浅谈Min_25筛在积性函数上的应用

武汉市武钢三中

向路加

August 31, 2018

1 简介

Min_25筛是一种积性函数求和的筛法.
因为该算法在 Min_25 使用后才开始普及,因此也被称为 Min_25 筛法.
因为它可以解决一些杜教筛解决不了的东西，所以还是有学习的价值.
条件:

- $F(x), x \in Prime$可以用多项式表示.
- $F(x^p), x \in Prime$可以快速计算。

这个条件与洲阁筛相同，但实测比洲阁筛优秀。
有人说**洲阁筛应该可以退出积性函数求和的历史舞台了**，所以Min_25筛是一个相对更好的方法。

2 前提

先考虑这么一个问题,对于一个积性函数$F(x)$
要求低于线性复杂度计算

$$\sum_{i=2}^{n} F(x) \cdot [i \in Prime]$$

举例子,$F(i) = i^k$,
那么则求

$$\sum_{i=2}^{n} i^k \cdot [i \in Prime]$$

设预处理素数集合P,P中第j个素数是P_j
设$g(n,j)$表示在n以内,**是质数**或者**最小质因子大于**P_j**的合数**的F函数值之和

$$g(n,j) = \sum_{i=2}^{n} i^k \cdot [i \in Prime \quad or \quad Min_{t|i,t\in Prime} > P_j]$$

- 对于$P_j > n$,最小因子是P_j的数是P_j^2,它大于n,说明不会有新的贡献,那么

$$g(n,j) = g(n,j-1) \qquad (P_j^2 > n)$$

- 对于$P_j \le n$, 考虑从$g(n,j-1)$到$g(n,j)$的一个转移.

$$g(n,j) = g(n,j-1) - F(x) \cdot \sum_{x=2}^{n} [Min_{t|x,t\in Prime} t = P_j]$$

$g(n,j)$会在$g(n,j-1)$的基础上减去最小质因子刚好等于P_j的数对答案的贡献.

后面的$\sum_{x=2}^{n}[Min_{t|x,t\in Prime}t=P_j]$就是最小质因子等于$p_j$的数的个数.

可以把P_j提出来,那么要求的是剩下的数中最小质因数仍大于等于P_j的那些数的个数.

因为前面的$g(\frac{n}{p[j]})$会多计算小于$p[j]$的质数的贡献

答案是$g(\frac{n}{p[j]},P_j-1)$减去小于P_j的所有质数的贡献$g(P_j-1,j-1)$

总结一下,

$$g(n,j)=g(n,j-1)-F(P_j)\cdot[g(\frac{n}{p[j]},j-1)-g(P_j-1,j-1)]$$

如果$F(i)=i^k$
那么

$$g(n,j)=g(n,j-1)-P_j^k\cdot[g(\frac{n}{p[j]},j-1)-g(P_j-1,j-1)]$$

于是得到了

$$g(n,j)=\begin{cases}g(n,j-1) & P_j^2>n\\ g(n,j-1)-F(P_j)\cdot[g(\frac{n}{p[j]},j-1)-g(P_j-1,j-1)] & P_j^2\le n\end{cases}$$

实现细节

我们最后要求的东西就是$g(n,|Prime|)$,这里的$Prime$只需要用到$\sqrt{n}$.

初始条件

$$g(n,0)=\sum_{i=1}^{n}F(i)-1$$

$$g(P_j-1,j-1)=\sum_{i=1}^{j-1}F(P_j)$$

因为转移的时候，$g(?,j)$都是从$g(?,j-1)$转移过来的.

那么可以按照j作为层次转移,省去n那一维.

因为$\frac{n}{p_j}$最多只有$\sqrt{n}$种,那么转移的时候状态只有$\sqrt{n}$层.

时间复杂度的证明

素数定理:设$\pi(x)$为不超过x的素数个数,则$\pi(x) = O(\frac{x}{\ln x})$

总复杂度:

$$\sum_{i=1}^{\sqrt{n}} O(\pi(i)) + \sum_{i=1}^{\sqrt{n}} O(\pi(n/i))$$

$$= \sum_{i=1}^{\sqrt{n}} O(\frac{\sqrt{i}}{\ln \sqrt{i}}) + \sum_{i=1}^{n} O(\frac{\sqrt{\frac{n}{i}}}{\ln \sqrt{\frac{n}{i}}})$$

$$= O(\int_1^n \frac{\sqrt{\frac{n}{x}} dx}{\ln n})$$

$$= O(\frac{n^{\frac{3}{4}}}{\ln n})$$

3 前缀和

设$S(n,j)$表示所有最小质因子大于等于P_j的i的$F(i)$之和

$$S(n,j)=\sum_{i=1}^{n}F(i)\cdot[Min_{t|x,t\in Prime}t\geq P_j]$$

首先对于质数的S值可以算出
等于

$$g(n,|p|)-\sum_{i=1}^{j-1}F(P_i)$$

现在来考虑合数,可以枚举最小质因子为P_k,再枚举倍数e,满足$P_k^{e+1}\leq n$
合数的贡献可以由积性函数的性质得到
等于

$$\sum_{k\geq j}^{|P|}\sum_{e}^{P_k^{e+1}\leq n}S(\frac{n}{P_k^e},k+1)F(P_k^e)$$

这样会算漏$F({P_k}^e\cdot P_k)$的贡献,也就是$F(P_k^{e+1})$,所以把它加上.

得到

$$S(n,j)=g(n,|p|)-\sum_{i=1}^{j-1}F(P_i)+\sum_{k\geq j}^{|P|}\sum_{e}^{P_k^{e+1}\leq n}(S(\frac{n}{P_k^e},k+1)F(P_k^e)+F(p_k^{e+1}))$$

这个部分的复杂度是$\Theta(n^{1-\omega})$的

实现细节

可以通过递归实现或者非递归实现.

根据DoBelieve博客中所述,递归会快与非递归,因为会少遍历一些状态.

但是非递归可以求出$S(a_1,1),S(a_2,1)$....等状态

3.1 练习题

LOJ 6053 简单的函数

分析:

由于$f(p^c) = p \oplus c(p \in Prime)$,所以除了$f(2) = 3$,其他的$f(Prime_j) = Prime_j - 1$.但是$f(i) = i - 1$并不是一个积性函数,并不好求.

那么设$g(i) = i$,$h(i) = 1$,把这两个积性函数相减就是f

那么同样的方法

求出$g(n, |p|)$和$h(n, |p|)$,$g(n, |p|) - h(n, |p|) = \sum_{i=2}^{n} F(i)[i \in prime]$

得到$\sum F(i)$以后就可以通过Min_25的套路求出S了.

注意到$f(2) = 3$,所以要特判一下加上2.

犯过的错误:

1.线性筛打成maxn.

2.S函数递归的时候调用的是S(n/l,...),事实上应该是S(x/l,...).

3.在S函数中枚举质数的时候没判$pri[j] \times pri[j] \leq x$

4.调用dec, pls, mul函数的时候没有保证传入的参数在模意义下,结果直接爆long long了.

常数优化的方法:

1.减少取模次数

2.把int和long long换成unsigned long long

3.加一下register

实测可以从2000ms优化到900ms.

4 参考资料

- zzq's blog《一种筛法》
- yyb's blog《min_25筛》
- DoBelieve's blog《Min_25筛学习小记》
- 2018IOI国家集训队论文 朱震霆《一些特殊的数论函数求和问题》