数论筛法

清华大学 何昊天

2019 年 6 月 11 日

数论函数

若 $f(n)$ 的定义域为正整数，值域为复数，则称 $f(n)$ 为数论函数，大多数情况下我们都讨论值域为正整数的积性函数

若 $f(n)$ 为数论函数，且对任意两个互质的正整数 a, b，都有 $f(a \cdot b) = f(a) \cdot f(b)$，则称 $f(n)$ 为积性函数，注意积性函数 $f(n)$ 蕴含了 $f(1) = 1$，显然两个积性函数的乘积也是积性函数

若 $f(n)$ 为数论函数，且对任意两个正整数 a, b，都有 $f(a \cdot b) = f(a) \cdot f(b)$，则称 $f(n)$ 为完全积性函数

我们最感兴趣的是积性函数，同时要注意积性函数中有哪些是完全积性函数

常见的积性函数

除数函数 $\sigma_k(n) = \sum_{d|n} d^k$，表示 n 的约数的 k 次幂之和，有两个特殊的除数函数：

(i) 约数个数函数 $\tau(n) = \sigma_0(n) = \sum_{d|n} 1$，表示 n 的约数个数

(ii) 约数和函数 $\sigma(n) = \sigma_1(n) = \sum_{d|n} d$，表示 n 的约数之和，注意区分 $\sigma^k(n)$ 和 $\sigma_k(n)$，它们都是积性函数，但含义不同

欧拉函数 $\phi(n)$，表示 $1 \sim n$ 中与 n 互质的数的个数

莫比乌斯函数 $\mu(n)$，$\mu(1) = 1$，若 $n > 1$ 有多重质因数，则 $\mu(n) = 0$，否则 $\mu(n) = (-1)^r$，r 表示 n 的质因数个数

常见的完全积性函数

刚才介绍的积性函数都不是完全积性函数，在竞赛中用到的完全积性函数都很简单，这里相当于做一个符号的约定

狄利克雷卷积单位 $\epsilon(n) = [n = 1]$

常函数 $1(n) = 1$，注意其它恒等函数都不是积性函数

单位函数 $id(n) = n$，这里要注意区分函数和常数

幂函数 $id_k(n) = n^k$，可以看作 k 个单位函数的乘积

狄利克雷卷积

设 $f(n), g(n)$ 是两个数论函数，我们可以构造两个新的数论函数：

(i) 乘积函数：$(f \cdot g)(n) = f(n) \cdot g(n)$

(ii) 狄利克雷卷积函数：$(f \times g)(n) = \sum_{d|n} f(d)g(\frac{d}{n})$

定理：若 f, g 都是积性函数，则两个新函数都是积性函数，若 f, g 都是完全积性函数，则两个新函数也都是完全积性函数

证明：将上述式子依次逐点展开即可得证

狄利克雷卷积的性质

交换律： $f \times g = g \times f$

结合律： $(f \times g) \times h = f \times (g \times h)$

分配律： $(f + g) \times h = (f \times h) + (g \times h)$

单位元： $f \times \epsilon = \epsilon \times f$

证明： 将上述式子依次逐点展开即可得证

狄利克雷卷积相当于给了一个观点更高的方式来看待一些和式，熟练运用上面的性质不仅可以大幅简化推导过程，还有助于理解问题本质

两个重要结论

结论一：$\mu \times 1 = \epsilon$

证明：假设 $n = 1$，则显然 $\mu \times 1(n) = 1$

假设 $n \neq 1$，设 n 有 r 个质因数，因为有多重质因数时 $\mu(d) = 0$，故只需考虑所有的只含一重质因数的 d，由二项式定理即可得证

结论二：$\phi \times 1 = id$

证明：假设 p^c 为质数的幂，则显然 $\phi \times 1(p^c) = p^c$

其它情况由 $\phi \times 1$ 和 id 都是积性函数即可得证

狄利克雷卷积的计算

问题：已知积性函数 $f(1)\sim f(n)$ 和 $g(1)\sim g(n)$ 的值，如何快速求出它们的狄利克雷卷积 $(f\times g)(1)\sim(f\times g)(n)$ 的值？

解法一：直接计算，枚举每个数对其倍数的贡献，时间复杂度 $O(nlogn)$

解法二：线性筛，因为 $f\times g$ 也是积性函数，所以一定可以在 $O(n)$ 的时间复杂度内完成计算，但这个方法要具体问题具体分析，这里只是提供了一种理论上存在的方法

