数论

kczno1

2020.2.10.

整除理论

对于两个整数 a, b，且 $b > 0$，则存在唯一的整数 q 和 r，使得 $a = qb + r, r \in [0, b)$。
记 $q = \lceil \frac{a}{b} \rceil, r = a \bmod b$。
如果 $a \bmod b = 0$，则称 a 被 b 整除，或 b 整除 a，记作 $b|a$。

整除理论

引理：
若正整数 a, b, c, q 满足 $a = bq + c$ ，则 $\gcd(a, b) = \gcd(b, c)$
欧几里得算法(Euclidean algorithm))：
$\gcd(a, b) = \gcd(b, a \bmod b)$ ，递归计算。
若 $a \geq b$ ，则 $a \bmod b \leq \min(b - 1, a - b) \leq \frac{a}{2}$
故时间复杂度为 $O(\log a + \log b)$

整除理论

斯坦因算法(Binary GCD algorithm)：

$$\gcd(a,b) = \begin{cases} a & \text{if a = b} \\ 2\gcd\left(\frac{a}{2},\frac{b}{2}\right) & \text{if a,b are even} \\ \gcd\left(\frac{a}{2},b\right) & \text{if a is even,b is odd} \\ \gcd\left(a,\frac{b}{2}\right) & \text{if a is odd,b is even} \\ \gcd(a-b,b) & \text{if a,b are odd} \end{cases}$$

时间复杂度 $O(\log a + \log b)$

整除理论

引理($min-max$ 容斥):

$$对于集合S, \max(S)=\sum_{\emptyset\neq T\subseteq S}(-1)^{|T|+1}\min(T),$$

$$\min(S)=\sum_{\emptyset\neq T\subseteq S}(-1)^{|T|+1}\max(T).$$

对于正整数 a,b ，有 $\gcd(a,b)\operatorname{lcm}(a,b)=ab$ 。

对于集合 S ，$\operatorname{lcm}(S)=\prod_{\emptyset\neq T\subseteq S}\gcd(T)^{(-1)^{|T|+1}}$ ，

$\gcd(S)=\prod_{\emptyset\neq T\subseteq S}\operatorname{lcm}(T)^{(-1)^{|T|+1}}$ 。

XVI Open Cup, Grand Prix of SPb, D At Least Half

给定一个长度为 n 的正整数序列 $r_{1\ldots n}$，找到一个最长的区间满足能从区间中选出一半的数字，使得它们的 gcd > 1 。
$1 \leq n \leq 5 \times 10^5, 1 \leq r_i \leq 10^6$

CF 1034A Enlarge GCD

给定 n 个正整数 $x_{1\ldots n}$，从中删除尽量少的数字，使得剩下的数的 gcd 变大，或者确定无解。

$1 \leq n \leq 3 \times 10^5, 1 \leq x_i \leq 1.5 \times 10^7$

题目

Easy:
CF 1114C Trailing Loves
SDOI 2009 SuperGCD
Medium:
LOJ 530 最小倍数
CF 438D The Child and Sequence
BZOJ 4921 互质序列
CCPC 长春 2016 F Harmonic Value Description
Hard:
Hackerrank Fibonacci LCM

同余理论

对于正整数 m，如果 $a \bmod m = b \bmod m$，则称 a 与 b 关于 m 同余，记作 $a \equiv b \pmod{m}$。
将所有模 m 同余的元素归为一个集合，那么全体整数被分为 m 个集合，这些集合被称为模 m 的剩余类。
在模 m 的每个剩余类里各取一个数，得到一个大小为 m 的集合，称为模 m 的一个完全剩余系。

