数论函数求和问题（上）

2022 年 1 月 20 日　成都

1. 数论函数
2. 莫比乌斯反演
3. 整除
4. 常见积性函数
5. 高维变换

基本概念

数论函数

（简单起见）值域为正整数的函数。

完全积性函数

若数论函数 f 满足 $f(nm)=f(n)f(m)$，则称为完全积性函数。

积性函数

若数论函数 f 满足 $n\perp m \Rightarrow f(nm)=f(n)f(m)$，则称为积性函数。

积性分解

对于积性函数 f，给出质因数分解 $n=p_1^{c_1}p_2^{c_2}\dots p_m^{c_m}$。则有

$$f(n)=\prod_{i=1}^{m} f(p_i^{c_i})$$

推论：只需质数幂处的取值，就足可以确定一个积性函数。

基本概念

狄利克雷卷积

两个数论函数 f,g 的狄利克雷卷积为一个新的数论函数，记作 $f*g$ 。其满足：

$$(f*g)(n)=\sum_{x\times y=n} f(x)g(y)=\sum_{d|n} f(d)g(n/d)$$

狄利克雷卷积的基本性质

- 交换律：$f*g=g*f$ 。
- 结合律：$(f*g)*h=f*(g*h)$ 。
- 单位元：记数论函数 ϵ 满足 $\epsilon(n)=[n=1]$ 。对于任意数论函数 f ，都有 $f*\epsilon=f$ 。

以上性质根据定义容易证明。

狄利克雷卷积的计算

乘积：给出 f,g ，计算 $f*g$。
直接用定义式计算。为了方便，转枚举因数为枚举倍数。

求逆：给出 f 满足 $f(1)=1$ ，求出 g 满足 $f*g=\epsilon$ 。
显然有 $g(1)=1$ 。假设对于 $n>1$ 已求出 $g(1)\sim g(n-1)$ ，需要求 $g(n)$ 。

$$(f*g)(n)=\epsilon(n)$$

$$\sum_{d|n} f(d)g(n/d)=0$$

$$g(n)f(1)+\sum_{d|n,d\neq 1} f(d)g(n/d)=0$$

$$g(n)=-\sum_{d|n,d\neq 1} f(d)g(n/d)$$

若计算前 n 项，两者复杂度均为 $O\left(\sum_{i=1}^{n} n/i\right)=O(n\log n)$ 。

狄利克雷卷积的积性性质

定理：两个积性函数的卷积仍是积性函数。

设有两个积性函数 f_1, f_2，令 $g = f_1 * f_2$。

设 a, b 满足 $a \perp b$。

$$
\begin{aligned}
g(a)g(b) &= \sum_{d|a} f_1(d)f_2(a/d) * \sum_{t|b} f_1(t)f_2(b/t) \\
&= \sum_{dt|ab} f_1(d)f_2(a/d)f_1(t)f_2(b/t) \\
&= \sum_{dt|ab} f_1(dt)f_2(ab/dt) \\
&= g(ab)
\end{aligned}
$$

第二行：$a \perp b \Rightarrow (d|a, t|b \Leftrightarrow dt|ab)$

第三行：根据 f_1, f_2 的积性进行合并。

狄利克雷卷积的积性性质

定理：积性函数的逆仍是积性函数。

设有积性函数 f （注意到必有 $f(1)=1$），它的逆是 g 。

设 a,b 满足 $a\perp b$ ，欲证 $g(ab)=g(a)$。

使用归纳法。当 a 或 b 为 1 时，显然成立。假设我们对 $a'<a,b'<b$ 的 (a',b') 都已经完成了证明。

$$
\begin{aligned}
g(ab) &= -\sum_{d|ab,d\neq 1} f(d)g(ab/d)\\
&= -\sum_{i|a,j|b,ij\neq 1} f(ij)g(ab/ij)\\
&= -\sum_{i|a,j|b,ij\neq 1} f(i)f(j)g(a/i)g(b/j)\\
&= f(1)f(1)g(a)g(b)-\sum_{i|a} f(i)g(a/i)\sum_{j|b} f(j)g(b/j)\\
&= f(1)f(1)g(a)g(b)-\epsilon(a)(b)=g(a)g(b)
\end{aligned}
$$

莫比乌斯函数 μ

定义:

$$\mu = I^{-1}$$

根据“积性函数的逆是积性函数”，μ 是积性函数。

尝试写出 μ 在质数幂处的取值，这是研究积性函数的通用方法。
根据 $\mu * I = \epsilon$ ，可得:

- $\mu(1) = 1$
- $\mu(p)I(1) + \mu(1)I(p) = 0 \quad \Rightarrow \quad \mu(p) = -1$
- 对于 $k > 1$ ，$\sum_{i=0}^{k} \mu(p^i) = 0$ ，归纳可证得 $\mu(p^k) = 0$ 。

$$\mu(p^k) = \begin{cases} 1 & k = 0 \\ -1 & k = 1 \\ 0 & k \geq 2 \end{cases}$$

莫比乌斯函数 μ

可以得到总的定义式：

$$\mu(n) = \begin{cases} (-1)^{n\text{的素因子数目}} & n\text{不含平方因子} \\ 0 & \text{otherwise} \end{cases}$$

莫比乌斯反演

唯一分解与高维点表示

记 p_k 为从小到大第 k 个质数。根据唯一分解定理，正整数 n 可以唯一地分解为素数幂的乘积 $\prod_{i=1}^{\infty} p_i^{c_i}$ 。可以用数列 $(c_1, c_2, c_3 \ldots)$ 表示 n ，记 $\vec{n} = (c_1, c_2, c_3 \ldots)$ 。