莫比乌斯反演定理

莫比乌斯反演定理：若 $f\times g=1$，则 $g\times\mu=f$，反之亦然成立

我们用卷积的性质给出一个证明：

$$f\times 1=g\Rightarrow(f\times 1)\times\mu=g\times\mu\Rightarrow f\times(1\times\mu)=g\times\mu$$

$$\Rightarrow f\times\epsilon=g\times\mu\Rightarrow f=g\times\mu$$

$$g\times\mu=f\Rightarrow(g\times\mu)\times 1=f\times 1\Rightarrow g\times(\mu\times 1)=f\times 1$$

$$\Rightarrow g\times\epsilon=f\times 1\Rightarrow g=f\times 1$$

由该定理可以直接得到两个推论：

(i) $\phi\times 1=id\Leftrightarrow id\times\mu=\phi$

(ii) $\mu\times 1=\epsilon\Leftrightarrow\epsilon\times 1=1$

一个重要推论

设 f 是积性函数，假设对于正整数 n 有 $n = \prod_{i=1}^{t} p_i^{k_i}$，则有：

(i) $f(n) = \prod_{i=1}^{t} f(p_i^{k_i})$

(ii) 若 f 还是完全积性函数，则 $f(n) = \prod_{i=1}^{t} f(p_i)^{k_i}$

考虑 f 和常函数 1 的狄利克雷卷积，假设 $g = f \times 1$，则 $g(n) = \sum_{d|n} f(d)$

推论：若 $n = \prod_{i=1}^{t} p_i^{k_i}$，则 $g(n) = \prod_{i=1}^{t}(\sum_{j=0}^{k_i} f(p_i^j))$

证明：假设有 $k'_1, k'_2, \cdots, k'_t$ 满足 $0 \leq k'_i \leq k_i$，考虑因子 $\prod_{i=1}^{t} f(p_i^{k'_i})$，每个这样的因子恰好在等式两边各出现一次，故推论得证

莫比乌斯函数和欧拉函数的关系

已知 $\phi \times 1 = id$，由莫比乌斯反演定理得 $\mu \times id = \phi$

计算等式两边在 n 处的值得 $\phi(n) = \sum_{d|n} \mu(d)\frac{n}{d}$，整理得到 $\frac{\phi(n)}{n} = \sum_{d|n} \frac{\mu(d)}{d}$

解释一下这个关系，我们知道若 $n = \prod_{i=1}^{t} p_i^{k_i}$，则 $\phi(n) = n\prod_{i=1}^{t}(1 - \frac{1}{p_i})$，也即 $\frac{\phi(n)}{n} = \prod_{i=1}^{t}(1 - \frac{1}{p_i})$，假设有 $k_1', k_2', \cdots, k_t'$ 满足 $0 \le k_i' \le k_i$，考虑因子 $\prod_{i=1}^{t} \frac{1}{p_i^{k_i'}}$，每个这样的因子恰好在等式两边各出现一次，且符号刚好可由莫比乌斯函数计算得到

换句话说，这个关系式可以看作欧拉函数计算式的一个推论

欧拉函数的线性筛

注意到 $\phi(p^k) = p^{k-1}(p-1)$，对于一个数 n，假设 p 是它的最小质因数，且最大重数为 k，则我们应该依次算出 $\frac{n}{p^k}, \frac{n}{p^{k-1}}, \cdots, n$ 的欧拉函数值来

实际上，我们会用 $\frac{n}{p^t}$ 的 p 倍来筛掉 $\frac{n}{p^{t-1}}$，依次类推，所以只需要在每次筛去的时候，用 $\phi(\frac{n}{p^t})$ 乘上 p 即得 $\phi(\frac{n}{p^{t-1}})$

但有一个例外，当 $t = k$ 时，我们不应乘以 p，而应该乘以 $p-1$，不过这并不难办，只需要在执行上述流程时，判断 $\frac{n}{p^t} \bmod p$ 是否等于 0，若不是，则乘以 $p-1$，否则就乘以 p

递归地考虑 n 除了 p 以外的其它质因数，也即 $\frac{n}{p^k}$ 的所有质因数，它们共同贡献了 $\phi(\frac{n}{p^k})$，所以这个方法是正确的

莫比乌斯函数的线性筛

类似于欧拉函数的线性筛，我们只需要简单区分一个质数是否是当前枚举的数的因数即可

对于一个数 n，假设 p 是它的最小质因数，且最大重数为 k，假设我们正准备用 $\frac{n}{p^t}$ 的 p 倍来筛掉 $\frac{n}{p^{t-1}}$，若 $t \neq k$，则 p 为 n 的多重质因数，有 $\mu(\frac{n}{p^{t-1}})$，否则 $\mu(\frac{n}{p^{t-1}}) = -\mu(\frac{n}{p^t})$

递归地考虑 n 除了 p 以外的其它质因数，也即 $\frac{n}{p^k}$ 的所有质因数，它们共同贡献了 $\mu(\frac{n}{p^k})$，所以这个方法是正确的

除数函数的筛法

利用 Euler 筛法预处理除数函数的值不太容易，但是根据除数函数的定义，我们很容易得到一个利用 Eratosthenes 筛法的 $O(n \log\log n \log k)$ 的算法预处理出 σ_k 的值，一般情况下这个复杂度已经可以接受了

