数论进阶

HocRiser

吉林大学 20 级唐计

2021 年 1 月 27 日

1. 预备知识
2. 积性函数优化算法
3. 杜教筛

1 预备知识
- 素数计数
- 数论函数与积性函数
- 常见积性函数

2 积性函数优化算法

3 杜教筛

- 令素数计数函数 $\pi(n)$ 表示不超过 n 的素数个数。我们有如下的素数定理：

$$\pi(n) \sim \frac{n}{\ln n}$$

- 令素数计数函数 $\pi(n)$ 表示不超过 n 的素数个数。我们有如下的素数定理：
$$\pi(n) \sim \frac{n}{\ln n}$$
- 推论：n 附近的素数密度近似是 $\frac{1}{\ln n}$

- 令素数计数函数 $\pi(n)$ 表示不超过 n 的素数个数。我们有如下的素数定理：
$$\pi(n) \sim \frac{n}{\ln n}$$
- 推论：n 附近的素数密度近似是 $\frac{1}{\ln n}$
- 第 n 个素数 $p_n \sim n \ln n$

1. 预备知识
   - 素数计数
   - 数论函数与积性函数
   - 常见积性函数
2. 积性函数优化算法
3. 杜教筛

- 定义域为正整数、值域是复数的子集的函数称为数论函数。

- 定义域为正整数、值域是复数的子集的函数称为数论函数。
- 设 f 是数论函数，若 $\forall a, b \in \mathbb{N}*$ 且 $a \perp b$，$f(ab) = f(a)f(b)$，则称 f 是积性函数。

- 定义域为正整数、值域是复数的子集的函数称为数论函数。
- 设 f 是数论函数，若 $\forall a, b \in \mathbb{N}*$ 且 $a \perp b$，$f(ab) = f(a)f(b)$，则称 f 是积性函数。
- 若 $\forall a, b \in \mathbb{N}*$，都有 $f(ab) = f(a)f(b)$，则称 f 是完全积性的。

- 若 $f(n)$ 是积性函数，且 $n=p_1^{\alpha_1}p_2^{\alpha_2}\cdots p_s^{\alpha_s}$ 是 n 的标准分解，则有
$$f(n)=f(p_1^{\alpha_1})f(p_2^{\alpha_2})\cdots f(p_s^{\alpha_s})$$

- 若 $f(n)$ 是积性函数，且 $n=p_1^{\alpha_1}p_2^{\alpha_2}\cdots p_s^{\alpha_s}$ 是 n 的标准分解，则有
$$f(n)=f(p_1^{\alpha_1})f(p_2^{\alpha_2})\cdots f(p_s^{\alpha_s})$$
- 因此积性函数 f 可以转化为研究 $f(p^{\alpha})$，即 f 在素数和素数的幂上的取值。

- 若 $f(n)$ 是积性函数，且 $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_s^{\alpha_s}$ 是 n 的标准分解，则有
$$f(n) = f(p_1^{\alpha_1}) f(p_2^{\alpha_2}) \cdots f(p_s^{\alpha_s})$$
- 因此积性函数 f 可以转化为研究 $f(p^\alpha)$，即 f 在素数和素数的幂上的取值。
- 对于完全积性函数，往往只需研究 f 在素数上的取值。

- 若 $f(n)$ 是积性函数，且 $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_s^{\alpha_s}$ 是 n 的标准分解，则有

$$f(n) = f(p_1^{\alpha_1}) f(p_2^{\alpha_2}) \cdots f(p_s^{\alpha_s})$$

- 若 $f(n)$ 是积性函数，且 $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_s^{\alpha_s}$ 是 n 的标准分解，则有
$$f(n) = f(p_1^{\alpha_1}) f(p_2^{\alpha_2}) \cdots f(p_s^{\alpha_s})$$
- 因此积性函数 f 可以转化为研究 $f(p^{\alpha})$，即 f 在素数和素数的幂上的取值。

- 若 $f(n)$ 是积性函数，且 $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_s^{\alpha_s}$ 是 n 的标准分解，则有
$$f(n) = f(p_1^{\alpha_1}) f(p_2^{\alpha_2}) \cdots f(p_s^{\alpha_s})$$
- 因此积性函数 f 可以转化为研究 $f(p^{\alpha})$，即 f 在素数和素数的幂上的取值。
- 对于完全积性函数，往往只需研究 f 在素数上的取值。

