写在前面

笔者在刷题过程中遇到一些求积性函数前缀和的问题，其中有一类问题需要在低于线性时间复杂度的算法，今天就来浅析一下这类问题的求解方法，当作以后讲课使用的讲义。若之后有了新的研究，再来继续完善这篇文章。

前置技能

积性函数的定义

1. 若$f(n)$的定义域为正整数域，值域为复数，即$f: Z^+ \rightarrow C$，则称$f(n)$为**数论函数。**
2. 若$f(n)$为数论函数，且$f(1)=1$，对于互质的正整数p, q有$f(p \cdot q) = f(p) \cdot f(q)$，则称其为**积性函数。**
3. 若$f(n)$为积性函数，且对于任意正整数p, q都有$f(p \cdot q) = f(p) \cdot f(q)$，则称其为**完全积性函数。**

积性函数的性质与例子

1. 常见的积性函数有
   - 除数函数$\sigma_k(n) = \sum_{d|n} d^k$，表示$n$的约数的$k$次幂和，注意$\sigma_k(n)$与$\sigma^k(n)$是不同的。
   - 约数个数函数$\tau(n) = \sigma_0(n) = \sum_{d|n} 1$，表示$n$的约数个数，一般也写为$d(n)$。
   - 约数和函数$\sigma(n) = \sigma_1(n) = \sum_{d|n} d$，表示$n$的约数之和。
   - 欧拉函数$\phi(n) = \sum_{i=1}^{n} [(n, i) = 1] \cdot 1$表示不大于$n$且与$n$互质的正整数个数，另外$\sum_{i=1}^{n} [(n, i) = 1] \cdot i = \frac{n \cdot \phi(n) + [n=1]}{2}$，且对于正整数$n > 2$来说$\phi(n)$是偶数。
   - 莫比乌斯函数$\mu(n)$，在狄利克雷卷积的乘法中与恒等函数互为逆元，$\mu(1) = 1$，对于无平方因子数$n = \prod_{i=1}^{t} p_i$有$\mu(n) = (-1)^t$，对于有平方因子数n有$\mu(n) = 0$。
   - 元函数$e(n) = [n = 1]$，狄利克雷卷积的乘法单位元，完全积性。
   - 恒等函数$I(n) = 1$，完全积性。
   - 单位函数$id(n) = n$，完全积性。

- 幂函数$id^k(n)=n^k$，完全积性。

2. 关于莫比乌斯函数和欧拉函数有两个经典的公式
   - $[n=1]=\sum_{d|n}\mu(d)$，将$\mu(d)$看作是容斥的系数即可证明。
   - $n=\sum_{d|n}\phi(d)$，将$\frac{i}{n}(1\leqslant i\leqslant n)$化为最简分数统计个数即可证明。
3. 若$f(n)$为积性函数，则对于正整数$n=\prod_{i=1}^t p_i^{k_i}$有$f(n)=\prod_{i=1}^t f(p_i^{k_i})$；若$f(n)$为完全积性函数，则对于正整数$n=\prod_{i=1}^t p_i^{k_i}$有$f(n)=\prod_{i=1}^t f(p_i)^{k_i}$。

狄利克雷卷积与莫比乌斯反演

1. 数论函数f和g狄利克雷卷积定义为$(f*g)(n)=\sum_{d|n}f(d)\cdot g(\frac{n}{d})$，狄利克雷卷积满足交换律、结合律，对加法满足分配律，存在单位元函数$e(n)=[n=1]$使得$f*e=f=e*f$，若f和g为积性函数则$f*g$也为积性函数。
2. 狄利克雷卷积的一个常用技巧是对于积性函数f与恒等函数I的卷积的处理，例如$n=\prod_{i=1}^t p_i^{k_i}, g(n)=\sum_{d|n}f(d)$，则有$g(n)=\prod_{i=1}^t\sum_{j=0}^{k_i}f(p_i^j)$。
3. 莫比乌斯反演也是对于$g(n)=\sum_{d|n}f(d)$的讨论，但是不要求f是积性函数，适用于已知$g(n)$求$f(n)$的情况，由于$I*\mu=e$，则$g*\mu=f*I*\mu=f*e=f$，即$f(n)=\sum_{d|n}g(d)\cdot\mu(\frac{n}{d})$，类似地有$g(n)=\sum_{n|d}f(d)\Rightarrow f(n)=\sum_{n|d}g(d)\cdot\mu(\frac{d}{n})$，二项式反演也是类似的技巧。有一个例子可以看出欧拉函数和莫比乌斯函数之间的关系，由于$\sum_{d|n}\phi(d)=id(n)$，所以$\phi(n)=\sum_{d|n}\mu(d)\frac{n}{d}$，也即$\frac{\phi(n)}{n}=\sum_{d|n}\frac{\mu(d)}{d}$。

正文：黑科技

这种黑科技大概起源于 `Project Euler` 这个网站，由 `xudyh` 引入中国的OI、ACM界，目前出现了一些OI模拟题、OJ月赛题、ACM赛题是需要这种技巧在低于线性时间的复杂度下解决一类积性函数的前缀和问题。