考虑约数和函数，如果我们筛出了其前 n 项的值，自然也就得到了其前 n 项和的值，时间复杂度为 $O(n)$ 或 $O(n \log\log n)$

如果我们只是想计算 $\sum_{i=1}^{n} \sigma(i)$ 的值，能否有一个低于 $O(n)$ 的复杂度呢？

约数和函数前缀和

不妨设 $\Sigma(n)=\sum_{i=1}^{n}\sigma(i)$，如果使用筛法求出 $\sigma(1)\sim\sigma(n)$ 的值再进行累加，我们实际上计算出了 $\Sigma(1)\sim\Sigma(n)$ 的值，但如果我们只需要 $\Sigma(n)$ 的值，那可以认为这个做法是进行了一些冗余的计算，我们尝试如下化简：

$$\sum_{i=1}^{n}\sigma(i)=\sum_{i=1}^{n}\sum_{j=1}^{i}[i\%j=0]\cdot j=\sum_{i=1}^{n}\sum_{j=1}^{n}[i\%j=0]\cdot j$$
$$=\sum_{i=1}^{n}\sum_{j=1}^{n}[j\%i=0]\cdot i=\sum_{i=1}^{n}i\sum_{j=1}^{n}[j\%i=0]=\sum_{i=1}^{n}i\cdot\lfloor\frac{n}{i}\rfloor$$

由于 $\lfloor\frac{n}{i}\rfloor$ 只有 $O(\sqrt{n})$ 种不同的取值，所以我们可以在 $O(\sqrt{n})$ 的时间复杂度内计算 $\Sigma(n)$ 的值，这比筛出 $\sigma(n)$ 值的做法要优秀很多

莫比乌斯函数前缀和（梅滕斯函数）

定义梅滕斯函数 $M(n)=\sum_{i=1}^{n}\mu(i)$，也即莫比乌斯函数的前缀和，现在需要求出 $M(n)$ 的值，与约数和函数类似，如果筛出 $\mu(1)\sim\mu(n)$ 的值实际上获得了 $M(1)\sim M(n)$ 的值，这里面有冗余的部分，我们希望得到一个复杂度低于 $O(n)$ 的做法，考虑如下化简：

$$\mu\times 1=\epsilon\Rightarrow\sum_{d|n}\mu(d)=[n=1]\Rightarrow\sum_{i=1}^{n}\sum_{d|i}\mu(d)=1$$

$$\Rightarrow\sum_{i=1}^{n}\sum_{j=1}^{\lfloor\frac{n}{i}\rfloor}\mu(j)=1\Rightarrow\sum_{i=1}^{n}\mu(i)=1-\sum_{i=2}^{n}\sum_{j=1}^{\lfloor\frac{n}{i}\rfloor}\mu(j)$$

也即 $M(n)=1-\sum_{i=2}^{n}M(\lfloor\frac{n}{i}\rfloor)$，换句话说我们只需要计算这个递归式即可

递归计算梅滕斯函数

根据 $M(n) = 1 - \sum_{i=2}^{n} M(\lfloor \frac{n}{i} \rfloor)$，由于 $\lfloor \frac{n}{i} \rfloor$ 只有 $O(\sqrt{n})$ 种不同的取值，所以我们只需计算 $O(\sqrt{n})$ 个 $M(\lfloor \frac{n}{i} \rfloor)$ 的值就可以得到 $M(n)$ 的值了

考虑递归，时间复杂度为 $T(n) = O(\sqrt{n}) + \sum_{i=1}^{\sqrt{n}} (T(i) + T(\frac{n}{i}))$，将式子右侧展开一阶得 $O(\sqrt{n}) + \sum_{i=1}^{\sqrt{n}} (O(i) + O(\frac{n}{i}))$，注意到高阶量必然小于 $O(\sqrt{n})$，故可以忽略不计，有 $T(n) = O(\sqrt{n}) + \sum_{i=1}^{\sqrt{n}} (O(i) + O(\frac{n}{i})) = \sum_{i=1}^{\sqrt{n}} (O(i) + O(\frac{n}{i}))$

运用积分进行估计，有 $\sum_{i=1}^{\sqrt{n}} (O(i) + O(\frac{n}{i})) \sim O(\int_{1}^{\sqrt{n}} \sqrt{x} dx) \sim O(n^{\frac{3}{4}})$，所以直接递归计算梅滕斯函数的时间复杂度为 $O(n^{\frac{3}{4}})$

杜教筛

递归计算已经给出了一个低于 $O(n)$ 的时间复杂度，但其实还有优化的余地，原因在于当递归层数较多时，对于较小的 n，$M(n)$ 实际上会被重复计算很多次，如果我们能进行记忆化，还能继续改善时间复杂度

由于 n 可能很大，不能对所有的 n 进行记忆化，考虑筛出莫比乌斯函数的前 k 项并直接计算 $M(1) \sim M(k)$，我们将这部分视为记忆化的结果即可

这个做法就称为杜教筛：对于 $n \leq k$ 的部分，直接使用线性筛的结果给出 $M(n)$ 的值，对于 $n > k$ 的部分，采用刚才的递归法进行计算

杜教筛复杂度

根据描述，杜教筛递归部分时间复杂度为 $T(n) = O(\sqrt{n}) + \sum_{i=1}^{\frac{n}{k}} T(\frac{n}{i})$，求和部分为大于 k 的需要递归的部分，这里假设 $k > \sqrt{n}$