当考虑的值域有限时，数列长度（即素数个数）也就是有限的。可以将数列视作高维点，根据高位点的偏序，定义 $\le, <$ 等。

有下列性质：

- $n|m$ 等价于 $\vec{n} \le \vec{m}$ 。
- $\overrightarrow{\gcd(n,m)} = \left(\min(\vec{n}_1, \vec{m}_1), \min(\vec{n}_2, \vec{m}_2), \ldots\right)$ （逐位取 min）
- $\overrightarrow{\operatorname{lcm}(n,m)} = \left(\max(\vec{n}_1, \vec{m}_1), \max(\vec{n}_2, \vec{m}_2), \ldots\right)$ （逐位取 max）

莫比乌斯反演

约数求和

对函数 f 记 $\mathrm{Lsum}(f)$ 为:

$$\big(\mathrm{Lsum}(f)\big)(n)=\sum_{d|n}f(d)=\sum_{\vec{d}\le\vec{n}}f(d)$$

以高维点的视角来看，这是高维前缀和。

定义约数差分 Ldif 满足 $\mathrm{Ldif}(\mathrm{Lsum}(f))=f$ ，即“约数求和”的逆操作。

倍数求和

对函数 f 记 $\mathrm{Rsum}(f)$ 为:

$$\big(\mathrm{Rsum}(f)\big)(n)=\sum_{n|d}f(d)=\sum_{\vec{n}\le\vec{d}}f(d)$$

以高维点的视角来看，这是高维后缀和。

类似地定义倍数差分 Rdif。

莫比乌斯反演

约数求和以及倍数求和的计算容易 $O(n\log n)$ 完成。

约数差分的计算

给出 $g=\mathrm{Lsum}(f)$ ，求 f 。

条件等价于 $g=f*I$ ，则 $f=g*\mu$ ，只需将 g 卷上 μ 即可得到 f 。

倍数差分的计算

给出 $g=\mathrm{Rsum}(f)$ ，求 f 。

结论：

$$f(n)=\sum_{n|d}\mu(n/d)g(d)$$

在求得 μ 的情况下，两者都可以 $O(n\log n)$ 计算。

莫比乌斯反演

证明：

$$\begin{aligned}
&\sum_{n|d}\mu(d/n)g(d)\\
=&\sum_{n|d}\mu(d/n)\sum_{d|t}f(t)\\
=&\sum_{k=1}\mu(k)\sum_{nk|t}f(t)\\
=&\sum_{t=1}f(t)\sum_{nk|t}\mu(k)\\
=&\sum_{t=1}f(t)[n|t]\epsilon(t/n)\\
=&f(n)
\end{aligned}$$

另一种基于容斥的理解如下。观察 $n=1$ 的式子：

$$f(1)=\sum_{d=1}\mu(d)g(d)$$

先加上全集，再减去含一个质因子的，减多了，再加上含两个质因子的，再减去含三个质因子的，以此类推。

对于 $n>1$ 的情况，将下标统一除以 n 即等价。

gcd 卷积

定义：给出函数 f,g ，记两者的 “gcd 卷积” 为函数 h ，满足：

$$h(n)=\sum_{\gcd(x,y)=n} f(x)g(y)$$

算法：先求出 $Rsum(h)$ ，然后倍数差分即可得到 h 。

$$\begin{aligned}\big(Rsum(h)\big)(n) &= \sum_{n\mid \gcd(x,y)} f(x)g(y)\\ &= \sum_{n\mid x, n\mid y} f(x)g(y)\\ &= \big(\mathrm{Rsum}(f)\big)(n)\ \big(\mathrm{Rsum}(g)\big)(n)\end{aligned}$$

尤其注意 $n\mid \gcd(x,y) \Leftrightarrow n\mid x, n\mid y$ ，这条结论很常用。
只需计算 $\mathrm{Rsum}(f),\mathrm{Rsum}(g)$ ，点乘之后就是 $\mathrm{Rsum}(h)$ 。
如果你了解位运算卷积，可以联想 and 卷积，有类似的结论。

线性筛

线性筛可以在 $O(n)$ 的时间内求出 n 以内的素数，并对于合数给出其最小素因子。

算法流程如下：

- 枚举 $i = 2 \to n$
  - 若 i 没有标记，说明 i 是质数，将其加入质数集合 P
  - 从小到大枚举 $p \in P$ ，记 $t = i \times p$
    - 标记 t
    - 记 t 的最小质因子为 p
    - 若 $p|i$ ，退出循环

若 n 被二元组 (i, p) 标记，根据退出循环的条件，i 中没有比 p 小的质因子，即 p 是 $n = i \times p$ 的最小质因子。

也不难发现，每个合数都必然会被标记。(且只标记一次)

这就证明了算法的正确性。

线性筛

（其实前面用狄利克雷卷积逆就可以 $O(n\log n)$ 求 μ ）
线性筛更多地用于求积性函数的前缀值，比如说 μ 。
记 p 为 n 的最小质因子，则有

$$\mu(n)=\begin{cases}0 & p\mid(n/p)\\ -\mu(n/p) & \text{otherwise}\end{cases}$$

这样，根据最小质因子，可以递推求出 μ 的前缀值。

线性筛

对于其他某些更复杂的积性函数 f，也许不能仅用最小质因子来递推，需要额外记录最小质因子的次数（容易求）。

记 p 为 n 的最小质因子，c 为 n 中 p 的次数，则必有

$$f(n) = f(n/p^c)f(p^c)$$

预处理所有质数幂处取值之后，可以递推。

n 以内质数的幂的个数为 $O(\sum_{p\in P}\sum_{c=1}[p^c \le n]) = O(\sum_{c=1}\sum_{p\in P}[p \le n^{1/c}]) = O(\sum_{c=1} n^{1/c}/\log n^{1/c}) = O(n/\log n)$ 。