同余理论

对于正整数 x, y ，对于任意整数 s, t ，有 $\gcd(x, y) | sx + ty$ 。
所以如果要解方程 $sx + ty = a$ ，如果 $\gcd(x, y) \nmid a$ 则无解，否则只需解出 $sx + ty = \gcd(x, y)$ 再乘上 $\frac{a}{\gcd(x,y)}$ 。
扩展欧几里得算法 (Extended Euclidean algorithm)：
求出 $sx + ty = \gcd(x, y)$ 的一组特解 (s_0, t_0) ，则通解为 $\{s' = s_0 + k\frac{y}{\gcd(x,y)}, t' = t_0 + k\frac{x}{\gcd(x,y)} | k \in \mathbb{Z}\}$ 。
令 $x = ky + z$, $sx + ty = \gcd(x, y) \to s(ky + z) + ty = \gcd(x, y) \to (sk + t)y + sz = \gcd(x, y) = \gcd(y, z)$ ，所以只需对 y, z 递归操作。
时间复杂度同欧几里得算法。

同余理论

乘法逆元：对于正整数 m ，整数 x ，若存在整数 a 满足 $ax \equiv 1 \pmod{m}$ ，则称 a 是 x 关于 m 的乘法逆元。
这等价于存在整数 b ， $ax + mb = 1$ 。
故 a 存在当且仅当 $\gcd(x, m) = 1$ ，且可用扩展欧几里得算法求出。

同余理论

考虑解同余方程组

$$\begin{cases} x \equiv r_1 \pmod{m_1} \\ x \equiv r_2 \pmod{m_2} \\ \dots \\ x \equiv r_n \pmod{m_n} \end{cases}$$

中国剩余定理(Chinese remainder theorem):

当 $m_1, m_2 \dots m_n$ 两两互质时，解为 $x \equiv \sum_{i=1}^{n} r_i M_i' M_i \pmod{M}$ ，

其中 $M = \prod_{i=1}^{n} m_i, M_i = \frac{M}{m_i}, M_i' \equiv M_i^{-1} \pmod{m_i}$ 。

同余理论

对于一般的情况，考虑每次合并两个方程，重复 $n-1$ 次。
对于方程 $x \equiv r_1 \pmod{m_1}$ 和 $x \equiv r_2 \pmod{m_2}$,存在整数 k_1, k_2 ，使得 $x = k_1 m_1 + r_1 = k_2 m_2 + r_2$ ，即 $k_1 m_1 - k_2 m_2 = r_2 - r_1$,使用扩展欧几里得算法即可。

CF 819D Mister B and Astronomers

有 n 个观察员，第一个观察员在 0 秒会观察星空一次，第 i 个会在第 $i-1$ 个观察员之后 a_i 秒观察，第一个观察员会在第 n 个观察员之后 a_1 观察。
有一颗星星随机在 $[-T,-1]$ 中的一个整数秒开始闪烁，每隔 T 秒闪烁一次。
问每个观察员有多大概率称为第一个观察到这颗星星的人，答案乘以 T 输出。
$1 \leq T \leq 10^9, 2 \leq n \leq 2 \times 10^5, 1 \leq a_i \leq 10^9$

题目

Easy:
POJ 青蛙的约会
The Balance
NOI2018 屠龙勇士
Medium:
CF 492E Vanya and Field
Hard:
projecteuler Chinese leftovers II

同余理论

在模 m 意义下与 m 互质的剩余类中各取一数组成的集合，叫做模 m 的一个简化剩余系，也叫缩系。
将模 m 缩系的大小记作欧拉函数 $\varphi(m)$ 。
若 $\gcd(k, m) = 1$ ，且 $\{a_0, a_1, \ldots a_{\varphi(m)-1}\}$ 为模 m 的一个缩系，则 $\{ka_0, ka_1, \ldots ka_{\varphi(m)-1}\}$ 也是模 m 的一个缩系。

