来
- 接下来，我们考虑实际上只需要解出这个同余方程组，其中第 i 条方程为：$c_i x \equiv a_i \pmod{p_i}$。
- 类似的，我们依旧先将每一个方程左右两边，同时除掉 $\gcd(c_i, p_i)$，然后同乘 c_i' 的逆元。然后我们就将该问题转化成 excrt 的模板题了。

屠龙勇士

- 首先，c_i 可以用一个 muliset 来维护出来
- 接下来，我们考虑实际上只需要解出这个同余方程组，其中第 i 条方程为：$c_i x \equiv a_i \pmod{p_i}$。
- 类似的，我们依旧先将每一个方程左右两边，同时除掉 $\gcd(c_i, p_i)$，然后同乘 c_i' 的逆元。然后我们就将该问题转化成 excrt 的模板题了。
- 最后要注意一下，我们需要保证 $xc_i \geq p_i$，将解出来的解平移即可。

1. 乘法逆元
2. exgcd
3. CRT
4. lucas
5. BSGS
6. 整除分块

lucas 定理

加法原理：如果完成一件事情有 n 类方式，每类方式有 a_i 种方法，则完成这件事有 $\sum_{i=1}^{n} a_i$ 种方法。
乘法原理：如果完成一件事情有 n 个步骤，每个步骤有 a_i 种方法，则完成这件事有 $\prod_{i=1}^{n} a_i$ 种方法。

排列数：$\mathrm{A}_n^m = n(n-1)(n-2)\cdots(n-m+1) = \frac{n!}{(n-m)!}$。表示从 n 个不同元素中选出 m 个元素任意排列的方案数。
组合数：$\mathrm{C}_n^m = \binom{n}{m} = \frac{A_n^m}{m!}$。表示从 n 个不同元素中选出 m 个元素的方案数。

lucas 定理

组合数递推：$\binom{n}{m} = \binom{n-1}{m} + \binom{n-1}{m-1}$。

二项式定理：$(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} a^i b^{n-i}$。

插板法：有 n 个完全相同的元素，将其分为 k 组，保证每组至少有一个元素，方案数为 $\binom{n-1}{k-1}$。如果每组可以为空则方案数为 $\binom{n+k-1}{k-1}$。

P2638 安全系统

有 a 个黑球 b 个白球，相同颜色的球完全一样，将这些球任意放进 n 个不同的袋子（可以有球不放进任何袋子，袋子可以为空），求方案数。

做 $1 \le n, a, b \le 10^5$，对 998244353 取模。

lucas 定理

如何求组合数？线性预处理，$\mathcal{O}(1)$ 查询。

```
auto qpow = [&](long long a, long long b) -> long long {
    long long ret=1;
    while (b){
        if (b & 1) ret = ret * a % mod;
        a = a * a % mod; b >>= 1;
    }
    return ret;
};
f[0] = 1, finv[0] = 1;
for (int i = 1; i <= N; i++) f[i] = f[i - 1] * i % mod;
finv[N] = qpow(f[N], mod - 2);
for (int i = N - 1; i >= 1; i--)
    finv[i] = finv[i + 1] * (i + 1) % mod;
auto binom=[&](int n, int m) -> long long {
    if(n < m || m < 0) return 0;
    return f[n] * finv[m] % mod * finv[n - m] %mod;
};
```

lucas 定理

但是很尴尬的一点是，如果 p 比较小的话，那么 f 数组，和 finv 数组就会出现许许多多的 0, 我们并不希望这样，我们当然可以通过记录 p 出现次数来解决，但是 lucas 定理提供了一种更为优美的处理办法。

$$\binom{n}{m} \bmod p = \binom{n \bmod p}{m \bmod p} \binom{\lfloor n/p \rfloor}{\lfloor m/p \rfloor} \bmod p$$

证明可以考虑对比关于 x 的多项式：$(1+x)^p \bmod p = 1 + x^p$。那么 $(1+x)^n \bmod p = (1+x^p)^{n/p}(1+x)^{n \bmod p} \bmod p$，那么我们对比多项式左右 x^m 项的系数，那么就可以得到 lucas 定理。

[SDOI2010] 古代猪文

题目大意：给定 N, G，求出：$G^{\sum_{d|N}\binom{N}{d}} \bmod 999911659$。
其中，$n, g \le 10^9$。

[SDOI2010] 古代猪文

999911659 为质数，我们可以对原式使用费马小定理，只需要计算 $\sum_{d|N} \binom{N}{d} \bmod 999911658$。

但是 $999911658 = 2 \times 3 \times 4679 \times 35617$ 是一个大合数，我们没有办法直接计算出组合数，所以考虑 CRT，我们分别计算出 $\binom{N}{d} \bmod 2, 3, 4679, 35617$ 的余数，最后通过 CRT 合并起来，就可以得到 $\bmod 999911658$ 的结果。