所以，若求单个 $f(p^k)$ 的复杂度不超过 $O(\log n)$ ，则总复杂度是线性的。

Luogu P2714 四元组统计

题意：给出 n 个正整数 $a_1, a_2, ...a_n$ ，统计有多少个四元组 (i, j, k, t) 满足 $i < j < k < t$，$\gcd(a_i, a_j, a_k, a_t) = 1$

多组数据，$T \le 100, n, a_i \le 10^4$ 。

题解：

记 $F(n)$ 为 a 中 n 的出现次数。

记 $F_R = \mathrm{Rsum}(F)$ ，即 $F_R(n) = n$ 的倍数的出现次数和。

记 $G_R(n) = \binom{F_R(n)}{4}$ ，即 gcd 是 n 的倍数的四元组的个数。

计算 $G = \mathrm{Rdif}(G_R)$ ，$G(1)$ 即为答案。

CF1285F Classical?

题意：给出 n 个正整数 $a_1, a_2, ...a_n$ ，求

$$\max_{1\le i<j\le n} \operatorname{lcm}(a_i, a_j)$$

$n, a_i \le 10^5$ 。

题解：

a 中相同的元素是无用的，可以先去重。

注意到 $\operatorname{lcm}(a,b) = \dfrac{ab}{\gcd(a,b)}$ ，枚举其中的 $\gcd(a,b) = d$ 。（计算 lcm 是 d 的倍数的数对的贡献）

把 a 中 d 的倍数取出，并处以 d ，问题就变为求

$$\max_{1\le i<j\le n} [a_i \perp a_j] a_i a_j$$

CF1285F Classical?

从小到大加入，每次考虑 x 与大于 x 的集合 S 之间的对子。

若发现有 $y \in S$ 且 $x \perp y$，则可以贡献 xy。

对于之后加入的某个数 $z < x$，和 S 中某个满足 $k < y$ 的 k 的可能贡献为 $zk < xy$。于是，在获得 xy 的贡献之后可以舍弃 S 中所有 $\leq y$ 的数。

实现中，可以维护一个（从小到大的）单调栈，加入数 x 时，如果发现栈中存在某个数与 x 互质，则不断弹栈。

问题转化为，维护一个集合，支持插入删除，询问是否存在与 x 互质的数。

CF1285F Classical?

记 o_i 为 S 中 i 的出现次数，直接统计与 x 互质的数的个数。

$$
\begin{aligned}
&\sum_{i=1} o_i[\gcd(x,i)=1] \\
=&\sum_{i=1} o_i \sum_{d\mid \gcd(x,i)} \mu(d) \\
=&\sum_{i=1} o_i \sum_{d\mid x, d\mid i} \mu(d) \\
=&\sum_{d\mid x} \mu(d) \sum_{d\mid i} o_i
\end{aligned}
$$

第二行把 $[\gcd(x,i)=1]$ 视作 ϵ 函数，然后用 $\mu * I$ 来“迫害”，这是莫比乌斯反演中常用的手法。

只需动态维护 o 的倍数求和，查询时枚举因数。

由于数不重复，单次复杂度 $O(k\log k)$（其中 k 是数的个数），总复杂度 $O(a\log^2 a)$。

CF1285F Classical?

优化：

对于任意 x, y ，有 $\operatorname{lcm}(x,y)=\operatorname{lcm}\left(x,\frac{y}{\gcd(x,y)}\right)$ ，且 $x\perp\frac{y}{\gcd(x,y)}$。

注意到 $\frac{y}{\gcd(x,y)}$ 必然是 y 的因数，我们把每个数的所有因数加入，然后只跑一遍互质的情况即可，复杂度为 $O(n\log n)$。

习题

- CF585E Present for Vitalik the Philatelist
- Uoj62. 【UR 5】怎样跑得更快
- CFgym102354B. Yet Another Convolution

问题引入

经典问题：给出 n 求下列式子的值

$$\sum_{i=1}^{n} \left\lfloor \frac{n}{i} \right\rfloor$$

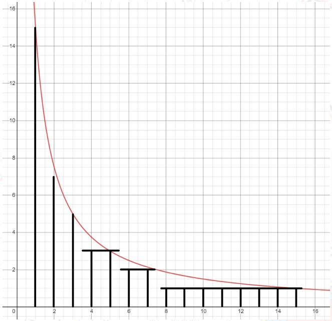
$n \leq 10^{15}$ 。

整除分块

经典结论：$\left\lfloor \frac{n}{i} \right\rfloor$ 的本质不同的取值只有 $O(\sqrt{n})$ 个。

证明：当 $i \le \lfloor \sqrt{n} \rfloor$ 时，至多有 $\lfloor \sqrt{n} \rfloor$ 个；当 $i > \lfloor \sqrt{n} \rfloor$ 时，$\left\lfloor \frac{n}{i} \right\rfloor \le \lfloor \sqrt{n} \rfloor$，也至多有 $\lfloor \sqrt{n} \rfloor$ 个。