同余理论

费马小定理(Fermat’s little theorem)：
对于质数 p 和整数 a，有 $a^p \equiv a \pmod{p}$。
欧拉定理(Euler’s totient theorem)：
对于正整数 m, a，若 $\gcd(m, a) = 1$，则 $a^{\varphi(m)} \equiv 1 \pmod{m}$。
威尔逊定理(Wilson’s theorem)：
对于大于 1 的正整数 n，n 是质数当且仅当 $(n-1)! \equiv -1 \pmod{n}$。
扩展欧拉定理：
对于正整数 m, a, b，$a^b \equiv a^{b \bmod \varphi(m)+\varphi(m)} \pmod{m}(b \geq \varphi(m))$。

题目

Medium:
BZOJ 上帝与集合的正确用法
Ynoi2016 炸脖龙

同余理论

离散对数问题：
给定 A, B, C ，解同余方程 $A^x \equiv B \pmod{C}$ 。
首先考虑 $gcd(A, C) = 1$ ，即 A 在$\bmod C$ 意义下存在逆元的情况。
注意到， $A^x \bmod C$ 随着 x 的变化具有周期性，最大周期不超过 C 。
故 $A^x \equiv B \pmod{C}$ 有解的充要条件为在 $x \in [0, C)$ 有解。
所以我们只要解出 $x \in [0, C)$ 的一个解即可。(再结合最小周期即可得到通解，但一般不需要)

同余理论

大步小步算法 (Baby-step giant-step,BSGS)：
设定一个块的大小 n，考虑把每 n 个 x 分成一块，这时第 i 块内要解的方程相当于：$A^{in-j} \equiv B \pmod{C}$，这里 $1 \le i \le \lceil \frac{C}{n} \rceil, 1 \le j \le n$。
两边同乘 A^j 得 $A^{in} \equiv B \times A^j \pmod{C}$。
枚举 j，把所有 $B \times A^j \bmod C$ 存入哈希表；然后枚举 i，在哈希表中查询 $A^{in} \bmod C$。
时间复杂度 $O(n + \frac{C}{n})$，当 n 取 $O(\sqrt{C})$ 时，时间复杂度取到最优值 $O(\sqrt{C})$。
注意到，当 $gcd(A, C) \neq 1$,即 A 在mod$C$ 意义下不存在逆元时，得到 $A^{in} \equiv B \times A^j \pmod{C}$ 后并不能在两边同乘 A^{-j}，因此这个做法是不可行的。

同余理论

扩展大步小步算法 (Extended Baby-step giant-step)：
考虑通过预处理把问题转化成 A, C 互质的情况。
将 $A^x \equiv B \pmod{C}$ 看做是 $A^x + Cy = B$ 方便叙述与处理。
考虑将方程一直除去 A, C 的 gcd 进行变形，最终使得 A, C 互质。

将方程左右同除 $d_1 = \gcd(A, C)$ ，得到 $B_1 = \frac{A}{d_1} A^{x-1} + C_1 y$ (其中 $B_1 = \frac{B}{d_1}, C_1 = \frac{C}{d_1}$)；如果 A 和 C_1 不互质，继续将方程左右同除 $d_2 = \gcd(A, C_1)$ 得到 $B_2 = \frac{A^2}{d_1 d_2} A^{x-2} + C_2 y$ 。重复操作直到 A 和 C_i 互质。

同余理论

需要注意，能将 $B_i = \frac{A^i}{d_1 d_2 .. d_i} A^{x-i} + C_i y$ 两边同除 $\gcd(A, C_i)$ 的前提是答案 $> i$ 。

一个方便的解决方法就是预先特判答案较小(比如 < 100)的情况。

在特判小答案之后，如果遇到 $\gcd(A, C_i) \nmid B_i$ 的情况就可以直接认为无解了。

最终得到 $B_n = \frac{A^n}{d_1 d_2 .. d_n} A^{x-n} + C_n y$ 。记 $D = \frac{A^n}{d_1 d_2 .. d_n}$ ，因为 $\gcd(A, C_n) = 1$ ，所以 $\gcd(D, C_n) = 1$ ，所以 D 在 $\bmod C_n$ 意义下存在逆元，所以原式等价于 $D \times A^{x-n} \equiv B_n \pmod{C}$ ，等价于 $A^{x-n} \equiv B_n \times D^{-1} \pmod{C_n}$ 。此时就可以求解了。