对递归的部分展开一阶得 $O(\sqrt{n}) + \sum_{i=1}^{\frac{n}{k}} O(\frac{n}{i})$，同理可以去掉高阶小量，得到 $T(n) = \sum_{i=1}^{\frac{n}{k}} O(\frac{n}{i})$

运用积分进行估计，有 $\sum_{i=1}^{\frac{n}{k}} O(\frac{n}{i}) \sim O(\int_1^{\frac{n}{k}} \sqrt{\frac{n}{x}} dx) \sim O(\frac{n}{\sqrt{k}})$，注意到杜教筛需要 $O(k)$ 的时间来线性筛，故总时间复杂度为 $O(k) + O(\frac{n}{\sqrt{k}})$，利用均值不等式知当 $k = n^{\frac{2}{3}}$ 时，复杂度达到最优为 $O(n^{\frac{2}{3}})$

至此，我们可以 $O(n^{\frac{2}{3}})$ 地计算梅滕斯函数 $M(n)$ 的值

欧拉函数前缀和

下面来计算欧拉函数的前缀和 $\Phi(n)=\sum_{i=1}^{n}\phi(i)$，利用 $\phi\times 1=id$ 化简：

$$\phi\times 1=id\Rightarrow\sum_{d|n}\phi(d)=n\Rightarrow\sum_{i=1}^{n}\sum_{d|i}\phi(d)=\frac{n(n+1)}{2}$$

$$\Rightarrow\sum_{i=1}^{n}\sum_{j=1}^{\lfloor\frac{n}{i}\rfloor}\phi(j)=\frac{n(n+1)}{2}\Rightarrow\sum_{i=1}^{n}\phi(i)=\frac{n(n+1)}{2}-\sum_{i=2}^{n}\sum_{j=1}^{\lfloor\frac{n}{i}\rfloor}\phi(j)$$

所以有 $\Phi(n)=\frac{n(n+1)}{2}-\sum_{i=2}^{n}\Phi(\lfloor\frac{n}{i}\rfloor)$，和梅滕斯函数类似地使用杜教筛，可以 $O(n^{\frac{2}{3}})$ 地计算欧拉函数前缀和

一般积性函数的杜教筛

设 f 为积性函数，现在想计算其前缀和函数 $F(n)=\sum_{i=1}^{n} f(i)$

受梅滕斯函数和欧拉函数前缀和启发，我们考虑一个卷积式 $f\times g=h$，然后进行如下化简：

$$f\times g=h\Rightarrow \sum_{d|n} g(d)f(\frac{n}{d})=h(n)\Rightarrow \sum_{i=1}^{n}\sum_{d|i} g(d)f(\frac{i}{d})=\sum_{i=1}^{n} h(i)$$

$$\Rightarrow \sum_{d=1}^{n} g(d)\sum_{i=1}^{\lfloor\frac{n}{d}\rfloor} f(i)=\sum_{i=1}^{n} h(i)\Rightarrow \sum_{d=1}^{n} g(d)F(\lfloor\frac{n}{d}\rfloor)=\sum_{i=1}^{n} h(i)$$

$$\Rightarrow g(1)F(n)=\sum_{i=1}^{n} h(i)-\sum_{d=2}^{n} g(d)F(\lfloor\frac{n}{d}\rfloor)$$

如果 g,h 的前缀和相对容易计算，我们就能够利用杜教筛得到一个时间复杂度低于 $O(n)$ 的方法来计算 $F(n)$

$id \cdot \phi$ 的前缀和

我们知道 $n \cdot \phi(n)$ 也是积性函数，下面考虑如何计算 $S(n) = \sum_{i=1}^{n} i \cdot \phi(i)$ 的值

假设 $h(n) = \sum_{d|n}(d \cdot \phi(d))g(\frac{n}{d})$，关键点在于 g 的选取，这里取 $g = id$ 是最简单的，这样有 $h(n) = n\sum_{d|n}\phi(d) = n^2$，则 $\sum_{i=1}^{n} h(i) = \frac{n(n+1)(2n+1)}{6}$

套用杜教筛公式，有 $S(n) = \frac{n(n+1)(2n+1)}{6} - \sum_{d=2}^{n} d \cdot S(\lfloor \frac{n}{d} \rfloor)$，$id$ 函数前缀和很好计算，所以直接杜教筛，时间复杂度为 $O(n^{\frac{2}{3}})$

这里实际上用到了一个公式 $(id \cdot \phi) \times id = id_2$，这个式子用得也比较多，最好可以记下来，同理我们还知道 $(id_k \cdot \phi) \times id_k = id_{k+1}$