而 $\binom{N}{d} \bmod p$ 就可以通过 lucas 定理计算出来。

1. 乘法逆元
2. exgcd
3. CRT
4. lucas
5. **BSGS**
6. 整除分块

BSGS

求解离散对数，即，找到满足 $b^x \equiv c \pmod{p}$ 的 x。$p \in \mathbb{P}$ 故一定有解。

BSGS

求解离散对数，即，找到满足 $b^x \equiv c \pmod p$ 的 x。$p \in \mathbb{P}$ 故一定有解。
考虑分块。假设 $B = \sqrt{N}$，将 b^x 看作 b^{iB+j}，其中 $0 \le i \le B, 0 \le j \le B$。枚举 i，计算 b^{iB} 然后查表算是否存在 j。
我们可以通过计算 $c \times b^{-iB}$ 和哈希表的方式，来使上述算法复杂度达到严格 $\mathcal{O}(\sqrt{n})$

[SDOI2013] 随机数生成器

有一个随机数生成器 $x_{i+1} = ax_i + b \bmod p$，找到第一次生成 t 的时刻。多测保证 p 是质数。
$p \leq 10^9$，数据组数不超过 50。无解输出 -1。

[SDOI2013] 随机数生成器

来推一下数列的通项（$a \neq 1$）：

$$x_{i+1} + \frac{b}{a-1} = a(x_i + \frac{b}{a-1})$$
$$x_k + \frac{b}{a-1} = a^k(x_0 + \frac{b}{a-1})$$
$$x_k = a^k(x_0 + \frac{b}{a-1}) - \frac{b}{a-1}$$

整理过后，我们可以得到一个，形如 $a^k = c$，$c = \frac{t+\frac{b}{a-1}}{x_0+\frac{b}{a-1}}$ 的方程。

使用 bsgs 即可解出答案。

BSGS 的一些应用

bsgs 是一种根号分治的思想，它的应用场景很多很多，不止于求离散对数，下面一道例题也运用了这种思想。

Guess Cycle Length

来源链接：https://codeforces.com/gym/104090/problem/I
题目大意：交互提，你有一个长度为 n 的环，环上每个点都有一个编号，这些编号构成了一个 1 到 n 的排列。$n \leq 10^9$。
初始有一枚棋子在 1 号点，你需要通过至多 10^4 次操作来得到 n 的值。
操作形如：给定一个不超过 10^9 的正整数 x，假设现在棋子在 u 号点，将其在环上移动 x 次，然后返回所在点的标号。

Guess Cycle Length

首先考虑一个询问次数为 $\mathcal{O}(\sqrt{n})$ 的算法。
我们考虑，如果我们操作过后，得到的标号和之前的标号重复了。那么我们可以计算出，环长一定是我们走过的长度的因子。我们至少也可以用 $\mathcal{O}(\log L)$ 次询问的代价来确定出 n。
我们取一个阈值，设为 $B \geq \sqrt{L}$，然后我们发现任意一个 n 可以被表示成 $n = kB + c$。仿照 bsgs，先走 B 次，每次长度为 B，然后再走 B 次，每次长度为 1。此时必然能找到重复的标号。
该算法的询问次数为 $2 \times \sqrt{10^9}$ 不足通过此题。

Guess Cycle Length

考虑这样一个随机化，我们随机问 k 个数，取这些数的最大值 m，m 应该不会距离 $\frac{k}{k+1}n$ 太远。
考虑这样一个算法：

- 随机问 $n^{1/3}$ 个值，取最大值为 m。

Guess Cycle Length

考虑这样一个随机化，我们随机问 k 个数，取这些数的最大值 m，m 应该不会距离 $\frac{k}{k+1}n$ 太远。

考虑这样一个算法：

- 随机问 $n^{1/3}$ 个值，取最大值为 m。
- 走 $n^{1/3}$ 次，长度为 1 的步。

Guess Cycle Length

考虑这样一个随机化，我们随机问 k 个数，取这些数的最大值 m，m 应该不会距离 $\frac{k}{k+1}n$ 太远。
考虑这样一个算法：

- 随机问 $n^{1/3}$ 个值，取最大值为 m。
- 走 $n^{1/3}$ 次，长度为 1 的步。
- 走一次，长度为 m 的步。

Guess Cycle Length

考虑这样一个随机化，我们随机问 k 个数，取这些数的最大值 m，m 应该不会距离 $\frac{k}{k+1}n$ 太远。
考虑这样一个算法：

- 随机问 $n^{1/3}$ 个值，取最大值为 m。
- 走 $n^{1/3}$ 次，长度为 1 的步。
- 走一次，长度为 m 的步。
- 走 $n^{1/3}$ 次，长度为 $n^{1/3}$ 的步。