注意到 n/i 的单调性，这些取值是一个个连续段。



整除分块

一个自然的想法是求出分段信息，然后容易求和。
考虑如何对于 i 快速计算最大的满足 $\lfloor n/i \rfloor = \lfloor n/j \rfloor$ 的 j。

- 上界：$\lfloor n/i \rfloor = \lfloor n/j \rfloor \Rightarrow \lfloor n/i \rfloor \le n/j \Leftrightarrow j \le \lfloor n/\lfloor n/i \rfloor \rfloor$
- 下界：令 $j = \lfloor n/\lfloor n/i \rfloor \rfloor$，可算得 $\lfloor n/i \rfloor = \lfloor n/j \rfloor$。

这就说明了最大的 j 即为 $\lfloor n/\lfloor n/i \rfloor \rfloor$。

典中典

题意：给出 n, m ，求

$$\sum_{i=1}^{n}\sum_{j=1}^{m}[i \perp j]$$

多组数据，$T \le 10^4, n, m \le 10^6$ 。

题解：

$$\sum_{i=1}^{n}\sum_{j=1}^{n}[i \perp j]$$

$$=\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{d|i,d|j}\mu(d)$$

$$=\sum_{d=1}\mu(d)\lfloor n/d\rfloor\lfloor m/d\rfloor$$

Luogu P2257 YY 的 GCD

题意：给出 n, m ，求

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

多组数据，$T \le 10^4, n, m \le 10^7$ 。

题解：

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{j=1}^{m}[\gcd(i,j)\in \mathrm{Prime}] &= \sum_{p\in \mathrm{Prime}}\sum_{i=1}^{\lfloor n/p\rfloor}\sum_{j=1}^{\lfloor n/p\rfloor}\sum_{d\mid i,d\mid j}\mu(d) \\
= \sum_{p\in \mathrm{Prime}}\sum_{i=1}^{n}\sum_{j=1}^{n}[\gcd(i,j)=p] &= \sum_{d=1}^{\min(n,m)}\mu(d)\sum_{p\in \mathrm{Prime}}\lfloor n/dp\rfloor\lfloor m/dp\rfloor \\
= \sum_{p\in \mathrm{Prime}}\sum_{i=1}^{\lfloor n/p\rfloor}\sum_{j=1}^{\lfloor n/p\rfloor}[i\perp j] &= \sum_{T=1}^{\min(n,m)}\lfloor n/T\rfloor\lfloor m/T\rfloor\sum_{p\mid T,p\in \mathrm{Prime}}\mu(T/p)
\end{aligned}$$

Luogu P2257 YY 的 GCD

第三行：将 i,j 分别替换成 $i/p,j/p$ 。
第六行：转而枚举 $T=dp$ 。

记 $S(T)=\sum_{p|T,p\in\mathrm{Prime}}\mu(T/p)$ ，预处理 S 后可以整除分块。
S 可以枚举素数的倍数来计算，总复杂度 $O(T\sqrt{n}+n\log\log n)$ 。

P4318 完全平方数

题意：给出 k ，求第 k 小的不是完全平方数的数。
多组数据，$T \le 50, k \le 10^9$ 。

题解：
二分，问题转化为求 n 以内的无平方因子的数的个数。
注意到 $\mu^2(n) = [n\text{无平方因子}]$ ，问题也等价于求 μ^2 的前缀和。
记 $f(n) = \sqrt{n\text{的最大平方因子}}$ 。注意到 $d|f(n) \Leftrightarrow d^2|n$ ，则有

$$\mu^2(n) = [f(n) = 1] = \sum_{d|f(n)} \mu(d) = \sum_{d^2|n} \mu(d)$$

也可以从容斥的角度理解。

P4318 完全平方数

$$\sum_{i=1}^{n}\mu^2(i)=\sum_{i=1}^{n}\sum_{d^2|i}\mu(d)$$
$$=\sum_{d=1}^{\lfloor\sqrt{n}\rfloor}\mu(d)\lfloor n/i^2\rfloor$$

$\lfloor n/i^2\rfloor$ 会形成 $O(n^{1/3})$ 段，也可以整除分块优化。
具体地，i 所在段的末尾是 $\lfloor\sqrt{\lfloor n/\lfloor n/i^2\rfloor\rfloor}\rfloor$。
总复杂度 $O(Tn^{1/3}\log n)$。

习题

- Luogu P5438 【XR-2】记忆
- Luogu P2260 [清华集训 2012] 模积和

整除差分

经典问题：CF915G Coprime Arrays

对于（正整数）序列 $a_{1\sim n}$ ，若 $\gcd(a_2, a_2, ..., a_n) = 1$ ，则称这个序列是好的。

分别对于 $k \in 1 \sim L$ ，求值域为 $1 \sim k$ 的好序列个数。

$n, L \leq 2 * 10^6$ ，不用考虑输出耗时。

题解：快进到式子

$$\text{Ans}[k] = \sum_{d=1}^{n} \mu(d) \lfloor n/d \rfloor^k$$

如果对每个 k 分别使用整除分块，复杂度为 $O(n\sqrt{n})$ ，无法通过。

整除差分

注意到 $\lfloor k/d \rfloor \neq \lfloor (k-1)/d \rfloor \Leftrightarrow d|k$ ，且此时 $\lfloor (k-1)/d \rfloor = \lfloor k/d \rfloor - 1$ 。