同余理论

将满足 $x^l \equiv 1 \pmod{m}$ 的最小正整数 l 称为 x 的阶 $ord_m(x)$ 。
$ord_m(x)$ 存在当且仅当 $\gcd(x, m) = 1$ 。
若存在正整数 N ， $x^N \equiv 1 \pmod{m}$，则 $ord_m(x)|N$ 。由此可知，当 $ord_m(x)$ 存在时有 $ord_m(x)|\varphi(m)$ 。
若 $ord_m(x) = u$ ，则对于任意正整数 k ， x^k 的阶为 $\frac{u}{\gcd(u,k)}$ 。

同余理论

若存在阶为 $\varphi(m)$ 的元素 g，则称 g 是模 m 意义下的一个原根。$\{g^i | i = 0, 1 \ldots \varphi(m) - 1\}$ 构成一组缩系。

若 $m = \prod_{i=1}^{k} p_i^{e_i}$，则

$ord_m(x) = \mathrm{lcm}(ord_{p_1^{e_1}}(x), ord_{p_2^{e_2}}(x) \ldots ord_{p_k^{e_k}}(x))$。

原根存在的充要条件是 $m = 2, 4, p^n, 2p^n$，这里 p 是奇质数，n 是正整数。

原根若存在，则有 $\varphi(\varphi(m))$ 种互不同余的原根。

求质数 p 的原根：

最小的原根都比较小，故可以通过从小到大枚举来寻找原根。对于一个待检查的正整数 g ,对于 $p-1$ 的每个质因子 a，检查 $g^{\frac{p-1}{a}} \equiv 1 \pmod{p}$ 是否成立，如果成立说明 g 不是原根；如果都不成立，说明 g 是原根。

同余理论

高次剩余问题：给定 B, C, M ，解同余方程 $A^B \equiv C \pmod{M}$ 。

由于该问题比较复杂，我们只考虑 M 是质数的情况。
令 g 为 M 的原根，通过 g 可以将 $1, 2, \ldots M-1$ 和 $g^1, g^2, \ldots g^{M-1}$ 建立一一映射。
特判 $C = 0$ 的情况。设 $g^s = A, g^t = C$ (t 可以用离散对数解出)，则 $g^{sB} \equiv g^t \pmod{M}$ ，即 $sB \equiv t \pmod{M-1}$ ，解同余方程即可。

题目

Easy:
BSGS 模板题
原根 模板题
Medium:
exBSGS 模板题
质数高次剩余 模板题
SDOI2015 序列统计
Very Hard:
高次剩余 模板题

素性检测

问题：快速判断一个数 n 是不是质数。
根据费马小定理，若 n 是质数，则有 $a^{n-1} \equiv 1 \pmod{n}$，其中 $0 < a < n$。
因此一个想法就是多次在 $[1, n-1]$ 随机一个 a，并检验是否有 $a^{n-1} \equiv 1 \pmod{n}$，一旦不成立，就说明 n 是合数。
这个算法是正确的，因为若 $n = x \times y (x > 1, y > 1)$，则 $x | x^{n-1} \bmod n$，因此 $x^{n-1} \bmod n \neq 1$。然而对于某些合数，使得同余式不成立的 a 很少，于是这个算法随机次数需要很大，因此这个算法的复杂度是很高的。

