狄利克雷卷积与莫比乌斯反演

posted on 2018-09-01 09:16:08 | under 数学 (.#type=数学) | 58

铃悬的数学小讲堂——狄利克雷卷积与莫比乌斯反演

在省选及以上的比赛中，我们逐渐会遇到越来越多的数学相关的题目（不是 $ab-a-b$ 那种小学奥数题啦）。

由此，便有了这篇 Blog 。// 博客背景图片是从 `ljh_2000` 的 cnblog 上摘下来的啦qwq

Warning ： 本文铺垫可能较长qwq 而且不太友好于公式恐惧症患者（公式恐惧症快点改啦 不然的话到省选难度会死掉的）

Warning 2：本文主要讲述数学知识，而具体实现较少（因为可以自行 Baidu `f(·ω·)f` ）

符号及规定

1. $[P]$ 是指，当 P 为真时，式子的值是 1 ；当 P 为假时，式子的值是 0 。// 可以理解成， `P` 是一个 0/1 布尔值， $[P]$ 就是 `(int)P` 。
2. $a\mid b$ 是指 b 被 a 整除，即存在一个整数 k 使得 $b=ka$ ； $n\perp m$ 是指 n 与 m 互质（注意， $1\perp 1$ 是成立的）。
3. 数论函数（见下）用小写粗体字母或普通希腊文字母（如 $\mathbf{f},\mathbf{g},\mathbf{h},\mu,\epsilon$ ）表示。// QAQ希腊文字母不能粗体

数论函数 & 狄利克雷卷积

数论函数是指这样一类函数：其定义域是正整数，值域是一个数集。

定义两个数论函数的加法，为逐项相加，即 $(\mathbf{f}+\mathbf{g})(n)=\mathbf{f}(n)+\mathbf{g}(n)$ ；

数乘（一个数乘到一个数论函数上），定义为这个数和每一项都相乘，即 $(x\mathbf{f})(n)=x\cdot\mathbf{f}(n)$。

// 这些都是很好理解哒qwq

接下来是**狄利克雷卷积。**

定义两个数论函数的狄利克雷卷积 $*$：

若 $\mathbf{t}=\mathbf{f}*\mathbf{g}$，则

$$\mathbf{t}(n)=\sum_{i|n}\mathbf{f}(i)\mathbf{g}\left(\frac{n}{i}\right)$$

或者等价地写，就是

$$\mathbf{t}(n)=\sum_{ij=n}\mathbf{f}(i)\mathbf{g}(j)$$

狄利克雷卷积有如下性质（ps：两个数论函数相等，是说它们每一项都相等）：

1. **交换律** $\mathbf{f}*\mathbf{g}=\mathbf{g}*\mathbf{f}$；// 这个结论还是很简单哒
2. **结合律** $(\mathbf{f}*\mathbf{g})*\mathbf{h}=\mathbf{f}*(\mathbf{g}*\mathbf{h})$。这是因为

$$\sum_{(i\cdot j)\cdot k=n}(\mathbf{f}(i)\mathbf{g}(j))\,\mathbf{h}(k)=\sum_{i\cdot(j\cdot k)=n}\mathbf{f}(i)\,(\mathbf{g}(j)\mathbf{h}(k))$$

   由此，我们以后不再区分 $(\mathbf{f}*\mathbf{g})*\mathbf{h}=\mathbf{f}*(\mathbf{g}*\mathbf{h})$，并将其写为 $\mathbf{f}*\mathbf{g}*\mathbf{h}$；