Guess Cycle Length

考虑这样一个随机化，我们随机问 k 个数，取这些数的最大值 m，m 应该不会距离 $\frac{k}{k+1}n$ 太远。
考虑这样一个算法：

- 随机问 $n^{1/3}$ 个值，取最大值为 m。
- 走 $n^{1/3}$ 次，长度为 1 的步。
- 走一次，长度为 m 的步。
- 走 $n^{1/3}$ 次，长度为 $n^{1/3}$ 的步。
- 我们可以理解成，对 $n-m \approx \frac{1}{k+1}n$ 的长度进行 bsgs，然后将 k 取成 $n^{1/3}$ 进行平衡。

Guess Cycle Length

考虑这样一个随机化，我们随机问 k 个数，取这些数的最大值 m，m 应该不会距离 $\frac{k}{k+1}n$ 太远。
考虑这样一个算法：

- 随机问 $n^{1/3}$ 个值，取最大值为 m。
- 走 $n^{1/3}$ 次，长度为 1 的步。
- 走一次，长度为 m 的步。
- 走 $n^{1/3}$ 次，长度为 $n^{1/3}$ 的步。
- 我们可以理解成，对 $n-m\approx\frac{1}{k+1}n$ 的长度进行 bsgs，然后将 k 取成 $n^{1/3}$ 进行平衡。
- 具体实现的时候，我们可以将上面的 $n^{1/3}$ 全取成 3333，此时错误率（m 距离 $\frac{1}{k+1}n$ 太远）已经极低了。

1 乘法逆元

2 exgcd

3 CRT

4 lucas

5 BSGS

6 整除分块

整除分块

我们来考虑一下整除函数的性质，$\left[\frac{n}{i}\right], 1 \le i \le n$。其含义是小于 $\frac{n}{i}$ 的最大整数。
我们不难发现 $\left[\frac{n}{i}\right]$ 是关于 i 单调减小的。并且当 $i \ge \sqrt{n}$ 时，$\left[\frac{n}{i}\right] \le \sqrt{n}$ 只有 $\sqrt{n}$ 种不同取值，同时当 $i \le \sqrt{n}$ 时，$\left[\frac{n}{i}\right]$ 也只会有 $\sqrt{n}$ 种不同取值。
所以我们得到一个惊人的结论，$\left[\frac{n}{i}\right], 1 \le i \le n$ 只有 $2\sqrt{n}$ 种不同取值，并且他们还是成段出现的（单调性保证）。

整除分块

记 $a = \left[\frac{n}{i}\right]$。我们考虑一下，如何找到最大的 k，使得 $\left[\frac{n}{k}\right] = a$。

$$\begin{aligned} \left[\frac{n}{k}\right] = a &\implies \frac{n}{k} \geq a \\ &\implies n \geq ak \implies k \leq \frac{n}{a} \\ &\implies k \leq \left[\frac{n}{a}\right] \end{aligned}$$

不难验证 $k = \left[\frac{n}{a}\right]$ 满足 $\left[\frac{n}{k}\right] = a$

整除分块

```
for (int l = 1, r = 1; l <= n; l = r + 1){
    r = n / (n / l);
}
```

上述代码，将 $[1, n]$ 区间划分成 $2\sqrt{n}$ 个连续区间 $[l_i, r_i]$，满足对于每个区间 $x \in [l_i, r_i]$，$\left\lfloor \frac{n}{x} \right\rfloor$ 的值相等。

[CQOI2007] 余数求和

题意：给出正整数 n 和 k，请计算

$$G(n,k)=\sum_{i=1}^{n} k \bmod i$$

其中 $k \bmod i$ 表示 k 除以 i 的余数。$n,k \le 10^9$。

[CQOI2007] 余数求和

我们将取模改写成整除：$a \bmod p = a - \left\lfloor \frac{a}{p} \right\rfloor p$。

原式可改写成：

$$
\begin{aligned}
G(n,k) &= \sum_{i=1}^{n} k - \left\lfloor \frac{k}{i} \right\rfloor i \\
&= nk - \sum_{i=1}^{n} \left\lfloor \frac{k}{i} \right\rfloor i
\end{aligned}
$$

我们使用之前提到的方法，将 $[1,n]$ 分成 $2\sqrt{n}$ 个区间，每个区间内 $\left\lfloor \frac{n}{i} \right\rfloor$ 相等，只需要计算 $\sum_{i=l}^{r} i$ 即可。

具体实现时，需要注意一下 $n > k$ 和 $n < k$ 分别是什么情况。

P2424 约数和

记 $f(x)$ 表示 x 所有约数的和。例如 $f(6)=1+2+3+6=12$。给定两个数 x,y，求出 $\sum_{i=x}^{y} f(i)$。

$1 \le x \le y \le 2 \times 10^9$

P2260 [清华集训 2012] 模积和

给定 n, m，求 $\sum_{i=1}^{n} \sum_{j=1}^{m} (n \bmod i) \times (m \bmod j), i \neq j$，对 19940417 取模。

$1 \leq n, m \leq 10^9$

Thanks!