- 若 $f(n)$ 是积性函数，且 $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_s^{\alpha_s}$ 是 n 的标准分解，则有
$$f(n) = f(p_1^{\alpha_1}) f(p_2^{\alpha_2}) \cdots f(p_s^{\alpha_s})$$
- 因此积性函数 f 可以转化为研究 $f(p^{\alpha})$，即 f 在素数和素数的幂上的取值。
- 对于完全积性函数，往往只需研究 f 在素数上的取值。
- 对于 n 以内 f 函数的计算，可以在 *Euler* 筛法的过程中线性得到结果。

1. 预备知识
   - 素数计数
   - 数论函数与积性函数
   - 常见积性函数
2. 积性函数优化算法
3. 杜教筛

- 单位函数 $\epsilon(n)$

- 单位函数 $\epsilon(n)$
- $$\epsilon(n) = [n = 1] = \begin{cases} 1, n = 1 \\ 0, n \neq 1 \end{cases}$$

- 单位函数 $\epsilon(n)$
- $$\epsilon(n) = [n = 1] = \begin{cases} 1, n = 1 \\ 0, n \neq 1 \end{cases}$$
- 除数函数 $\sigma_k(n) = \sum_{d|n} d^k$，$\sigma_0(n)$ 常记作 $d(n)$，约数和 $\sigma_1(n)$ 常记作 $\sigma(n)$。

- 单位函数 $\epsilon(n)$
- $$\epsilon(n) = [n = 1] = \begin{cases} 1, n = 1 \\ 0, n \neq 1 \end{cases}$$
- 除数函数 $\sigma_k(n) = \sum_{d|n} d^k$，$\sigma_0(n)$ 常记作 $d(n)$，约数和 $\sigma_1(n)$ 常记作 $\sigma(n)$。
- 幂函数 $Id_k(n) = n^k$，$Id = Id_1$

- 欧拉函数 $\varphi(n)$ 表示不超过 n 且与 n 互质的正整数的个数

$$n=\sum_{d|n}\varphi(d)$$

- 欧拉函数 $\varphi(n)$ 表示不超过 n 且与 n 互质的正整数的个数

$$n = \sum_{d|n} \varphi(d)$$

- 莫比乌斯函数 μ，

$$\mu(n) = \begin{cases} 1, & n = 1 \\ (-1)^s, & n = p_1 p_2 \ldots p_s \\ 0, & \text{otherwise} \end{cases}$$

其中 $p_1, \ldots, p_s$ 是不同素数。

1. 预备知识
2. 积性函数优化算法
   - Dirichlet 卷积
   - Mobius 反演
3. 杜教筛

1 预备知识

2 积性函数优化算法
- Dirichlet 卷积
- Mobius 反演

3 杜教筛

- 设 f，g 是数论函数，考虑数论函数 h 满足

$$h(n)=\sum_{d|n} f(d)g(\frac{n}{d})$$

- 设 f , g 是数论函数，考虑数论函数 h 满足

$$h(n) = \sum_{d|n} f(d)g(\frac{n}{d})$$

- 则称 h 为 f 和 g 的 Dirichlet 卷积，记作 $h = f * g$

- 单位函数 ϵ 是 Dirichlet 卷积的单位元，即对于任意函数 f ，有 $\epsilon * f = f * \epsilon = f$

- 单位函数 ϵ 是 Dirichlet 卷积的单位元，即对于任意函数 f ，有 $\epsilon * f = f * \epsilon = f$
- Dirichlet 卷积满足交换律、结合律和分配律

$$f * g = g * f$$

$$(f * g) * h = f * (g * h)$$

$$(f + g) * h = f * h + g * h$$

$$(xf) * g = x(f * g)$$

- 单位函数 ϵ 是 Dirichlet 卷积的单位元，即对于任意函数 f，有 $\epsilon * f = f * \epsilon = f$
- Dirichlet 卷积满足交换律、结合律和分配律

$$f * g = g * f$$

$$(f * g) * h = f * (g * h)$$

$$(f + g) * h = f * h + g * h$$

$$(xf) * g = x(f * g)$$

- 如果 f，g 都是积性函数，那么 $f * g$ 也是积性函数

- 单位函数 ϵ 是 Dirichlet 卷积的单位元，即对于任意函数 f，有 $\epsilon * f = f * \epsilon = f$
- Dirichlet 卷积满足交换律、结合律和分配律

$$f * g = g * f$$

$$(f * g) * h = f * (g * h)$$

$$(f + g) * h = f * h + g * h$$

$$(xf) * g = x(f * g)$$

- 如果 f, g 都是积性函数，那么 $f * g$ 也是积性函数
- 设 $f \cdot g(x) = f(x) \times g(x)$，$f$ 是完全积性函数，g, h 是数论函数，则 $(f \cdot g) * (f \cdot h) = f \cdot (g * h)$