3. **分配律** $(\mathbf{f}+\mathbf{g})*\mathbf{h}=\mathbf{f}*\mathbf{h}+\mathbf{g}*\mathbf{h}$；~~// 留作习题 读者自证 (~~
4. **不知道叫什么** $(x\mathbf{f})*\mathbf{g}=x(\mathbf{f}*\mathbf{g})$ ~~// 同上~~
5. **单位元** $\epsilon*\mathbf{f}=\mathbf{f}$，其中 $\epsilon(n)=[n=1]=\begin{cases}1 & n=1\\0 & n>1\end{cases}$；~~//同上，这三个结论都可以直接套定义得到~~
6. **逆元** 对每个 $\mathbf{f}(1)\neq 0$ 的函数 $\mathbf{f}$，都存在一个函数 $\mathbf{g}$ 使得 $\mathbf{f}*\mathbf{g}=\epsilon$。

PS: ϵ 读作 epsilon 英 /ˈepsɪlɒn/ 美 /ˈepsɪlɑːn/

我们来着重讨论一下第 6 个结论。

如何求出一个函数的逆呢?

只需要定义：

$$\mathbf{g}(n)=\frac{1}{\mathbf{f}(1)}\left([n=1]-\sum_{i|n,i\neq 1}\mathbf{f}(i)\mathbf{g}\left(\frac{n}{i}\right)\right)$$

这样的话

$$\begin{aligned}&\sum_{i|n}\mathbf{f}(i)\mathbf{g}\left(\frac{n}{i}\right)\\=&\mathbf{f}(1)\mathbf{g}(n)+\sum_{i|n,i\neq 1}\mathbf{f}(i)\mathbf{g}\left(\frac{n}{i}\right)\\=&[n=1]\end{aligned}$$

最后一步直接将 $\mathbf{g}(n)$ 的定义代进去就好啦~。

但是单独有狄利克雷卷积还不够（它并没有帮助我们写代码）。

我们接下来要介绍：**积性函数** 。

积性函数

如果一个数论函数 $\mathbf{f}$ 满足：当 $n\perp m$ 时有

$$\mathbf{f}(nm)=\mathbf{f}(n)\mathbf{f}(m)$$

则称其为 **积性函数** 。

一些常见的积性函数有：

$\epsilon(n)=[n=1];\mathbf{id}(n)=n;\mathbf{id}^k(n)=n^k$

这些函数的积性很容易验证（实际上，它们满足完全积性，也就是说，无论 n,m 是否互质，都有 $\mathbf{f}(nm)=\mathbf{f}(n)\mathbf{f}(m)$ ）。

特殊的，我们令 $\mathbf{1}(n)=\mathbf{id}^0(n)=1$ 。

另外两个常见的积性函数是 σ_0 和 φ ，其中 $\sigma_0(n)$ 表示 n 的因数个数（ σ_k 表示所有因数的 k 次方和），而 $\varphi(n)$ 表示 $[1,n]$ 中与 n 互质的数的个数。

如何证明其积性呢?

如果 $n\perp m$ ，我们发现每个 nm 的约数 t 都可以分解成一个 n 的约数 $\gcd(n,t)$ 和一个 m 的约数 $\gcd(m,t)$ 的积，并且这种分解是一一对应的。

这是因为 $n\perp m,t\mid nm\Rightarrow t=\gcd(t,nm)=\gcd(t,n)\gcd(t,m)$ ，而且 $a\mid n,b\mid m\Rightarrow ab\mid nm$ ，并且 $a\mid n,b\perp n\Rightarrow\gcd(ab,n)=a$ 。

于是 $\sigma_0(nm)=\sigma_0(n)\sigma_0(m)$ 。

同样的， $t\perp nm\Leftrightarrow t\perp n,t\perp m\Leftrightarrow(t\bmod n)\perp n,(t\bmod m)\perp m$ ，所以每个 $[1,nm]$ 之间的与 nm 互质的数 t 都可以对应到一个 $[1,n]$ 的与 n 互质的数 $t\bmod n$ 和一个 $[1,m]$ 的与 m 互质的数 $t\bmod m$ 。