素性检测

引理：若 p 是质数，则 $x^2 \equiv 1 \pmod{p}$ 的解为 $x \equiv \pm 1 \pmod{p}$ 。

证明：

$x^2 \equiv 1 \pmod{p}$

移项得：$(x-1)(x+1) \equiv 0 \pmod{p}$

即 $p|(x-1)(x+1)$

因此 $p|x-1$ 或 $p|x+1$

即 $x \equiv \pm 1 \pmod{p}$ 。

证毕。

素性检测

米勒拉宾测试 (Miller–Rabin primality test)：
考虑利用这个引理优化上面那个方法。
1. 多次在 $[1, n-1]$ 随机一个 a 。
2. 令 $x = n-1$ 。
3. 判断 $a^x \equiv 1 \pmod{n}$ 是否成立，若不成立就说明 n 是合数(根据费马小定理)。
4. 如果 x 是奇数，那么结束这次的检验；否则进行步骤 5 。
5. 判断 $a^{x/2} \equiv \pm 1 \pmod{n}$ 是否成立，若不成立就说明 n 是合数(根据引理)；
否则若 $a^{x/2} \equiv -1 \pmod{n}$ ，则结束这次的检验，否则令 $x/=2$ ，然后重复步骤 4 。

素性检测

如果分别选取 $2,3,5,7$ 作为 a，那么就能保证 10^9 以内所有数的正确判断。
如果分别选取最小的 12 个质数作为 a，那么就能保证 10^{18} 以内所有数的正确判断。
具体实现时，先求出 $n-1=n_1\times 2^l$，其中 n_1 是奇数，然后求出 $x=a^{n_1} \bmod n$，若 $x=1$ 或 $x=n-1$ 结束检验；否则重复 $l-1$ 次，每次让 $x=x^2 \bmod n$，若 $x=n-1$ 结束检验，若 $l-1$ 次后还没出现 $x=n-1$ 则说明 n 为合数。

大数分解

问题：给定一个 $\leq 10^{18}$ 的数，输出它分解质因数后的结果。
Pollard’s rho algorithm：
首先如果 n 是质数或者 $n = 1$，直接返回。(用 Miller-Rabin 算法判断)
否则我们找出 n 的一个非平凡因子 $i(1 < i < n)$，然后递归对 i 和 $\frac{n}{i}$ 操作。
如何找出一个 n 的非平凡因子呢？
考虑对于 n 的一个质因子 p，如果我们得到两个数 x, y，满足 $x \equiv y \pmod{p}$ 但 $x \not\equiv y \pmod{n}$，那么 $\gcd(|x - y|, n)$ 就一定是 n 的一个非平凡因子。

大数分解

这时候就有一个很巧妙的想法。
考虑这样生成一个模 n 意义下的随机数列 $\{x_m\}$：$x_1 = a$，$x_i = f(x_{i-1}) \bmod n\ (i > 1)$，其中 a 是一个 $\in [0, n)$ 的随机数。其中 $f(x)$ 为一个随机函数，一般取 $f(x) = x^2 + c$，其中 c 是一个常数。
由于 x_i 由 x_{i-1} 唯一确定，因此 $\{x_m\}$ 一定会形成一个 ρ 形，也就是说从某个 x_i 开始会进入一个循环，即对于所有 $j \geq i$ 有 $x_{j+t} = x_j$，其中 t 是这个循环节的长度。由生日悖论可知，$i + t$ (即 ρ 的长度)期望下是 $O(\sqrt{n})$ 的。

大数分解

对于 n 的一个质因子 p，考虑模 p 意义下的随机数列 $\{x'_m\}$，即满足 $x'_1 = a \bmod p$，$x'_i = f(x'_{i-1}) \bmod p \ (i > 1)$。类似的，假如 $\{x'_m\}$ 从 x'_i 开始进入循环节为 t 的循环，那么 $i + t$ 期望下是 $O(\sqrt{p})$ 的。由于 $x'_i = x_i \bmod p$，因此 x_i 和 x_{i+t} 模 p 相等，并且很有可能模 n 不相等，因此 $\gcd(|x_i - x_{i+t}|, n)$ 很可能是 n 的一个非平凡因子。

大数分解

那么我们考虑从 $i = 1$ 开始，每次检查 $v = gcd(|x_i - x_{2i}|, n)$ ，
如果 $v = 1$ ，那么接着枚举 i ；
如果 $v = n$ ，那么说明对于所有 p 第一个满足 $x_i = x_{2i}$ 的 i 都相等，这种情况很少出现，因此我们可以直接停止枚举，判定这次寻找失败，换一个初始值和随机函数继续寻找；
否则 v 就是 n 的一个非平凡因子，返回 v 即可。

