浅谈算法竞赛中的数论函数

KeyID
复旦大学自学研究会

2018.08.03

为什么要讲数论?

- 为了回答这个问题，我们先来看看在近几年的区域赛中，选手们面对数论题的表现如何。

近年区域赛情况（节选）

年份	赛区	题号	AC的正式队伍数
2015	ACM-ICPC 长春	B	5
2016	ACM-ICPC 青岛	E	1
2016	CCPC 合肥	J	14
2016	CCPC 杭州	J	3
2017	CCPC 哈尔滨	I	0
2017	CCPC 杭州	L	0

数论题很难?

- 可以看出选手们整体上在数论题上表现并不理想
- 不是说这个camp是冲金难度的吗，怎么讲这么难的东西，我要睡觉了

事实恰恰相反

- 刚才列出数据的几个赛区，除了16青岛E考察的结论有些生僻，17杭州L确有一定难度外，其他题目均没有榜单上表现的那么难。究其原因，一是很多队伍见到数论就丧失信心、果断放弃；二是很多队伍对数论的研究确实太少。
- 如果能掌握一定的数论基础，比赛中有可能轻松比其他队伍多出一道题。
- 以15长春为例，5道题手速偏快就可以拿金牌。该场比赛约有220支队伍参赛，有3道题AC数在200左右，另有1道题AC数在150以上。也就是说，快速写完签到题和数论题就可以拿金牌。而这道数论题只需要知道积性函数的一些基本性质就可以做出来。

今日的内容

- 今天的主要内容是数论函数，因为这是数论中基础、常见的一个考察方向。在之前举的6场区域赛的例子中，有3场的数论题都是考察的数论函数。
- 数论还包含其他的一些考察点，有志于成为队伍数论担当的选手还需要补充此课件之外的知识。

从一道dp题谈起

- NWERC2015 Debugging
- 有一份包含一个bug的n($1 \le n \le 10^6$)行代码，运行一次到崩溃需要的时间为r($1 \le r \le 10^9$)。
- 你可以任意行添加printf语句来输出调试，即你知道是否在执行printf语句前就崩溃了。每设置一个printf语句需要花费p($1 \le p \le 10^9$)时间，但是运行时不额外消耗时间。
- 问在最坏情况下，最少需要多少时间可以定位BUG在哪一行。

解法

- 设f(n)表示n行代码debug需要的最少时间。显然f是个不降的序列。
- 假如我们选择第一次运行时添加x行printf语句，则最少花费 $f\left(\left\lceil\frac{n}{x+1}\right\rceil\right)+r+x\times p$的时间。
- 枚举x，有dp转移式$f(n)=\min\limits_{1\le i<n} f\left(\left\lceil\frac{n}{i+1}\right\rceil\right)+r+i\times p$
- 显然当$\left\lceil\frac{n}{i+1}\right\rceil$相等时，只需要考虑最小的i
- $\left\lceil\frac{n}{i}\right\rceil$只有$O(\sqrt{n})$种取值
- 记忆化搜索+暴力转移dp

- 因为$\left\lceil\frac{n}{ab}\right\rceil = \left\lceil\frac{\left\lceil\frac{n}{a}\right\rceil}{b}\right\rceil$，所以记忆化搜索时遍历到的所有值都等于某个$\left\lceil\frac{n}{i}\right\rceil$
- 之前提到了$\left\lceil\frac{n}{i}\right\rceil$只有$O(\sqrt{n})$种取值，那么就意味着我们遍历到了$O(\sqrt{n})$种值，每种值我们又用了$O(\sqrt{n})$的时间去枚举转移，总时间复杂度是$O(n)$的。
- 可以有更好的分析，时间复杂度为$O\left(\sum_{i=1}^{\sqrt{n}}\sqrt{i}+\sqrt{\lceil\frac{n}{i}\rceil}\right) = O\left(\int_1^{\sqrt{n}}\sqrt{\frac{n}{i}}\,\mathrm{d}i\right) = O(n^{\frac{3}{4}})$。

一些定义和性质

- $\left\lceil\frac{n}{ab}\right\rceil = \left\lceil\frac{\left\lceil\frac{n}{a}\right\rceil}{b}\right\rceil$, $\left\lfloor\frac{n}{ab}\right\rfloor = \left\lfloor\frac{\left\lfloor\frac{n}{a}\right\rfloor}{b}\right\rfloor$
- $\left\lceil\frac{n}{i}\right\rceil$只有$O(\sqrt{n})$种取值
- **数论函数**：定义域为正整数，陪域为复数的函数。今天我们主要研究定义域为正整数，值域为整数的函数。
- **积性函数**：满足“若a,b互质，则f(ab)=f(a)f(b)”的数论函数称为积性函数。
- **完全积性函数**：满足“f(ab)=f(a)f(b)”的数论函数称为完全积性函数。
- **狄利克雷卷积**：设f,g为两个数论函数，则满足$h(n) = \sum_{d|n} f(d)g(\frac{n}{d})$的函数称为f与g的狄利克雷卷积。也可以理解为$h(n) = \sum_{ij=n} f(i)g(j)$。
- 两个积性函数的狄利克雷卷积仍为积性函数。