- 用 1 表示取值恒为 1 的常函数，则除数函数的定义可以写为 $\sigma_k = 1 * Id_k$

- 用 1 表示取值恒为 1 的常函数，则除数函数的定义可以写为 $\sigma_k = 1 * Id_k$
- 欧拉函数的性质可以写为 $Id = \varphi * 1$

- 用 1 表示取值恒为 1 的常函数，则除数函数的定义可以写为 $\sigma_k = 1 * Id_k$
- 欧拉函数的性质可以写为 $Id = \varphi * 1$
- 莫比乌斯函数的性质可以写为 $\epsilon = \mu * 1$

- 欧拉函数性质证明

- 欧拉函数性质证明
  - 将 1 到 n 之间的所有正整数 i 按照与 n 的最大公约数 $d=\gcd(i,n)$ 分类，我们分别统计 d 相同的类中 i 的个数：考虑 $\gcd(\frac{i}{d},\frac{n}{d})=1$，我们统计了满足这个条件的 $\frac{i}{d}$ 的个数就等价于统计了原类中 i 的个数，而这样的 $\frac{i}{d}$ 实际上就是小于等于 $\frac{n}{d}$ 且与 $\frac{n}{d}$ 互质的数的个数，就是 $\varphi(\frac{n}{d})$，而这里的 $\frac{n}{d}$ 又与 $\sum_{d\mid n}\varphi(d)$ 中的 d 是一一对应的，我们对每一类的个数求和写成式子就是 $\sum_{d\mid n}\varphi(d)$，这每个数都会被统计恰好 1 次，总和正好是 n

- 欧拉函数性质证明
  - 将 1 到 n 之间的所有正整数 i 按照与 n 的最大公约数 $d=\gcd(i,n)$ 分类，我们分别统计 d 相同的类中 i 的个数：考虑 $\gcd(\frac{i}{d},\frac{n}{d})=1$ ，我们统计了满足这个条件的 $\frac{i}{d}$ 的个数就等价于统计了原类中 i 的个数，而这样的 $\frac{i}{d}$ 实际上就是小于等于 $\frac{n}{d}$ 且与 $\frac{n}{d}$ 互质的数的个数，就是 $\varphi(\frac{n}{d})$ ，而这里的 $\frac{n}{d}$ 又与 $\sum_{d|n}\varphi(d)$ 中的 d 是一一对应的，我们对每一类的个数求和写成式子就是 $\sum_{d|n}\varphi(d)$ ，这每个数都会被统计恰好 1 次，总和正好是 n
- 莫比乌斯函数性质证明：

- 欧拉函数性质证明
  - 将 1 到 n 之间的所有正整数 i 按照与 n 的最大公约数 $d=\gcd(i,n)$ 分类，我们分别统计 d 相同的类中 i 的个数：考虑 $\gcd(\frac{i}{d},\frac{n}{d})=1$ ，我们统计了满足这个条件的 $\frac{i}{d}$ 的个数就等价于统计了原类中 i 的个数，而这样的 $\frac{i}{d}$ 实际上就是小于等于 $\frac{n}{d}$ 且与 $\frac{n}{d}$ 互质的数的个数，就是 $\varphi(\frac{n}{d})$ ，而这里的 $\frac{n}{d}$ 又与 $\sum_{d|n}\varphi(d)$ 中的 d 是一一对应的，我们对每一类的个数求和写成式子就是 $\sum_{d|n}\varphi(d)$ ，这每个数都会被统计恰好 1 次，总和正好是 n
- 莫比乌斯函数性质证明：
  - $n=1$ 时显然成立，如果 $n>1$ ，设 n 有 s 个不同的素因子，由于 $\mu(d)\neq 0$ 当且仅当 d 无平方因子，故 d 中每个素因子的指数只能为 0 或 1 才又贡献。故有

$$\sum_{d|n}\mu(d)=\sum_{k=0}^{s}(-1)^k\binom{s}{k}=(1-1)^s=0$$

1 预备知识

2 积性函数优化算法
  - Dirichlet 卷积
  - Mobius 反演

3 杜教筛

- 莫比乌斯变换:

- 莫比乌斯变换：
- 设 f 是数论函数，定义函数 g 满足

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

则称 g 是 f 的 Mobius 变换，f 是 g 的 Mobius 逆变换

- 莫比乌斯变换:
- 设 f 是数论函数，定义函数 g 满足

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

则称 g 是 f 的 Mobius 变换，f 是 g 的 Mobius 逆变换
- 用 Dirichlet 卷积表示即为 $g = f * 1$

- Mobius 反演定理指出，Mobius 变换的充要条件是

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

- Mobius 反演定理指出，Mobius 变换的充要条件是

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

- 即 $g = f * 1 \Leftrightarrow f = g * \mu$

- Mobius 反演定理指出，Mobius 变换的充要条件是

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

- 即 $g = f * 1 \Leftrightarrow f = g * \mu$
- 证明可以使用 Dirichlet 卷积：

$$g = f * 1 \Leftrightarrow f = f * \epsilon = f * 1 * \mu = g * \mu$$

$$\mu * 1 = \epsilon$$

$$\sigma_k = Id_k * 1$$

$$Id = \varphi * 1$$

$$\varphi = \mu * Id$$

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

$$\sum_{d|n} \frac{\mu(d)}{d} = \frac{\varphi(n)}{n}$$

$$\mu(ab)=\mu(a)\mu(b)[a\perp b]$$

$$\varphi(n)=\sum_{i=1}^{n}[\gcd(i,n)=1]=\sum_{i=1}^{n}\sum_{k|i,k|n}\mu(k)=\sum_{k|n}\mu(k)\lfloor\frac{n}{k}\rfloor$$

$$\sum_{i=1}^{n}i\times[\gcd(i,n)=1]=\begin{cases}1,n=1\\ \frac{n\times\varphi(n)}{2},n>1\end{cases}$$

$$\sum_{d|n}\sum_{i=1}^{d}i\times[\gcd(i,d)=1]=\frac{1}{2}(1+\sum_{d|n}d\times\varphi(d))$$

$$\gcd(a,b)=\sum_{d|\gcd(a,b)}\varphi(d)=\sum_{d|a,d|b}\varphi(d)$$

- 求 $f(n)=\sum_{i=1}^{n}\mu(i)$

$$\sum_{k=1}^{n}(\mu*1)(k)=\sum_{i=1}^{n}1(i)\sum_{j=1}^{\lfloor\frac{n}{i}\rfloor}\mu(j)=\sum_{i=1}^{n}f(\lfloor\frac{n}{i}\rfloor)$$

$$\sum_{k=1}^{n}(\mu*1)(k)=\sum_{k=1}^{n}\epsilon(k)=1$$

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

- 求 $f(n)=\sum_{i=1}^{n}\mu(i)$

$$\sum_{k=1}^{n}(\mu*1)(k)=\sum_{i=1}^{n}1(i)\sum_{j=1}^{\lfloor\frac{n}{i}\rfloor}\mu(j)=\sum_{i=1}^{n}f(\lfloor\frac{n}{i}\rfloor)$$

$$\sum_{k=1}^{n}(\mu*1)(k)=\sum_{k=1}^{n}\epsilon(k)=1$$

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

- 时间复杂度：

$$T(n)=\sum_{i=1}^{n}O(\sqrt{\lfloor\frac{n}{i}\rfloor})=\sum_{i=1}^{\sqrt{n}}O(\sqrt{\lfloor\frac{n}{i}\rfloor})+\sum_{i=1}^{\sqrt{n}}O(i)\approx O(n^{0.75})$$