首先看一个简单的例子，求前n个正整数的约数之和，即$\sum_{i=1}^n\sigma(i)$，其中$n\leqslant 10^{12}$。
显然不能直接做了，但是我们可以推导一番：

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

当$i\leqslant\sqrt{n}$时，$\lfloor\frac{n}{i}\rfloor$显然只有$O(\sqrt{n})$个取值；当$i>\sqrt{n}$时，$\lfloor\frac{n}{i}\rfloor<\sqrt{n}$显然也只有$O(\sqrt{n})$个取值；对于固定的$\lfloor\frac{n}{i}\rfloor$，$i$的取值是一段连续的区间，当$\lfloor\frac{n}{i}\rfloor>1$时这段区间是$[\lfloor\frac{n}{\lfloor\frac{n}{i}\rfloor-1}\rfloor+1,\lfloor\frac{n}{\lfloor\frac{n}{i}\rfloor}\rfloor]$，因此可以$O(\sqrt{n})$计算所求。

同样地，求前n个正整数的约数个数之和也可以这样计算，留给读者练习。

另外需要说明的是，$\sum_{i=1}^{n}\lfloor\frac{n}{i}\rfloor\cdot i=\sum_{i=1}^{n}\frac{\lfloor\frac{n}{i}\rfloor\cdot(\lfloor\frac{n}{i}\rfloor+1)}{2}$，这也是一种常见的表示形式。

现在我们来加大一点难度，求前n个正整数的欧拉函数之和，即$\sum_{i=1}^{n}\varphi(i)$，其中$n\leqslant 10^{11}$。

目前本文提到的有关欧拉函数的公式只有几个，是否能用它们来帮助化简呢？答案是肯定的，接下来我们就利用$\sum_{d|n}\varphi(d)=n$来化简这个式子。

这个公式也可以看成是$\varphi(n)=n-\sum_{d|n,d<n}\varphi(d)$，设$\phi(n)=\sum_{i=1}^{n}\varphi(i)$，则有

$$\begin{aligned}\phi(n)&=\sum_{i=1}^{n}\varphi(i)=\sum_{i=1}^{n}i-\sum_{d|i,d<i}\varphi(d)=\frac{n\cdot(n+1)}{2}-\sum_{i=2}^{n}\sum_{d|i,d<i}\varphi(d)=\frac{n\cdot(n+1)}{2}-\sum_{\frac{i}{d}=2}^{n}\sum_{d=1}^{\lfloor\frac{n}{\frac{i}{d}}\rfloor}\varphi(d)\\&=\frac{n\cdot(n+1)}{2}-\sum_{i=2}^{n}\sum_{d=1}^{\lfloor\frac{n}{i}\rfloor}\varphi(d)=\frac{n\cdot(n+1)}{2}-\sum_{i=2}^{n}\phi(\lfloor\frac{n}{i}\rfloor)\end{aligned}$$

那么只要计算出$O(\sqrt{n})$个$\phi(\lfloor\frac{n}{i}\rfloor)$的值，就可以计算出$\phi(n)$，这样的复杂度又如何呢？

假设计算出$\phi(n)$的复杂度为$T(n)$，则有$T(n)=O(\sqrt{n})+\sum_{i=1}^{\sqrt{n}}T(i)+T(\frac{n}{i})$，**只展开一层就可以了，更深层的复杂度是高阶小量**，所以有$T(n)=\sum_{i=1}^{\sqrt{n}}O(\sqrt{i})+O(\sqrt{\frac{n}{i}})=O(n^{\frac{3}{4}})$。

由于$\phi(n)$是一个积性函数的前缀和，所以筛法也可以预处理一部分，假设预处理了前k个正整数的$\phi(n)$，且$k\geqslant\sqrt{n}$，则复杂度变为

$T(n)=\sum_{i=1}^{\frac{n}{k}}\sqrt{\frac{n}{i}}=O(\frac{n}{\sqrt{k}})$，当$k=O(n^{\frac{2}{3}})$时可以取到较好的复杂度$T(n)=O(n^{\frac{2}{3}})$。

之前利用$\phi(n)=n-\sum_{d|n,d<n}\phi(d)$的地方是怎么想到的呢？不妨来看一下这个

$$\frac{n\cdot(n+1)}{2}=\sum_{i=1}^{n}i=\sum_{i=1}^{n}\sum_{d}[d|i]\cdot\phi(d)=\sum_{\frac{i}{d}=1}^{n}\sum_{d=1}^{\left\lfloor\frac{n}{\frac{i}{d}}\right\rfloor}\phi(d)=\sum_{i=1}^{n}\phi(\lfloor\frac{n}{i}\rfloor)$$

如果能通过狄利克雷卷积构造一个更好计算前缀和的函数，且用于卷积的另一个函数也易计算，则可以简化计算过程。例如上题就是利用了$\phi*I=id$的性质，但一定注意，不是所有的这一类题都只用配个恒等函数I就可以轻松完事的，有时需要更细致的观察。