$\sum\sum lcm$ 与积性函数前缀和

回顾经典问题 $\sum_{i=1}^{n}\sum_{j=1}^{i} lcm(i,j)$ 的计算，利用莫比乌斯反演一些技巧进行变形可以解决这个问题，但实际上用杜教筛可以更容易破解此题

设 $A(n)=\sum_{i=1}^{n} lcm(i,n)$，首先进行以下化简：

$$A(n)=\sum_{i=1}^{n} lcm(i,n)=n\sum_{i=1}^{n}\frac{i}{gcd(i,n)}$$
$$=n\sum_{d|n}\sum_{i=1}^{\lfloor\frac{n}{d}\rfloor} i\cdot[gcd(\frac{n}{d},i)=1]=\frac{n}{2}(1+\sum_{d|n} d\cdot\phi(d))$$

原问题等价于计算 $\sum_{i=1}^{n} A(i)=\sum_{i=1}^{n}\frac{i}{2}(1+\sum_{d|i} d\cdot\phi(d))$，实际上需要计算的就是积性函数 $id\cdot((id\cdot\phi)\times 1)$ 的前缀和，故可以考虑用杜教筛解决

$\sum\sum lcm$ 的杜教筛解法

令 $f = id \cdot ((id \cdot \phi) \times 1)$，则 $f(n) = n\sum_{d|n} d \cdot \phi(d) = \sum_{d|n} d^2\phi(d)\frac{n}{d}$，即 $f = (id_2 \cdot \phi) \times id$，考虑公式 $(id_k \cdot \phi) \times id_k = id_{k+1}$，我们令 $g = id_2$，则有 $h = f \times g = (id_2 \cdot \phi) \times id \times id_2 = id_3 \times id$

设 $F(n) = \sum_{i=1}^{n} f(i)$，则利用杜教筛公式得：

$$F(n) = \sum_{i=1}^{n}\sum_{d|i} d^3\frac{i}{d} - \sum_{d=2}^{n} d^2 F(\lfloor\frac{n}{d}\rfloor)$$

$$= \sum_{d=1}^{n} d^2 \sum_{i=1,d|i}^{n} i - \sum_{d=2}^{n} d^2 F(\lfloor\frac{n}{d}\rfloor)$$

$$= \frac{1}{2}\sum_{d=1}^{n} d^3(1 + \lfloor\frac{n}{d}\rfloor)\lfloor\frac{n}{d}\rfloor - \sum_{d=2}^{n} d^2 F(\lfloor\frac{n}{d}\rfloor)$$

前面的前缀和部分可以 $O(\sqrt{n})$ 数论分块计算，id_2 前缀和可以 $O(1)$ 计算，因此使用杜教筛的总时间复杂度仍为 $O(n^{\frac{2}{3}})$

HDU5608

已知数论函数 f 满足 $\sum_{d|n} f(d) = n^2 - 3n + 2$

计算 $F(n) = \sum_{i=1}^{n} f(i)$ 的值

计算结果对质数 $10^9 + 7$ 取模

HDU5608 Solution

令 $g(n) = n^2 - 3n + 2$，则 f 与 g 的关系满足 $f \times 1 = g$

注意到 $g = id_2 - 3 \cdot id + 2$，而 id_2, id 和常函数的前缀和都很好计算，则 g 的前缀和也很好计算，所以考虑使用杜教筛来计算 f 的前缀和

直接套用公式，唯一的难点在于如何预处理 f 的值，根据莫比乌斯反演定理，由 $f \times 1 = g$ 知 $g \times \mu = f$，我们可以线性地预处理出 g 和 μ 的值，然后直接计算狄利克雷卷积得到 f 的值

假设预处理出 $f(1) \sim f(k)$ 的值，计算卷积的复杂度为 $O(k \log k)$，结合杜教筛递归的复杂度，这里仍然取 $k = n^{\frac{2}{3}}$，总时间复杂度为 $O(n^{\frac{2}{3}} \log n)$

BZOJ4916

给出一个正整数 n，计算：

(i) $A(n) = \sum_{i=1}^{n} \mu(i^2)$

(ii) $B(n) = \sum_{i=1}^{n} \phi(i^2)$

$1 \le n \le 10^9$，计算结果对质数 $10^9 + 7$ 取模

BZOJ4916 Solution

这是一个考察莫比乌斯函数和欧拉函数性质的题目

第一问的答案肯定是 1，因为除了 $i=1$ 以外，其余 i^2 内的质因子必然出现两次以上，莫比乌斯函数值只能为 0

第二问考虑 $i=\prod_{j=1}^{t} p_j^{k_j}$，则 $i^2=\prod_{j=1}^{t} p_j^{2k_j}$，$\phi(i^2)=\prod_{j=1}^{t}(p_j-1)\prod_{j=1}^{t} p_j^{2k_j-1}$，从 $\phi(i^2)$ 中取出一个 $i=\prod_{j=1}^{t} p_j^{k_j}$ 得 $\phi(i^2)=i\cdot\prod_{j=1}^{t}(p_j-1)\prod_{j=1}^{t} p_j^{k_j-1}$，乘积部分刚好是 $\phi(i)$ 的值，所以 $\phi(i^2)=i\cdot\phi(i)$