并且根据中国剩余定理，这种对应是一一对应的（即已知 $a \perp n, b \perp m$ 后可以唯一确定一个 $[1, nm]$ 之间的 t 使得 $t \bmod n = a, t \bmod m = b$，且 $t \perp nm$）。因此 $\varphi(nm) = \varphi(n)\varphi(m)$。

PS: σ sigma, 英&美 /ˈsɪgmə/； φ phi, 英&美 /faɪ/ 。

接下来我们来证明一个重要的结论：**两个积性函数的狄利克雷卷积是积性函数。**

如果 $\mathbf{t} = \mathbf{f} * \mathbf{g}$ 且 $\mathbf{f}, \mathbf{g}$ 都是积性函数，为什么 $\mathbf{t}$ 也是积性函数呢?

考虑到上面的性质，即"若 $n \perp m$ 则每个 nm 的约数都可以分解成一个 n 的约数和一个 m 的约数的积"；

并且有另一个性质：若 $n \perp m, a \mid n, b \mid m$ 则 $a \perp b$。（这个性质读者自~~(bai)~~证~~(du)~~，毕竟我要讲的是"莫比乌斯反演"而不是"整除和互质的性质"）

于是若 $n \perp m$，我们就有

$$\begin{aligned}
\mathbf{t}(nm) &= \sum_{d \mid nm} \mathbf{f}(d)\mathbf{g}\left(\frac{nm}{d}\right) \\
&= \sum_{a \mid n, b \mid m} \mathbf{f}(ab)\mathbf{g}\left(\frac{nm}{ab}\right) \\
&= \sum_{a \mid n, b \mid m} \mathbf{f}(a)\mathbf{f}(b)\mathbf{g}\left(\frac{n}{a}\right)\mathbf{g}\left(\frac{m}{b}\right) \\
&= \left(\sum_{a \mid n} \mathbf{f}(a)\mathbf{g}\left(\frac{n}{a}\right)\right)\left(\sum_{b \mid m} \mathbf{f}(b)\mathbf{g}\left(\frac{m}{b}\right)\right) \\
&= \mathbf{t}(n)\mathbf{t}(m)
\end{aligned}$$

第二个重要的结论：**积性函数的逆是积性函数。**

对于一个积性函数 $\mathbf{f}$，如何证明其逆 $\mathbf{g}(n) = [n = 1] - \sum_{i \mid n, i \neq 1} \mathbf{f}(i)\mathbf{g}\left(\frac{n}{i}\right)$（注意，积性函数一定满足 $\mathbf{f}(1) = 1$，因为 $\mathbf{f}(1) = \mathbf{f}(1)\mathbf{f}(1)$，并且如果 $\mathbf{f}(1) = 0$ 则 $\mathbf{f}(n) \equiv 0$，这种情况我们不考虑）也满足积性呢?

考虑数学归纳法（qwq 这一段 初学可以跳过 毕竟铃悬酱读的数学书比较多 超喜欢把所有结论证明出来）：

对 nm 的大小进行归纳：

1. $nm = 1$ 时，$\mathbf{g}(1) = 1$，结论显然成立；
2. 假设 $nm > 1$，当 $n'm' < nm$ 的时候有 $\mathbf{g}(n'm') = \mathbf{g}(n')\mathbf{g}(m')$，

$$
\begin{aligned}
\mathbf{g}(nm) &= -\sum_{d|nm,d\neq 1}\mathbf{f}(d)\mathbf{g}\left(\frac{nm}{d}\right)\\
&= -\sum_{a|n,b|m,ab\neq 1}\mathbf{f}(ab)\mathbf{g}\left(\frac{nm}{ab}\right)\\
&= -\sum_{a|n,b|m,ab\neq 1}\mathbf{f}(a)\mathbf{f}(b)\mathbf{g}\left(\frac{n}{a}\right)\mathbf{g}\left(\frac{m}{b}\right)\\
&= \mathbf{f}(1)\mathbf{f}(1)\mathbf{g}(n)\mathbf{g}(m) - \sum_{a|n,b|m}\mathbf{f}(a)\mathbf{f}(b)\mathbf{g}\left(\frac{n}{a}\right)\mathbf{g}\left(\frac{m}{b}\right)\\
&= \mathbf{g}(n)\mathbf{g}(m) - \left(\sum_{a|n}\mathbf{f}(a)\mathbf{g}\left(\frac{n}{a}\right)\right)\left(\sum_{b|m}\mathbf{f}(b)\mathbf{g}\left(\frac{m}{b}\right)\right)\\
&= \mathbf{g}(n)\mathbf{g}(m) - \epsilon(n)\epsilon(m)\\
&= \mathbf{g}(n)\mathbf{g}(m)
\end{aligned}
$$