定义梅滕斯函数$M(n)=\sum_{i=1}^{n}\mu(i)$，给定正整数$n$，计算$M(n)$，其中$n\leqslant 10^{11}$。

有了欧拉函数的经验，这次似乎就轻车熟路了吧，使用$[n=1]=\sum_{d|n}\mu(d)$来试试？

$$1=\sum_{i=1}^{n}[i=1]=\sum_{i=1}^{n}\sum_{d|i}\mu(d)=\sum_{i=1}^{n}\sum_{d=1}^{\lfloor\frac{n}{i}\rfloor}\mu(d)=\sum_{i=1}^{n}M(\lfloor\frac{n}{i}\rfloor)$$

因此$M(n)=1-\sum_{i=2}^{n}M(\lfloor\frac{n}{i}\rfloor)$，问题可在$O(n^{\frac{2}{3}})$时间复杂度下解决。

看了上面的例子，是不是认为这种做法很 naive ，很好学啊，再来看一个题吧！

令$A(n)=\sum_{i=1}^{n}\frac{i}{(n,i)}$，$F(n)=\sum_{i=1}^{n}A(i)$，求$F(n)$模$(10^9+7)$的值，其中$n\leqslant 10^9$。

先做一番化简，变成积性函数前缀和的样子：

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

设$G(n)=2\cdot F(n)-n$，则

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

因此要求的是$\phi'(n)=\sum_{i=1}^{n}i\cdot\phi(i)$。
而对于$n=\prod_{i=1}^{t}p_i^{k_i}$，有

$$\sum_{d|n}d\cdot\phi(d)=\prod_{i=1}^{t}\sum_{j=0}^{k_i}p_i^j\cdot\phi(p_i^j)=\prod_{i=1}^{t}\frac{p_i^{2k_i+1}+1}{p_i+1}$$

这并不是什么好算前缀和的函数。
但是不难发现$(id\cdot\phi)*id=id^2$，即

$$\sum_{d|n}d\cdot\phi(d)\cdot\frac{n}{d}=n\cdot\sum_{d|n}\phi(d)=n^2$$

，这是一个很好计算前缀和的函数，于是有

$$\frac{n\cdot(n+1)\cdot(2n+1)}{6}=\sum_{i=1}^{n}i^2=\sum_{i=1}^{n}\sum_{d|i}d\cdot\phi(d)\cdot\frac{i}{d}=\sum_{\frac{i}{d}=1}^{n}\frac{i}{d}\cdot\sum_{d=1}^{\left\lfloor\frac{n}{\frac{i}{d}}\right\rfloor}d\cdot\phi(d)=\sum_{i=1}^{n}i\cdot\phi'(\lfloor\frac{n}{i}\rfloor)$$

因此$\phi'(n)=\frac{n\cdot(n+1)\cdot(2n+1)}{6}-\sum_{i=2}^{n}i\cdot\phi'(\lfloor\frac{n}{i}\rfloor)$，原问题可在预处理前$O(n^{\frac{2}{3}})$个值的基础上，在$O(n^{\frac{2}{3}}\log n)$的时间复杂度下解决。

但是注意到这种方法的常数与复杂度都可能较高，有时候可能再进行一些推导可以得到一个不使用正文方法的做法，例如ZOJ 3881 - From the ABC conjecture，网上存在一个解法是类似本文的方法，但是可以这样推导之后得到更简单的一个做法。

需要使用此种方法的题目一般数据规模较大，例如$n \leqslant 10^9, n \leqslant 10^{11}, n \leqslant 10^{12}$，但是并不是都要使用此类方法，有时候可能存在其他$O(\sqrt{n}), O(n^{\frac{2}{3}})$的做法，例如51Nod 1222 - 最小公倍数计数，会利用正文复杂度分析的方法即可，再例如ZOJ 5340 - The Sum of Unitary Totient，笔者不是很懂这题是否有其他做法，网上的一个解法是分段压缩打表，总之具体问题具体分析。

推荐题目

这里给出一些练手的题目供大家理解上述方法，这类题还是较少的，如有其他题的题源欢迎分享。

51Nod 1244 - 莫比乌斯函数之和

51Nod 1239 - 欧拉函数之和

BZOJ 3944 - Sum

HDU 5608 - function

51Nod 1238 - 最小公倍数之和 V3

51Nod 1237 - 最大公约数之和 V3

51Nod 1227 - 平均最小公倍数

Tsinsen A1231 - Crash的数字表格

51Nod 1222 - 最小公倍数计数（只利用正文的复杂度分析）

BZOJ 4176 - Lucas的数论

51Nod 1220 - 约数之和

ZOJ 3881 - From the ABC conjecture（可以不用正文的方法）

BZOJ 3512 - DZY Loves Math IV

ZOJ 5340 - The Sum of Unitary Totient（略诡异的一道题）