对答案差分，有

$$\begin{aligned}
\mathrm{Ans}[k] - \mathrm{Ans}[k-1] &= \sum_{d=1}^{k} \mu(d) \lfloor k/d \rfloor^n - \sum_{d=1}^{k-1} \mu(d) \lfloor (k-1)/d \rfloor^n \\
&= \sum_{d=1}^{k} \mu(d) \left(\lfloor k/d \rfloor^n - \lfloor (k-1)/d \rfloor^n \right) \\
&= \sum_{d|k} \mu(d) \left((k/d)^n - ((k/d)-1)^n \right)
\end{aligned}$$

线筛 id_k 之后，记 $h(n) = n^k - (n-1)^k$ ，则 $\Delta \mathrm{Ans} = h * \mu$ 。

复杂度 $O(n \log n)$ 。

习题

- Luogu P3911 最小公倍数之和
- Luogu P3704 [SDOI2017] 数字表格
- CF1139D Steps to One
- Luogu P6825 「EZEC-4」求和

常见积性函数

完全积性函数

- $\epsilon(n) = [n = 1]$
- $I(n) = 1$
- $id_k(n) = n^k$

积性函数

- $\mu = I^{-1}$
- $d(n) = n$ 的因数个数
- $\sigma_k(n) = n$ 的因数的 k 次方和
- $\varphi(n) = 1 \sim n$ 中与 n 互质的数的个数

欧拉函数 φ

$$\varphi(n)=\sum_{i=1}^{n}[i\perp n]$$

欧拉函数的卷积性质

$$\varphi(n)=\sum_{i=1}^{n}[i\perp n]=\sum_{i=1}^{n}\sum_{d|i,d|n}^{n}\mu(d)=\sum_{d|n}\mu(d)(n/d)$$

由此可得 $\varphi=\mu*id$ ，也可证得 φ 是积性函数。

欧拉函数 φ

欧拉函数的计算

根据定义不难得到

$$\varphi(p^k) = p^{k-1}(p-1)$$

记 $n = p_1^{c_1} p_2^{c_2} ... p_k^{c_k}$ ，积性分解并用上式代入，可推出

$$\varphi(n) = n\prod_{i=1}^{k} \frac{(p_i-1)}{p_i}$$

记 p 为 n 的最小质因子，则

$$\varphi(n) = \begin{cases} \varphi(n/p) * p & p|(n/p) \\ \varphi(n/p) * (p-1) & \text{otherwise} \end{cases}$$

此式可用于线性筛。

Luogu P4449 于神之怒加强版

题意：给定 n, m, k ，求

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

答案对 10^9+7 取模。多组数据，$T \le 2000, n, m, k \le 5\times 10^6$ ，其中 k 是定值。

题解：

记 $f = id_k * \mu$ ，则有 $id_k = f * I$ ，可以用 f 来迫害 $\gcd(i,j)^k$ 。

$$\text{原式} = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{d|i,d|j} f(d)$$

$$= \sum_{d=1}^{\min(n,m)} f(d)\lfloor n/d\rfloor\lfloor m/d\rfloor$$

欧拉反演？

当 $k=1$ 时，f 恰好是 φ，于是这种迫害 $\gcd(i,j)$ 的手法也被称为“欧拉反演”，其实本质还是卷 μ。

f 需用一般化的线性筛求出，具体细节不述。

复杂度 $O(T\sqrt{n}+n)$。

习题

- P2158 [SDOI2008] 仪仗队
- P2303 [SDOI2012] Longge 的问题
- P2398 GCD SUM

除数函数 d, σ_k

$$\sigma_k(n) = \sum_{d|n} d^k$$

除数函数的卷积性质

肉眼可见 $\sigma_k = I * id_k$ ，故 σ_k 是积性函数。

除数函数的计算

对于 $n = p_1^{c_1} p_2^{c_2} \dots p_k^{c_k}$ ，有：

$$d(n) = (c_1 + 1)(c_2 + 1)\dots(c_k + 1)$$

$$\sigma_k(n) = \prod_{i=1}^{k} (1 + p_i + p_i^2 + \dots + p_i^{c_i})$$

记 p 为 n 的最小质因子，有：

$$d(n) = \begin{cases} 2d(n/p) - d(n/p^2) & p|n \\ 2d(n/p) & \text{otherwise} \end{cases}$$

σ_k 则没有那么好筛。

Luogu P6060 [加油武汉] 传染病研究

题意：给出 n,k，求

$$\sum_{i=1}^{n} d(i^k)$$

多组询问，$T \le 10^4$，$n,k \le 10^7$。

题解：

对于 $n = p_1^{c_1} p_2^{c_2} \ldots p_m^{c_m}$ 有 $d(n) = (1+c_1)(1+c_2)\ldots(1+c_m)$，$d(n^k) = (1+kc_1)(1+kc_2)\ldots(1+kc_m)$。

可以发现 $d(n^k)$ 是一个关于 k 的多项式，且次数比较低，只有 $w(n) = 8$。

使用线性筛预处理多项式，并求前缀和。询问只需代入即可。复杂度为 $O((n+T)*w(n))$。

一些乘积结论

结论:

- $\mu(ij) = [i \perp j]\mu(i)\mu(j)$
- $\varphi(ij) = \dfrac{\varphi(i)\varphi(j)\gcd(i,j)}{\varphi(\gcd(i,j))}$
- $d(ij) = \sum_{x|i}\sum_{y|j}[x \perp y]$
- $\sigma_k(ij) = \sum_{x|i}\sum_{y|j}[x \perp y](\frac{xj}{y})^k$

证明:

对于 $\mu(ij)$ ，若 $i \perp j$ ，则根据积性有 $\mu(ij) = \mu(i)\mu(j)$ ；否则 ij 有平方因子，$\mu(ij) = 0$ 。

一些乘积结论

对于其余结论，先考虑 $i=p^a,j=p^b$ 的情况。由于结论式中 i,j 相联系的部分都和 gcd 有关，而 gcd 有积性，可以利用积性扩展得到原结论。

$$
\begin{aligned}
&\varphi(p^{a+b})\\
=&(p-1)p^{a+b-1}\\
=&\frac{(p-1)p^{a-1}(p-1)p^{b-1}p^{\min(a,b)}}{(p-1)p^{\min(a,b)-1}}\\
=&\frac{\varphi(p^a)\varphi(p^b)\gcd(p^a,p^b)}{\varphi(\gcd(p^a,p^b))}
\end{aligned}
$$

$$
\begin{aligned}
\sigma_k(p^{a+b})&=\sum_{i=0}^{a+b}p^i\\
&=\sum_{x=0}^{a}\sum_{y=0}^{b}[\min(x,y)=0]p^{x+b-y}\\
&=\sum_{x|p^a}\sum_{y|p^b}[x\perp y](\frac{xp^b}{y})^k
\end{aligned}
$$

Luogu P3327 [SDOI2015] 约数个数和

题意：给出 n, m ，求

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

多组询问，$T, n, m \leq 5 \times 10^4$ 。

Luogu P3327 [SDOI2015] 约数个数和

题解：

$$\sum_{i=1}^{n}\sum_{j=1}^{m} d(ij)$$
$$=\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{x|i}\sum_{y|j}[x \perp y]$$
$$=\sum_{x=1}^{n}\sum_{y=1}^{m}\lfloor n/x\rfloor\lfloor m/y\rfloor[x \perp y]$$
$$=\sum_{x=1}^{n}\sum_{y=1}^{m}\lfloor n/x\rfloor\lfloor m/y\rfloor \sum_{d|x,d|y}\mu(d)$$
$$=\sum_{d=1}^{\min(n,m)}\mu(d)\left(\sum_{x=1}^{\lfloor n/d\rfloor}\lfloor n/dx\rfloor\right)\left(\sum_{y=1}^{\lfloor m/d\rfloor}\lfloor m/dy\rfloor\right)$$
$$=\sum_{d=1}^{\min(n,m)}\mu(d)S(\lfloor n/d\rfloor)S(\lfloor m/d\rfloor)$$

记 $S(n)=\sum_{i=1}^{n}\lfloor n/i\rfloor=\sum_{i=1}^{n}d(i)$（用整除差分证明），可以线性筛预处理。

然后可以整除分块回答询问，总复杂度 $O(n+T\sqrt{n})$。

Luogu P4240 毒瘤之神的考验

题意：给出 n, m ，求

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

多组询问，$T \le 10^4, n, m \le 10^5$ 。

题解：

$$\begin{aligned}\sum_{i=1}^{n}\sum_{i=1}^{m}\varphi(ij) &= \sum_{i=1}^{n}\sum_{i=1}^{m}\frac{\varphi(i)\varphi(j)\gcd(i,j)}{\varphi(\gcd(i,j))} \\ &= \sum_{d=1}^{\min(n,m)}\frac{d}{\varphi(d)}\sum_{i=1}^{n}\sum_{i=1}^{m}\varphi(i)\varphi(j)[\gcd(i,j)=d] \\ &= \sum_{d=1}^{\min(n,m)}\frac{d}{\varphi(d)}\sum_{i=1}^{\lfloor n/d\rfloor}\sum_{i=1}^{\lfloor m/d\rfloor}\varphi(id)\varphi(jd)[i\perp j]\end{aligned}$$

Luogu P4240 毒瘤之神的考验

$$= \sum_{d=1}^{\min(n,m)} \frac{d}{\varphi(d)} \sum_{i=1}^{\lfloor n/d \rfloor} \sum_{i=1}^{\lfloor m/d \rfloor} \varphi(id)\varphi(jd) \sum_{t|i,t|j} \mu(t)$$

$$= \sum_{d=1}^{\min(n,m)} \frac{d}{\varphi(d)} \sum_{t=1}^{\lfloor \min(n,m)/d \rfloor} \mu(t) \sum_{i=1}^{\lfloor n/dt \rfloor} \sum_{i=1}^{\lfloor m/dt \rfloor} \varphi(idt)\varphi(jdt)$$

$$= \sum_{T=1}^{\min(n,m)} \sum_{d|T} \frac{d\mu(T/d)}{\varphi(d)} \left(\sum_{i=1}^{\lfloor n/T \rfloor} \varphi(idt) \right) \left(\sum_{i=1}^{\lfloor m/T \rfloor} \varphi(jdt) \right)$$

$$= \sum_{T=1}^{\min(n,m)} f(T)S(\lfloor n/T \rfloor, T)S(\lfloor m/T \rfloor, T)$$

其中 $f(n) = \sum_{d|n} \dfrac{d\mu(n/d)}{\varphi(d)}$, $S(x,y) = \sum_{i=1}^{x} \varphi(iy)$ 。

Luogu P4240 毒瘤之神的考验

注意到需要的 $S(x,y)$ 中必有 $xy \le n$ ，可以 $O(n\log n)$ 求。f 是积性函数，为了方便可以 $O(n\log n)$ 求出。