现在问题变成了计算 $id\cdot\phi$ 的前缀和，这个在前面已经介绍过杜教筛解法了，此处不再赘述

BZOJ3512

给出两个正整数 n, m，计算 $\sum_{i=1}^{n}\sum_{j=1}^{m}\phi(ij)$ 的值

$1 \le n \le 10^5$，$1 \le m \le 10^9$，计算结果对质数 10^9+7 取模

BZOJ3512 Solution

由于 n 较小，考虑枚举 n，问题变成计算 $S(n,m)=\sum_{i=1}^{m}\phi(in)$ 的值，当 $n=1$ 时这就是 ϕ 的前缀和，使用杜教筛可以在 $O(m^{\frac{2}{3}})$ 的复杂度内解决

设 $n=wx$，其中 w 是 n 最大的无平方因子约数，则 $\phi(in)=\phi(iwx)=\phi(iw)x$，故 $S(n,m)=x\sum_{i=1}^{m}\phi(iw)$

根据 ϕ 的计算式，假设 $d=gcd(i,w)$，则有 $\phi(iw)=\phi(i)\phi(\frac{w}{d})d$

考虑用 $\phi\times 1=id$ 来替换上式的 d，有 $\phi(iw)=\phi(i)\phi(\frac{w}{d})\sum_{e|d}\phi(\frac{d}{e})=\phi(i)\sum_{e|d}\phi(\frac{w}{e})$

代入到 $S(n,m)$ 中化简得到 $S(n,m)=x\sum_{d|w}\phi(\frac{w}{d})S(d,\lfloor\frac{m}{d}\rfloor)$，然后递归进行计算即可，注意到右侧的第二维始终是 $\lfloor\frac{m}{t}\rfloor$ 的形式，所以总共只有 $O(\sqrt{m})$ 种取值，因此总时间复杂度仅为 $O(n\sqrt{m}+m^{\frac{2}{3}})$

杜教筛的局限性

我们已经讨论了用杜教筛求解一类积性函数前缀和的方法，简单来说能够使用杜教筛求和的函数 f 需要满足如下条件：

(i) 存在函数 g 和 h 使得 $f \times g = h$，且 g 和 h 的前缀和相对容易计算

(ii) f 的前若干项相对容易计算，若不满足这一条，直接递归的复杂度为 $O(n^{\frac{3}{4}})$

可以看出，并非所有函数都可以用杜教筛来求前缀和，即使可以求，g 和 h 的构造也是有一定难度的，但杜教筛的复杂度非常优秀，仅 $O(n^{\frac{2}{3}})$ 就能解决问题，很多时候是我们的首选

如果不能杜教筛或者很难进行杜教筛，我们实际上还有其它方法可用，也即后面会介绍的洲阁筛

特殊的积性函数

考虑具有如下性质的积性函数 F:

(i) 当 $n = p$ 为质数时，$F(p) = G(p)$

(ii) 当 $n = p^c$ 为质数的幂时，$F(p) = T(p^c)$

(iii) $G(p), T(p^c)$ 为相对简单的函数，通常来说是关于 p, c 的低阶多项式或幂函数

例如，对于欧拉函数 ϕ 有 $G(p) = p - 1, T(p^c) = (p-1)p^{c-1}$，对于莫比乌斯函数 μ 有 $G(p) = -1, T(p^c) = 0$，它们都是满足上述性质的函数

注意到这个性质是可以检验的，我们之后会从这个性质出发引出洲阁筛，相比于构造性很强的杜教筛，这个方法会显得更加直接

初步转化

以下均设 F 是积性函数，具有之前描述的性质，我们希望计算 $\sum_{i=1}^{n} F(i)$ 的值

对于一个正整数 $x \le n$，显然 x 至多只有一个大于 $\sqrt{n}$ 的质因数，所以我们按照这个标准将所有的 $1 \le x \le n$ 分为两类：

(i) x 没有大于 $\sqrt{n}$ 的质因数，这类 x 的集合记为 A

(ii) x 有一个大于 $\sqrt{n}$ 的质因数，这类 x 的集合记为 B

设 P 表示质数集，根据 F 的积性，我们得到如下计算式：

$$\sum_{i=1}^{n} F(i) = \sum_{x \le n, x \in A} F(x) \left(1 + \sum_{\sqrt{n} < p \le \lfloor \frac{n}{x} \rfloor, p \in P} G(p) \right)$$

洲阁筛

$$\sum_{i=1}^{n} F(i) = \sum_{x \leq n, x \in A} F(x) \left(1 + \sum_{\sqrt{n} < p \leq \lfloor \frac{n}{x} \rfloor, p \in P} G(p) \right)$$