注意前面几步把 $\mathbf{g}(\frac{nm}{ab})$ 拆成 $\mathbf{g}(n/a)\mathbf{g}(m/b)$ 的时候 $\frac{nm}{ab} < nm$，可以运用归纳条件。

最后一步是因为 $nm > 1$ 所以 n, m 之间至少一个不为 1，则 $\epsilon(n)\epsilon(m) = [n = 1][m = 1] = 0$。

那么问题来了，学了这么多东西，有什么用呢?

首先，积性函数可以进行线性筛。

由于每个正整数 n 都可以写成唯一的质因数分解形式 $n = \prod_{i=1}^{t} p_i^{k_i}$，并且不同的 $p_i^{k_i}$ 显然互质，所以

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

于是我们有另一种方法表示积性函数，即给出它在素数幂处的取值（这些值决定了整个积性函数）。

并且由于在线性筛素数的时候我们可以顺便求出每个数的最小质因数 p_1、最小质因数的次数 k_1 以及 $n/p_1^{k_1}$，就可以利用递推式 $\mathbf{f}(n) = \mathbf{f}(p_1^{k_1})\mathbf{f}(n/p_1^{k_1})$ 直接计算 $\mathbf{f}$ 了。

顺便，由于 σ_0 和 φ 在素数幂处的值很容易得到：$k > 0$ 时 $\sigma_0(p^k) = k + 1, \varphi(p^k) = p^{k-1}(p-1)$。

所以对于 $n = \prod_{i=1}^{t} p_i^{k_i}$，

$$
\sigma_0(n) = \prod_{i=1}^{t}(k_i + 1), \varphi(n) = \prod_{i=1}^{n} p_i^{k_i - 1}(p_i - 1) = n\prod_{i=1}^{t}\left(1 - \frac{1}{p_i}\right)
$$

因为 $\varphi * \mathbf{1}$ 也是积性函数并且我们很容易得到 $(\varphi * \mathbf{1})(p^k) = p^k$，所以我们得到：$\mathbf{id} = \varphi * \mathbf{1}$！

除去线性筛，还有一项秘技：

莫比乌斯反演

我们定义 $\mathbf{1}$ 的逆是 μ 。

这样的话，如果 $\mathbf{g} = \mathbf{f} * \mathbf{1}$ ，就有 $\mathbf{f} = \mathbf{f} * \mathbf{1} * \mu = \mathbf{g} * \mu$ 。

换句话说，如果 $\mathbf{g}(n) = \sum_{d|n} \mathbf{f}(d)$ ，就有 $\mathbf{f}(n) = \sum_{d|n} \mu\left(\frac{n}{d}\right) \mathbf{g}(d)$ 。

或者类似的，我们可以发现 $\mathbf{id}^k$ 的逆是 $\mathbf{t}(n) = \mu(n) n^k$ ，于是如果

$$\mathbf{g}(n) = \sum_{d|n} \left(\frac{n}{d}\right)^k \mathbf{f}(d)$$

就有

$$\mathbf{f}(n) = \sum_{d|n} \mu\left(\frac{n}{d}\right) \left(\frac{n}{d}\right)^k \mathbf{g}(d)$$

所以我们得到，比如， $\varphi = \mu * \mathbf{id}$ ，也就是

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

如何求 μ 呢?