还不能直接整除分块，设 $G(a,b,t)=f(t)S(a,t)S(b,t)$ ，则

$$\sum_{T=1}^{\min(n,m)} f(T)S(\lfloor n/T\rfloor,T)S(\lfloor m/T\rfloor,T)=\sum_{T=1}^{\min(n,m)} G(\lfloor n/T\rfloor,\lfloor m/T\rfloor,T)$$

对 $G(a,b,t)$ 在 t 上做前缀和即可整除分块。

$G(a,b,t)$ 的状态量非常多，不能全部求出。考虑定一个阈值 B ，只预处理 $a,b\le B$ 的 G ，注意到 $at\le n, bt\le m$ ，状态量为 $O(Bn\log n)$ 。

回答询问时，当 $T\le \max(n,m)/B$ 时暴力求和，否则利用 G 进行整除分块。总复杂度 $O(Tn/B+T\sqrt{n}+Bn\log n)$ ，取 $B=\sqrt{T/\log n}$ 可得最优复杂度 $O(n\sqrt{T\log n})$ 。

习题

- Luogu P1891 疯狂 LCM
- Luogu P4917 天守阁的地板
- 51Nod1594 Gcd and Phi
- 51Nod1584 加权约数和
- 51Nod1594 Gcd and Phi
- Luogu P5176 公约数
- Luogu P4466 [国家集训队] 和与积
- 51Nod1222 最小公倍数计数
- Luogu P1829 [国家集训队]Crash 的数字表格 / JZPTAB
- Luogu P3312 [SDOI2014] 数表
- Luogu P3700 [CQOI2017] 小 Q 的表格

快速求和差分

我们已经知道，如果将正整数质因数分解后视作高维点，约数求和与倍数求和分别可以看做高维前缀和与高维后缀和。

注意到，传统的对集合定义的高维前缀/后缀和，暴力求解法是枚举子集，复杂度为 $O(3^n)$ ，也有 $O(n2^n)$ 的更优的做法。同理，计算约数/倍数求和时，我们暴力枚举因数/倍数可以做到 $O(n\log n)$ ，这同样不是最优的。

模仿传统高维前缀和，对每个维度（质数）逐个做前缀和，即可计算因数求和。复杂度 $O(\sum_p n/p) = O(n\log\log n)$ 。代码如下

```
for(int i=1;i<=tn;i++)
    for(int j=1;j*p[i]<=n;j++)
      F[j*p[i]]+=F[j];
```

类似地，也可 $O(n\log\log n)$ 计算约数差分，倍数求和/差分。

快速卷积性函数

给出一个数论函数 g 与积性函数 f 的前 n 项，计算 $f*g$ 的前 n 项。

考虑把 f 分解为若干个只和单个素数 p 有关的函数的卷积。记：

$$F_p(n)=\begin{cases}F(n) & (n=p^k)\\ 0 & \text{otherwise}\end{cases}$$

也就是说，只在 $1,p,p^2,...,p^k$ 处有值，其余替换为 0 。
可以发现（其中 $\prod$ 对应狄利克雷卷积）

$$F=\prod_p F_p$$

说明：$F_{p_1}*F_{p_2}$ 能够得到 $n=p_1^{k_1}p_2^{k_2}$ 处的所有值，可以进一步推广到不交的质数集合卷积的情况。

快速卷积性函数

综上，我们将卷一个积性函数转化为了卷 $\pi(n)$ 个只和单个素数相关的函数。把这些函数分别卷到 G 上去，就能得到答案。

$$(G * F_p)(n) = \sum_{d|n} G(n/d)F_p(d) = \sum_{p^k|n} G(n/p^k)F_p(p^k)$$

枚举每个 p^k 的倍数计算该和式。对于单个质数 p，其贡献的复杂度为 $O\left(\sum_{k=1}^{\infty} n/p^k\right) = O(\frac{n}{p-1}) = O(\frac{n}{p})$。

总复杂度是 $O(\sum_p^n \frac{n}{p}) = O(n \log\log n)$。

此外，对于每个 p^k 都需要求出 $F(p^k)$。前文已经证明，这样的 p^k 的个数是 $O(n/\log n)$ 的。

Luogu P6222 「P6156 简单题」加强版

题意：给出 k ，多次给出 n ，求

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(i+j)^k\mu^2(\gcd(i,j))\gcd(i,j)$$

$n \le 10^7, K < 2^{31}, T \le 10^4$

题解：设 $f(n)=\mu^2(n)n,\ g=f*\mu$ ，用 g 来迫害 f 。

$$原式=\sum_{i=1}^{n}\sum_{j=1}^{n}(i+j)^k f(gcd(i,j))$$

$$=\sum_{i=1}^{n}\sum_{j=1}^{n}(i+j)^k\sum_{d|i,d|j}g(d)$$

$$=\sum_{d=1}^{n}g(d)\sum_{d|i}^{n}\sum_{d|j}^{n}(i+j)^k$$

$$=\sum_{d=1}^{n}g(d)d^k\sum_{i=1}^{\lfloor n/d\rfloor}\sum_{j=1}^{\lfloor n/d\rfloor}(i+j)^k$$

Luogu P6222 「P6156 简单题」加强版

记 $S(n)=\sum_{i=1}^{n}\sum_{j=1}^{n}(i+j)^k$ 。线性筛出 id_k 后，对 S 进行差分，可以发现是 id_k 的部分和。$O(n)$ 预处理 S 之后

$$\text{原式}=\sum_{d=1}^{n} g(d)d^k S(\lfloor n/d \rfloor)$$

已经可以整除分块，但是我们不满足于这个形式，进行差分。将答案设为 $R(n)$ ，有

$$\begin{aligned}\Delta R(n)&=\sum_{d=1}^{n} g(d)d^k S(\lfloor n/d \rfloor)-\sum_{d=1}^{n-1} g(d)d^k S(\lfloor (n-1)/d \rfloor)\\&=\sum_{d|n} g(d)d^k \Delta S(n/d)\end{aligned}$$