大数分解

考虑对于一个 p 第一个满足 $x_i = x_{2i}$ 的 i 是多少，相当于 x, y 两个人从同一个起点出发走同一个 ρ 形，每次 x 走一步，y 走两步，那么当 x 第一次走到环上的时候，接下来就变成一个追及问题，再走当前距离次，y 就会追上 x 了。因此 i 不超过 ρ 的长度的两倍，期望下是 $O(\sqrt{p})$ 的。

由于合数 n 的最小质因子不超过 $O(\sqrt{n})$ ，因此分解一次的期望时间复杂度为 $O(n^{1/4}\log n)$ ，其中 $\log n$ 是因为要求 gcd 。由于 n 每次会被分成两个数，可以证明总复杂度同样为 $O(n^{1/4}\log n)$ 。

大数分解

同时，Pollard's rho 有一个不常用的优化，可以将期望时间复杂度变成 $O(n^{1/4})$ 。
设定一个参数 β ，每 β 个 i 一起搞，计算这 β 个 $|x_i - x_{2i}|$ 的乘积和 n 的 gcd ：
如果 gcd 为 1 ，说明每个 $|x_i - x_{2i}|$ 和 n 的 gcd 都是 1 ，接着枚举；
否则说明存在至少一个 i ，$|x_i - x_{2i}|$ 和 n 的 gcd 不是 1 ，这时我们一个个 i 枚举过来就好了。
分解一次时间复杂度为 $O(n^{1/4} + \frac{n^{1/4}\log n}{\beta} + \beta \log n)$ ，合理设置 β (比如设为 $\log n$)即可做到 $O(n^{1/4})$ 。

大数分解

试除法：
预处理不超过 $n^{\frac{1}{K+1}}$ 的质数，筛去 n 的这部分质因子，则剩下的质因子数量不超过 K 。
利用素性检测做一些分类讨论，避免大数分解。

IX Open Cup, Kharkiv Grand Prix K Minimal Power of Prime

T 组询问，每组询问给定一个整数 n ，问 n 的质因数分解里最小的幂指数是多少。
$T \leq 10^5, 2 \leq n \leq 10^{18}$

题目

Easy:
Miller-Rabin 模板题
Medium:
Pollard's rho 模板题
CQOI 2016 密钥破解
HDU 5447 Good Numbers
Hard:
POI 2010 Divine divisor

卢卡斯定理

Lucas' s theorem：
对于非负整数 n, m 和质数 p，令 n, m 的 p 进制表示分别为 $\sum_{k\geq 0} n_k p^k, \sum_{k\geq 0} m_k p^k (0 \leq n_k, m_k < p)$，则有 $\binom{n}{m} \equiv \prod_{k\geq 0} \binom{n_k}{m_k} \pmod{p}$。

一些扩展

卢卡斯定理

基于母函数的证明 [编辑]

本证明由Nathan Fine[2]给出。

对于素数p和n，满足1≤n≤p-1，二项式系数

$$\binom{p}{n} = \frac{p\cdot(p-1)\cdots(p-n+1)}{n\cdot(n-1)\cdots 1}$$

可被p整除。由此可得，在母函数中

$$(1+X)^p \equiv 1+X^p \pmod{p}.$$

应用数学归纳法可证，对于任意非负整数i，有

$$(1+X)^{p^i} \equiv 1+X^{p^i} \pmod{p}.$$

对于任意非负整数m和素数p，将m用p进制表示，即$m=\sum_{i=0}^{k} m_i p^i$，其中k为非负整数、m_i为整数且 0 ≤ m_i ≤ p-1。注意到