由于 $\mathbf{1}$ 是积性的，而且 μ 是 $\mathbf{1}$ 的逆，所以 μ 也是积性的。简单的算术可以得出：

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

于是：

$$\mu(n) = \begin{cases} (-1)^t & n = p_1 p_2 \dots p_t \text{且} p_i \text{互不相同} \\ 0 & n \text{不满足上述条件（或者说 } n \text{ 有平方因子）} \end{cases}$$

我们轻而易举地（如果你不打算看上面的两个证明的话）得出了莫比乌斯反演的结论qwq。

另一个方向的莫比乌斯反演：

$$\mathbf{g}(x) = \sum_{x|y} \mathbf{f}(y) \Leftrightarrow \mathbf{f}(x) = \sum_{x|y} \mu\left(\frac{y}{x}\right) \mathbf{g}(y)$$

对此只需要定义 $(\mathbf{f} \oplus \mathbf{g})(x) = \sum_{x|y} \mathbf{f}(y/x)\mathbf{g}(y)$ ，并容易证明 $(\mathbf{f} * \mathbf{g}) \oplus \mathbf{h} = \mathbf{f} \oplus (\mathbf{g} \oplus \mathbf{h})$ 。于是

$$\mathbf{f} = (\mu * \mathbf{1}) \oplus \mathbf{f} = \mu \oplus (\mathbf{1} \oplus \mathbf{f}) = \mu \oplus \mathbf{g}$$

这样就证明了上述结论。

那么，下面我们就可以来看例题啦QwQ！

例题

P2257 YY 的 GCD

题意

给定 n, m，求 $\sum_{i=1}^{n} \sum_{j=1}^{m} [\gcd(i, j)是素数]$。

多组询问，每组 $n, m \leqslant 10000000$，数据组数 $T \leqslant 10000$。

解法

PS：本解法与本题 luogu 题解中多数题解方法不同（本质是相同的），建议参阅。

$\gcd(a, b)$ 有一个很好的性质：$d \mid \gcd(a, b) \Leftrightarrow d \mid a, d \mid b$。

于是对于一个关于 $\gcd(a, b)$ 的式子 $\mathbf{f}(\gcd(a, b))$，如果有 $\mathbf{f}(x) = \sum_{d|x} \mathbf{g}(d)$ 的话，就可以写成 $\sum_{d|a, d|b} \mathbf{g}(d)$ 了。

考虑令 $\mathbf{f}(x) = \begin{cases} 1 & x是素数 \\ 0 & x不是素数 \end{cases}$。我们要找到一个 $\mathbf{g}$ 使得 $\mathbf{f} = \mathbf{1} * \mathbf{g}$，那么 $\mathbf{g} = \mu * \mathbf{f}$，

$$\mathbf{g}(x) = \sum_{d|x} \mu\left(\frac{x}{d}\right) [d是素数] = \sum_{p|x} \mu\left(\frac{x}{p}\right)$$

其中 p 取素数。这个函数可以简单地这样求出（另外也可以线性筛，但比较繁琐，此处略去）：

```
for (int j = 0; j < prime_count; j++)
  for (int p = prime[j], i = 1; i * p <= N; i++)
    g[i * p] += mu[i];
```

（这段代码的复杂度是 $O\left(\sum_{p \leq n} (n/p)\right) = O(n \log \log n)$。具体证明需要一些微积分知识，此处略去）

我们要求的是

$$\begin{aligned}&\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{d\mid i,d\mid j}\mathbf{g}(d)\\=&\sum_{d=1}^{\min(n,m)}\mathbf{g}(d)\sum_{i=1}^{n}\sum_{j=1}^{m}[d\mid i][d\mid j]\\=&\sum_{d=1}^{\min(n,m)}\mathbf{g}(d)\left\lfloor\frac{n}{d}\right\rfloor\left\lfloor\frac{m}{d}\right\rfloor\end{aligned}$$