注意到对于每个 $F(x), x \in A$，后面乘的系数是一个与 $\lfloor \frac{n}{x} \rfloor$ 有关、不与 x 直接有关的量，由于 $\lfloor \frac{n}{x} \rfloor$ 只有 $O(\sqrt{n})$ 种不同的取值，所以我们考虑将相同取值的部分单独进行计算，设 $y = \lfloor \frac{n}{x} \rfloor$ 为一组 x 对应的 $\lfloor \frac{n}{x} \rfloor$ 的取值，我们需要计算：

(i) $\sum_{\sqrt{n} < p \leq y, p \in P} G(p)$

(ii) $\sum_{\lfloor \frac{n}{x} \rfloor = y, x \in A} F(x)$

这就是洲阁筛的基本想法，接下来我们来看这两部分分别应该如何计算

$G(p)$ 部分的计算方法

根据 F 的性质，不妨设 $G(p)$ 是一个关于 p 的低阶多项式，我们只需要对每一项单独进行计算即可

设不大于 $\sqrt{n}$ 的质数共有 m 个，依次记作 $p_1, p_2, \cdots, p_m$，$g_k[i][j]$ 表示 $[1, j]$ 以内与前 i 个质数互质的数的 k 次方之和，其中 $g_k[0][j]$ 可以利用插值 $O(k)$ 完成计算

考虑从 $g_k[i-1][j]$ 转移到 $g_k[i][j]$，考虑一个数 a，假设 a 与前 $i-1$ 个质数互质，那么 $a \cdot p_i$ 也与前 $i-1$ 个质数互质、但不与第 i 个质数互质，且满足这个性质的数都有这样的形式，而 $(a \cdot p_i)^k = a^k \cdot p_i^k$，所以有如下递推式：

$$g_k[i][j] = g_k[i-1][j] - p_i^k g_k[i][\lfloor \frac{j}{p_i} \rfloor]$$

$G(p)$ 部分的计算实现

首先来看如何从 $g_k[i][j]$ 得到答案，事实上 $\sum_{\sqrt{n}<p\le y, p\in P} G(p)$ 中 k 次方项不计常数的结果就是 $g_k[m][y]-1$，因为 $[1,y]$ 中与前 m 个质数互质的数只能是 $\sqrt{n}$ 到 y 之间的质数或 1，所以我们现在只需递推求出 $g_k[i][j]$ 即可

考虑直接朴素递推，注意到第二维始终是 $\lfloor \frac{n}{t} \rfloor$ 的形式，所以至多有 $O(\sqrt{n})$ 种取值，并且每种取值恰好对应一个需要计算的 $G(p)$ 部分，所以可以统一处理

根据质数密度公式 $\pi(n)\sim\frac{n}{\log n}$，不大于 $\sqrt{n}$ 的质数有 $O(\frac{\sqrt{n}}{\log n})$ 个，再结合第二维有 $O(\sqrt{n})$ 种取值，总的时间复杂度为 $O(\frac{n}{\log n})$

虽然 $G(p)$ 部分的计算已经低于 $O(n)$ 了，但还可以进一步对这个方法进行优化

$G(p)$ 部分的计算优化

当 $p_{i+1} > j$ 时，显然有 $g_k[i][j] = 1$，故当 $p_i \le j < p_i^2$ 时 $g_k[i-1][\lfloor \frac{j}{p_i} \rfloor] = 1$，此时递推式可以简化为 $g_k[i][j] = g_k[i-1][j] - p_i^k$

因此 $j < p_i^2$ 的部分是可以忽略不算的，换句话说对于每个 $y = \lfloor \frac{n}{x} \rfloor$，实际上只需转移不超过 $\sqrt{y}$ 的质数，总的时间复杂度为 $\sum_{i=1}^{\sqrt{n}} O(\frac{\sqrt{\frac{n}{i}}}{\log \frac{n}{i}})$

考虑积分估值，我们得到 $G(p)$ 部分的计算时间复杂度为：

$$\sum_{i=1}^{\sqrt{n}} O(\frac{\sqrt{\frac{n}{i}}}{\log \frac{n}{i}}) \sim O(\frac{\sum_{i=1}^{\sqrt{n}} \sqrt{\frac{n}{i}}}{\log n}) \sim O(\frac{\int_1^{\sqrt{n}} \sqrt{\frac{n}{x}} dx}{\log n}) \sim O(\frac{n^{\frac{3}{4}}}{\log n})$$

$F(x)$ 部分的计算方法

不妨先用线性筛预处理出所有 $x<\sqrt{n}$ 的 $F(x)$ 的值，乘上对应的系数累加进答案，所以我们实际需要计算的部分等价于 $\sum_{\sqrt{n}\le x\le n, x\in A} F(x)$

设不大于 $\sqrt{n}$ 的质数共有 m 个，从大到小倒序依次记作 $p_1, p_2, \cdots, p_m$，设 $f[i][j]$ 表示只包含其中前 i 种质因数且 $x\le j$ 的 $F(x)$ 之和，容易通过 $f[i][j]$ 计算出 $F(x)$ 部分