$$\begin{aligned}\sum_{n=0}^{m}\binom{m}{n}X^n &= (1+X)^m = \prod_{i=0}^{k}\left((1+X)^{p^i}\right)^{m_i}\\ &\equiv \prod_{i=0}^{k}\left(1+X^{p^i}\right)^{m_i} = \prod_{i=0}^{k}\left(\sum_{n_i=0}^{m_i}\binom{m_i}{n_i}X^{n_i p^i}\right)\\ &= \prod_{i=0}^{k}\left(\sum_{n_i=0}^{p-1}\binom{m_i}{n_i}X^{n_i p^i}\right) = \sum_{n=0}^{m}\left(\prod_{i=0}^{k}\binom{m_i}{n_i}\right)X^n \pmod{p},\end{aligned}$$

其中 n_i是n的p进制表达的第i位。此即证明了本定理。

题目

Medium:
ZOJ 2116 Christopher’ s Christmas Letter
Hard:
2017 计蒜之道复赛商汤智能机器人
BZOJ 2629 binomial
Hackerrank Binomial Coefficients Revenge

类欧几里得

问题：给定正整数 a, b, c, n ，求 $f(a, b, c, n) = \sum_{i=0}^{n} \lfloor \frac{ai + b}{c} \rfloor$ 。要求时间复杂度为 $O(\log \max(a, b, c, n))$ 。

首先做可以简单的转化使得 a, b 分别对 c 取模。

然后令 $m = \lfloor \frac{an+b}{c} \rfloor$

$$f(a, b, c, n) = \sum_{i=0}^{n} \sum_{j=0}^{m} [j < \lfloor \frac{ai + b}{c} \rfloor] = \sum_{j=0}^{m} \sum_{i=0}^{n} [j < \lfloor \frac{ai + b}{c} \rfloor]$$

由于 $j < \lfloor \frac{ai+b}{c} \rfloor \Leftrightarrow \lfloor \frac{jc+c-b-1}{a} \rfloor < i$

所以 $f(a, b, c, n) = \sum_{j=0}^{m} \sum_{i=0}^{n} [\lfloor \frac{jc+c-b-1}{a} \rfloor < i] =$

$\sum_{j=0}^{m} (n - \lfloor \frac{jc+c-b-1}{a} \rfloor) = nm - f(c, c - b - 1, a, m - 1)$

注意到 a, c 交换了位置，所以又可以取模，再递归。这就是个辗转相除的过程。故时间复杂度为 $O(\log a + \log c)$ 。

题目

Medium:
CCPC 杭州 2016 Mod, Xor and Everything
POJ A Modular Arithmetic Challenge
Hard:
洛谷 类欧几里得算法
LOJ 类欧几里得算法
Vijos 强大的区间
2016 Petrozavodsk Winter, Makoto Soejima Contest 4 K Stains
Timus 1797 Summit Online Judge 2
Very Hard:
LOJ 万能欧几里得
2014 Petrozavodsk Summer, Petr Mitrichev Contest 12 F
Recognize Power of Two

Min_25筛

问题:求积性函数 f(x) 的前缀和。
以 简单的函数 为例。
对于例题，显然 $f(x)$ 是积性函数，然而想找到其他的性质却不容易。
这时直接上 Min_25 筛就好了。
定义 $g(n,m)=\sum_{2\le x\le n}[x\text{不含}\le m\text{的质因子}]f(x)$ 。
那么答案就是 $g(n,0)+f(1)$ 。

Min_25筛

$g(n,m)$ 可以暴力枚举下一个最小质因子 p 及其幂次 e 转移:

$$g(n,m)=\sum_{m<p\le n,p\ is\ prime}\ \sum_{e\ge 1,p^e\le n} f(p^e)(1+g(\lfloor\frac{n}{p^e}\rfloor,p))$$