1 预备知识

2 积性函数优化算法

3 杜教筛
- 莫比乌斯函数前缀和
- 欧拉函数前缀和
- 变形

- 求 $f(n)=\sum_{i=1}^{n}\varphi(i)$

$$\sum_{k=1}^{n}(\varphi*1)(k)=\sum_{i=1}^{n}1(i)\sum_{j=1}^{\lfloor\frac{n}{i}\rfloor}\varphi(j)=\sum_{i=1}^{n}f(\lfloor\frac{n}{i}\rfloor)$$

$$\sum_{k=1}^{n}(\varphi*1)(k)=\sum_{k=1}^{n}Id(k)=\frac{n(n+1)}{2}$$

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

- 杜教筛的本质是求 $F=\sum f$，但 F 很难计算，于是尝试构造 g 使得 $f*g=h$，满足 G 和 H 很好计算

- 杜教筛的本质是求 $F=\sum f$，但 F 很难计算，于是尝试构造 g 使得 $f*g=h$，满足 G 和 H 很好计算
- 有等式 $\sum_{k=1}^{n}(f*g)(k)=\sum_{i=1}^{n}g(i)\sum_{j=1}^{\lfloor\frac{n}{i}\rfloor}f(j)=\sum_{i=1}^{n}g(i)F(\lfloor\frac{n}{i}\rfloor)$

- 杜教筛的本质是求 $F=\sum f$，但 F 很难计算，于是尝试构造 g 使得 $f*g=h$，满足 G 和 H 很好计算
- 有等式
  $\sum_{k=1}^{n}(f*g)(k)=\sum_{i=1}^{n}g(i)\sum_{j=1}^{\lfloor\frac{n}{i}\rfloor}f(j)=\sum_{i=1}^{n}g(i)F(\lfloor\frac{n}{i}\rfloor)$
- 又有 $\sum_{k=1}^{n}(f*g)(k)=\sum_{k=1}^{n}h(k)=H(n)$

- 杜教筛的本质是求 $F=\sum f$，但 F 很难计算，于是尝试构造 g 使得 $f*g=h$，满足 G 和 H 很好计算
- 有等式
  $\sum_{k=1}^{n}(f*g)(k)=\sum_{i=1}^{n}g(i)\sum_{j=1}^{\lfloor\frac{n}{i}\rfloor}f(j)=\sum_{i=1}^{n}g(i)F(\lfloor\frac{n}{i}\rfloor)$
- 又有 $\sum_{k=1}^{n}(f*g)(k)=\sum_{k=1}^{n}h(k)=H(n)$
- 移项有 $F(n)=H(n)-\sum_{i=2}^{n}F(\lfloor\frac{n}{i}\rfloor)$

1 预备知识

2 积性函数优化算法

3 杜教筛
   - 莫比乌斯函数前缀和
   - 欧拉函数前缀和
   - 变形

- 求 $f(k) = \mu(k) * k$，$F(n) = \sum_{i=1}^{n} f(i)$

- 求 $f(k)=\mu(k)*k$，$F(n)=\sum_{i=1}^{n} f(i)$
- $$(f*Id)(n)=\sum_{d|n} f(d)*Id(\frac{n}{d})=\sum_{d|n}\mu(d)*d*\frac{n}{d}=n*\sum_{d|n}\mu(d)=\epsilon(n)$$

- 求 $f(k)=\mu(k)*k$，$F(n)=\sum_{i=1}^{n}f(i)$
- $$(f*Id)(n)=\sum_{d|n}f(d)*Id(\frac{n}{d})=\sum_{d|n}\mu(d)*d*\frac{n}{d}=n*\sum_{d|n}\mu(d)=\epsilon(n)$$
- 即得到 $f*Id=\epsilon$，之后可以得到 $F(n)=1-\sum_{i=2}^{n}i*F(\lfloor\frac{n}{i}\rfloor)$

- 求 $\sum \varphi(i) * i$，$\sum \mu(i) * i^2$ 与上面类似

- 求 $\sum \varphi(i) * i$，$\sum \mu(i) * i^2$ 与上面类似
- 本质是对于 $f(k) = f_0(k) * h(k)$，其中 h 为完全积性函数

- 求 $\sum \varphi(i) * i$，$\sum \mu(i) * i^2$ 与上面类似
- 本质是对于 $f(k) = f_0(k) * h(k)$，其中 h 为完全积性函数
- 令 $g(k) = r(k) * h(k)$，那么 $(f * g)(k) = (f_0 * r)(k) * h(k)$，如果 G 和 $\sum(f * g)$ 很好计算，就可以类比之前的方法