考虑如何计算 $f[i][j]$，枚举质因子 p_i 的幂次 c，得到以下递推式：

$$f[i][j] = f[i-1][j] + G(p_i)f[i-1][\lfloor\frac{j}{p_i}\rfloor] + \sum_{c\ge 2} T(p_i^c)f[i-1][\lfloor\frac{j}{p_i^c}\rfloor]$$

朴素递推的复杂度也是 $O(\frac{n}{\log n})$，我们考虑进行优化

$F(x)$ 部分的计算优化

同样考虑 $j < p_i^2$ 的情况，注意到这里的质数是从大到小排列的，所以此时有 $f[i][j] = f[i-1][j] + F(p_i)$，故这部分不需要进行转移，直接用已经筛出来的结果进行更新即可

与 $G(p)$ 部分的复杂度分析类似，得到优化后 $F(x)$ 部分的复杂度为 $O(\frac{n^{\frac{3}{4}}}{\log n})$

计算出了 $G(p)$ 部分与 $F(x)$ 部分的值后，用数论分块进行合并即得答案，总时间复杂度为 $O(\frac{n^{\frac{3}{4}}}{\log n})$，这就是完整的洲阁筛

同杜教筛相比，洲阁筛的复杂度略高一些，而且实现起来相对复杂，但可以处理的函数范围更广，对函数本身的要求更低

筛法

杜教筛更多的是借助狄利克雷卷积进行了一个构造，称之为“筛法”更多是习惯的问题，实际上这个算法不太有筛法的功能

但洲阁筛实际上完成了一部分筛法的功能，由于质因数是有序加入答案的，因此在使用洲阁筛的过程当中可以顺带统计一些额外的信息，利用这个性质可以解决更多的题目

有兴趣深入研究的同学可以参考 UOJ#188 Sanrd 一题，这道题需要计算 $1 \sim n$ 中所有数的次大质因子之和，其方法就是利用洲阁筛的筛法功能，由于这道题难度非常大，这里就不进行讲解了

素数个数与素数和问题

给定一个数 $n \le 10^{11}$，该如何计算不超过 n 的素数个数呢？

直接考虑洲阁筛，其实这就是 $G(p) = 1, T(p^c) = 0$ 的函数求前缀和的问题，虽然这个函数不是积性函数，但我们也可以筛出其前 $\sqrt{n}$ 项的值，故直接套用洲阁筛的模板即可解决

类似的，对 $G(p) = p, T(p^c) = 0$ 的函数用洲阁筛求前缀和，我们就可以计算出不超过 n 的素数之和

如果用常规方法考虑，很难想到区间素数个数或区间素数和能够比线性筛更快，但洲阁筛的处理方式简单直接，往往直击问题

SPOJ DIVCNT

设 $\tau(n)$ 表示 n 的约数个数，对于给定的 n，试计算：

(i) $A(n) = \sum_{i=1}^{n} \tau(i)$
(ii) $B(n) = \sum_{i=1}^{n} \tau(i^2)$
(iii) $C(n) = \sum_{i=1}^{n} \tau(i^3)$

$1 \le n \le 10^{11}$，结果不超过 long long 范围

SPOJ DIVCNT Solution

$A(n)$ 用莫比乌斯反演相关技巧就可以解决，$B(n)$ 有一个杜教筛的解法，留给大家作为练习，但前两种方法都没办法处理 $C(n)$

不妨直接考虑洲阁筛：

(i) $A(n)$ 对应 $G(p)=2, T(p^c)=c+1$ 的函数的前缀和
(ii) $B(n)$ 对应 $G(p)=3, T(p^c)=2c+1$ 的函数的前缀和
(iii) $C(n)$ 对应 $G(p)=4, T(p^c)=3c+1$ 的函数的前缀和

注意到上面几个函数实际上都是积性函数，用洲阁筛不需要进行额外的化简，能够直接套用模板解决问题，这就是其优势所在

积性函数前缀和的其它求法

除了杜教筛和洲阁筛以外，现在常被用到的积性函数前缀和求法还有两种：Min_25 筛和扩展埃拉托色尼筛法

Min_25 筛的复杂度为 $O(n^{1-\omega})$，其思路和洲阁筛比较接近，实际效率较高，甚至能解决 $n \le 10^{13}$ 范围内的问题，但难度更大，感兴趣的同学可以去参考 2018 年集训队论文《一些特殊的数论函数求和问题》

扩展埃拉托色尼筛法的复杂度渐进为 $O(n^{0.7})$，其思想顾名思义是埃拉托色尼筛法的一个扩展，主要优势在于只需很小的空间复杂度，因此在 $n \le 10^{12}$ 范围内效果很好，感兴趣的同学可以参考 SPOJ 上的文章 Tutorial: Extended Eratosthenes Sieve

END

Thank You For Listening!

Q&A