注意到，当 $p>\sqrt{n}$ 时，贡献就是 $f(p)$ ，因此考虑特判所有质数的贡献来减少 p 的枚举。

$$g(n,m)=\sum_{m<p\le\sqrt{n},p\ is\ prime}\ \sum_{e\ge 1,p^e\le n} f(p^e)([e>1]+g(\lfloor\frac{n}{p^e}\rfloor,p))$$
$$+\sum_{m<p\le n,p\ is\ prime} f(p)$$

Min_25筛

定义 $h(n)=\sum_{2\le p\le n,p\ is\ prime} f(p)$，则

$$g(n,m)=\sum_{m<p\le\sqrt{n},p\ is\ prime}\sum_{e\ge 1,p^e\le n} f(p^e)([e>1]+g(\lfloor\frac{n}{p^e}\rfloor,p))$$
$$+h(n)-h(m)$$

如果预处理出了 h，那么 g 只用这样暴力 dfs 递归即可，当 $n\le 10^{13}$ 时，时间复杂度接近 $O(\frac{n^{\frac{3}{4}}}{\log n})$ (实际时间复杂度为 $O(n^{1-\epsilon})$，见朱震霆论文)。

Min_25筛

下面考虑如何求 h 。
可以发现对于用到的 $h(i)$ 一定存在 m ，使得 $\lfloor \frac{n}{m} \rfloor = i$ 。
而且在例题及大多题目中，$f(p)$ (p是质数) 是一个关于 p 的低次多项式，即 $f(p) = \sum_t a_t p^t$ 且 t 的最大值很小，而每个 t 显然是可以分开计算贡献然后相加的，因此下面只考虑 $f(p) = p^t$ 的情况。
也就是说，我们的问题为，对所有的 $i = \lfloor \frac{n}{m} \rfloor$ ，求 $h(i) = \sum_{p \le i, p \ is \ prime} p^t$ 。

Min_25筛

考虑埃式筛法的过程，即从小到达枚举每个 p ，筛去所有 $\geq p^2$ 的 p 的倍数。

定义 $h'_{i,j}$ 表示埃式筛法枚举了前 i 个质数后，$\leq j$ 的还剩下的数的 t 次方之和，p_i 表示第 i 个质因子。

即 $h'_{i,j}=\sum_{1\leq x\leq j}[x是1,质数,或没有\leq p_i的质因子的数]x^t$ 。

转移：

对于 $j\geq p_i^2$ ，$h'_{i,j}=h'_{i-1,j}-p_i^t(h'_{i-1,\lfloor\frac{j}{p_i}\rfloor}-h'_{i-1,p_i-1})$ ，因为所有被 p_i 第一次筛去的数一定是 $p_i\times x$ ，其中 x 没有 $<p_i$ 的质因子；

对于 $j<p_i^2$ ，$h'_{i,j}=h'_{i-1,j}$ 。

使用滚动数组，时间复杂度 $O(\frac{n^{\frac{3}{4}}}{\log n})$ (可以用积分证明)。

Min_25筛

有时候我们需要对每个 $\lfloor \frac{n}{i} \rfloor$ 求前缀和，这时如果跑 $\sqrt{n}$ 次会很慢，如果记忆化，那么时间复杂度不变，空间复杂度为 $O(\frac{n^{\frac{3}{4}}}{\log n})$ 。

实际上 $g(n,m)$ 也可以非递归的求出，且时间复杂度为 $O(\frac{n^{\frac{3}{4}}}{\log n})$ 。

令

$$g'(n,m)=\sum_{m<p\le\sqrt{n},p\ is\ prime}\ \sum_{e\ge 1,p^e\le n} f(p^e)([e>1]+g(\lfloor\frac{n}{p^e}\rfloor,p))$$ 。

考虑从大到小枚举 i ，对 $d(n,i)=g'(n,p_i)-g'(n,p_{i+1})=\sum_{e\ge 1,p_{i+1}^e\le n} f(p_{i+1}^e)([e>1]+g(\lfloor\frac{n}{p_{i+1}^e}\rfloor,p_{i+1}))$ 暴力计算即可。

题目

见 tangjz's blog。