可以通过*整除分块*（也叫数论分块）在预处理 $\mathbf{g}$ 的前缀和之后单次 $O(\sqrt{n}+\sqrt{m})$ 的时间内求出，此处不再赘述。

简单题

题意

T 组询问，每次给定 n,m，求 $\sum_{i=1}^{n}\sum_{j=1}^{m}\gcd(i,j)$。

$T\leqslant 10000,n,m\leqslant 10^6$，答案对 998244353 取模。

解法

和上题相同，这次求的是 $\sum\mathbf{id}(\gcd)$。由于 $\mathbf{id}=\mathbf{1}*\varphi$，我们可以直接把它拆出来：

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{m}\gcd(i,j)&=\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{d\mid i,d\mid j}\varphi(d)\\&=\sum_{d=1}^{\min(n,m)}\varphi(d)\left\lfloor\frac{n}{d}\right\rfloor\left\lfloor\frac{m}{d}\right\rfloor\end{aligned}$$

直接预处理 φ 的前缀和，查询时数论分块就好啦qwq。（确实是简单题吧qwq

难题

题意

给出数列 $\mathbf{f}_1\dots\mathbf{f}_N$，

T 组询问，每次给定 n,m，求 $\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbf{f}_{\gcd(i,j)}$。

$T\leqslant 10000,n,m\leqslant N\leqslant 10^6$，答案对 998244353 取模。

解法

好吧这题还是搞笑用的（雾）。

为了找到一个 $\mathbf{f} = \mathbf{1} * \mathbf{g}$，我们只需要令 $\mathbf{g} = \mu * \mathbf{f}$ 即可。

如果求出了 $\mathbf{g}$，那么就和上题一样了，只不过把 φ 换成了 $\mathbf{g}$。

有三种求 $\mathbf{g}$ 的方法：

```
void get_g_1(int N, const int *f, int *g) {
  for (int i = 1; i <= N; i++) g[i] = 0;
  for (int i = 1; i <= N; i++)
    for (int j = 1; i * j <= N; j++)
      g[i * j] = (g[i * j] + mu[i] * f[j]) % mod;
} // 依照定义，O(nlogn)

void get_g_2(int N, const int *f, int *g) {
  for (int i = 1; i <= N; i++) g[i] = f[i];
  for (int i = 1; i <= N; i++)
    for (int j = 2; i * j <= N; j++)
      g[i * j] = (g[i * j] - g[i]) % mod;
} // 类似求狄利克雷卷积逆的方式，不需要线性筛 mu ，O(nlogn)

void get_g_3(int N, const int *f, int *g) {
  for (int i = 1; i <= N; i++) g[i] = f[i];
  for (int i = 0; i < prime_count; i++)
    for (int j = N / prime[i]; j >= 1; j--)
      g[j * prime[i]] = (g[j * prime[i]] - g[j]) % mod;
} // Magic! O(nloglogn)
```

~~第三种方式简单地解释方法是群直积的卷积变换可以分解成独立的卷积变换~~。

第三种方式，可以理解成 DP:

$$g_{i,n} = \sum_{d|n, d\text{只含前i种质因子}} \mu(n/d) f_d$$

具体转移就是

$$g_{i,n} = \begin{cases} g_{i-1,n} & p_i \nmid n \\ g_{i-1,n} - g_{i-1,n/p_i} & p_i \mid n \end{cases}$$

复杂度是 $O(\sum_p n/p) = O(n \log\log n)$ 的。

小结

本篇文章中讲述了狄利克雷卷积及莫比乌斯反演，并从一个完全数论的角度剖析了莫比乌斯反演。

希望这篇文章能让没有学过的同学们看的更明白，或者让学过的同学们从另一个角度来更深入的理解。

谢谢阅读！