记 $w=g\cdot id_k$ ，则有 $\Delta R=w*S$ 。w 是积性函数，用前文技巧可做到 $O(n\log\log n+T)$ 。

Loj2476. 「2018 集训队互测 Day 3」蒜头的奖杯

以下将 $\gcd(a,b)$ 简写为 (a,b)。

题意：给出 n 以及六个长为 n 的序列 A,B,C,D,E,F ，求

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}A[i]B[j]C[k]D[(i,j)]E[(i,k)]F[(j,k)]$$

$n \le 10^5$

题解：这是个奇怪的三重 Σ 问题。

对于类似 P4449 的两重 Σ 问题，我们可以直接把 (i,j) 迫害掉。然而此时是三重 Σ ，有 $\binom{3}{2}=3$ 个 gcd ，一起迫害掉，则很难处理，须采用不对称策略。

记 $E_*=E*\mu, F_*=F*\mu$ ，可以用 E_*,F_* 来迫害 $E[(i,j)],F[(i,j)]$ 。

Loj2476. 「2018 集训队互测 Day 3」蒜头的奖杯

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} A[i]B[j]C[k]D[(i,j)]E[(i,k)]F[(j,k)]$$

$$=\sum_{k=1}^{n} C[k]\sum_{i=1}^{n}\sum_{j=1}^{n} A[i]B[j]D[(i,j)]\sum_{d|i,d|k} E^*[d]\sum_{t|j,t|k} F^*[t]$$

$$=\sum_{d=1}^{n} E^*[d]\sum_{t=1}^{n} F^*[t]\sum_{d|k,t|k} C[k]\sum_{d|i}^{n}\sum_{t|j}^{n} A[i]B[j]D[(i,j)]$$

$$=\sum_{\mathrm{lcm}(d,t)\le n}^{n} E^*[d]F^*[t]C_S[\mathrm{lcm}(d,t)]\sum_{d|i}^{n}\sum_{t|j}^{n} A[i]B[j]D[(i,j)]$$

其中 $C_S(k)=\sum_{k|d}^{n} C[d]$。

Loj2476. 「2018 集训队互测 Day 3」蒜头的奖杯

接着考虑枚举 $r=(d,t)$，则有 $lcm(d,t)=dt/r$。

$$=\sum_{r=1}^{n}\sum_{(d,t)=r,dt/r\le n}E^*[d]F^*[t]SC[dt/r]\sum_{d|i}^{n}\sum_{t|j}^{n}A[i]B[j]D[(i,j)]$$

$$=\sum_{r=1}^{n}\sum_{(d,t)=1,dt\le n/r}E^*[dr]F^*[tr]SC[dtr]\sum_{dr|i}^{n}\sum_{tr|j}^{n}A[i]B[j]D[(i,j)]$$

当 r 确定时，不难发现上式只会涉及到为 r 的倍数的位置。我们取出这些位置，将其分解成若干个子问题。形如:

$$\sum_{(d,t)=1,dt\le m}E^*[d]F^*[t]SC[dt]\sum_{d|i}^{m}\sum_{t|j}^{m}A[i]B[j]D[(i,j)]$$

Loj2476. 「2018 集训队互测 Day 3」蒜头的奖杯

对于给定的一对 d,t ，后方式子可化为：

$$\sum_{d|i}^{m}\sum_{t|j}^{m}A[i]B[j]D[(i,j)]$$
$$=\sum_{d|i}^{m}\sum_{t|j}^{m}A[i]B[j]\sum_{r|i,r|j}^{m}D^*[r]$$
$$=\sum_{d|i}^{m}A[i]\sum_{r|i}^{m}D^*[r]\sum_{r|j}^{m}[t|j]B[j]$$

给定 t 之后，对于所有 $d\in 1\sim m$ ，这个式子可以一层层预处理出来。

对每个 r 预处理 $G_1(r)=\sum_{r|j}^{m}[t|j]B[j]$ ，是倍数求和。

对每个 i 预处理 $G_2(i)=\sum_{r|i}^{m}D^*[r]G1(r)$ ，是约数求和。

对每个 d 预处理出 $G_3(d)=\sum_{d|i}^{m}A[i]G2(i)$ ，是倍数求和。

使用高维前缀（后缀）和可以做到 $O(m\log\log m)$ 。

Loj2476. 「2018 集训队互测 Day 3」蒜头的奖杯

由于 $dt \le m$ ，通过合适的交换（与容斥）可以使得 $t \le \sqrt{m}$ 。枚举 t ，计算出各个 d 的贡献 $G3(d)$ 求和。

子问题总复杂度为 $O(m\sqrt{m}\log\log m)$ 。

各个子问题相加，总复杂度为

$$O\Big(\sum_{i=1}^{n}(\tfrac{n}{i})^{1.5}\log\log n\Big) = O(n\sqrt{n}\log\log n)$$ 。

此外，容易在 $O(n\log n)$ 内求出 D^*, E^*, F^*, SC ，等等辅助数组。

有一个小细节，E_*, F_* 要在提取子问题之前卷 μ ，而 D_* 要在提取之后卷。(因为化式子的顺序不同)

Thanks!