一些定义和性质

- 狄利克雷卷积满足交换律和结合律
- 考虑使用$h(n) = \sum_{ij=n} f(i)g(j)$这个定义式，那么多个函数的狄利克雷卷积实际上就是$h(n) = \sum_{i_1 i_2 \cdots i_k = n} f_1(i_1) f_2(i_2) \cdots f_k(i_k)$，由乘法的交换律和结合律可以直接得到狄利克雷卷积的交换律和结合律。

一些常见的积性函数

- 单位函数$\mathrm{e}(x) = \begin{cases} 1, & x = 1 \\ 0, & x > 1 \end{cases}$
- 常函数$I(x) = 1$
- 幂函数$id(x) = x^k$
- 欧拉函数$\varphi(x) = 1$~x中与x互质的数的个数=$x \prod_{p|x,p\ is\ a\ prime} 1 - \frac{1}{p}$
- 莫比乌斯函数$\mu(x) = \begin{cases} 1, x = 1 \\ (-1)^k, x = p_1 p_2 \cdots p_k \\ 0, x = others \end{cases}$

积性函数性质例题1

- 定义f(n)=选两个[0,n)的整数a,b，且ab不是n的倍数的方案数。(2015 ICPC 长春 B)
- 求$g(n) = \sum_{d|n} f(d)$。
- 数据组数$1 \leq T \leq 20000$，$1 \leq n \leq 10$^9。

解法

- $f(n) = n^2 - \sum_{d|n} \varphi\left(\frac{n}{d}\right) * d$
- 设$h(n) = \sum_{d|n} \varphi\left(\frac{n}{d}\right) * d$，则$f(n) = n^2 - h(n)$。
- $g(n) = \sum_{d|n} f(d) = \sum_{d|n} d^2 - \sum_{d|n} \mathrm{h}(d)$。
- 设$P(n) = \sum_{d|n} d^2$，$Q(n) = \sum_{d|n} \mathrm{h}(d)$。
- 根据积性函数的性质，我们只需要计算$P(p^k)$和$Q(p^k)$，乘起来就可以得到$P(n)$ 和$Q(n)$。而这是很容易计算的，因为p^k的因数只有$p^0, p^1, \cdots, p$^k。
- 只剩下了质因数分解的时间复杂度。

一个相对简化的解法

- 设$id(n) = n$，则有$I * \varphi = id$，即$\sum_{d|n} \varphi(d) = n$
- 证明：$1 \le i \le n$且$\gcd(n, i) = d$的i有$\varphi(\frac{n}{d})$个，枚举1~n中每个数与n的最大公因数，则有$\sum_{d|n} \varphi(\frac{n}{d}) = \sum_{d|n} \varphi(d) = n$
- $Q(n) = \sum_{d|n} h(d) = \sum_{d|n} \sum_{w|d} \varphi(w) \times \frac{d}{w} = I * \varphi * id = (I * \varphi) * id = id * id = \sum_{d|n} d \times \frac{n}{d} = n \times n$的因数个数

积性函数性质例题2

- 定义$f_0(n) =$ 满足$pq = n$且$\gcd(p, q) = 1$的二元组(p, q)个数
- 定义$f_{r+1}(n) = \sum_{uv=n} \frac{f_r(u)+f_r(v)}{2}$
- 若干组询问，每次给出r, n，询问$f_r(n)$的值，对$10^9 + 7$取模。
- 询问次数$1 \le q \le 10^6$
- $0 \le r \le 10^6, 1 \le n \le 10^6$

解法

- 定义$\omega(n) = n$的不同质因子个数，则$f_0(n) = 2^{\omega(n)}$，由积性函数的定义可验证f_0为积性函数。
- 若f_r为积性函数，则由$f_{r+1}(n) = \sum_{d|n} f_r(d)$得$f_{r+1}$也为积性函数。
- 由积性函数性质，我们只需要求$f_r\left(p^k\right)$，由于$\forall p, f_0(p) = 2$，所以对于固定的k, r，$\forall p, f_r(p^k)$相等。
- 又注意到k是$O(\log n)$的，前缀和优化求$f_{r+1}\left(p^k\right) = \sum_{i=0}^{k} f_r(p^i)$，使用$O(r \log n)$的时间预处理出所有可能的$f_r(p^k)$的询问。
- 每组询问只剩下了质因数分解的复杂度。这部分也可以使用筛法预处理。

质因数分解

- 最开始一次预处理，找出不大于$10^{4.5}$的所有质数。因为$10^{4.5}$并不大，所以这题中很多质数筛都可以使用。
- 对于每组数据，检验$1\sim\sqrt{n}$中的每个质数是否是n的因数。
- 由质数定理得1~n中有$O(\frac{n}{\log n})$ 个质数，所以上述分解算法的时间复杂度为$O(\frac{\sqrt{n}}{\log n})$。

欧拉筛选法求质数

- 思路：令每个合数被最小的质因子筛去，且仅被筛去这一次。
- 实现方式：设合数$x = i \times p$，其中p是x的最小质因子。先枚举从小到大枚举i，再从小到大枚举质数p。若i是p的倍数，则更大的质数p_2不可能是$i \times p_2$的最小质因子，因为$i \times p_2$含有质因子p。
- 时间复杂度：$O(n)$

```
for (int i=2;i<=n;i++)
{
    if (!vis[i])
        prime[cnt++]=i;
    for (int j=0;;j++)
    {
        int x=i*prime[j];
        if (x>n)
            break;
        vis[x]=true;
        if (i%prime[j]==0)
            break;
    }
}
```

数论函数的前缀和

- 给出一个积性函数f，求$\sum_{i=1}^{n} f(i)$。
- 最简单的想法是分别求出来$f(1), f(2), \cdots, f(n)$，然后求和。显然这种思路的时间复杂度不可能低于$O(n)$。
- 如果$f(p)$可以在$O(\log n)$的时间内求出来，求出质数项的总时间是$O(n)$的；通常，$f(p^k)$可以比较容易的由$f(p^{k-1})$等值递推出来，其他项可以直接由积性函数的性质由$f(x) = f(d) * f(\frac{x}{d})$得到。因此，很多积性函数都可以在欧拉筛的过程中顺便递推出前n项的值，时间复杂度为$O(n)$。

低于线性时间对数论函数求和

- 2018 四川省赛GRISAIA
- 求$\sum_{i=1}^{n}\sum_{j=1}^{i} n \,\%\, (i \times j)$，$1 \le n \le 10^{11}$

解法1

- $\sum_{i=1}^{n}\sum_{j=1}^{i} n \,\%\, (i\times j)=\sum_{i=1}^{n}\sum_{j=1}^{i} n\ -\lfloor\frac{n}{ij}\rfloor ij$，现在只需要求$\sum_{i=1}^{n}\sum_{j=1}^{i}\lfloor\frac{n}{ij}\rfloor ij$。
- $\sum_{i=1}^{n}\sum_{j=1}^{i}\lfloor\frac{n}{ij}\rfloor ij=\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left\lfloor\frac{n}{ij}\right\rfloor ij+\sum_{i=1}^{n}\left\lfloor\frac{n}{i^2}\right\rfloor i^2}{2}$，重点是如何计算$\sum_{i=1}^{n}\sum_{j=1}^{n}\left\lfloor\frac{n}{ij}\right\rfloor ij$（以后简记为$ans$）。
- 定义$f(n)=\sum_{i=1}^{n}\lfloor\frac{n}{i}\rfloor i$，则$ans=\sum_{i=1}^{n}\sum_{j=1}^{n} i\times\left\lfloor\frac{\left\lfloor\frac{n}{i}\right\rfloor}{j}\right\rfloor\times j=\sum_{i=1}^{n} i\sum_{j=1}^{n}\left\lfloor\frac{\left\lfloor\frac{n}{i}\right\rfloor}{j}\right\rfloor\times j=\sum_{i=1}^{n} i\times f(\left\lfloor\frac{n}{i}\right\rfloor)$

解法1

- 定义$f(n)=\sum_{i=1}^{n}\lfloor\frac{n}{i}\rfloor i$，则$ans=\sum_{i=1}^{n}\sum_{j=1}^{n}i\times\left\lfloor\frac{\lfloor\frac{n}{i}\rfloor}{j}\right\rfloor\times j=\sum_{i=1}^{n}i(\sum_{j=1}^{n}\left\lfloor\frac{\lfloor\frac{n}{i}\rfloor}{j}\right\rfloor\times j)=\sum_{i=1}^{n}i\times f(\left\lfloor\frac{n}{i}\right\rfloor)$。
- $O(\sqrt{n})$地枚举$\lfloor\frac{n}{i}\rfloor$的值，然后$O(\sqrt{\left\lfloor\frac{n}{i}\right\rfloor})$地计算$f(\left\lfloor\frac{n}{i}\right\rfloor)$，时间复杂度为$O\left(\sum_{i=1}^{\sqrt{n}}\sqrt{\lfloor\frac{n}{i}\rfloor}\right)=O(n^{\frac{3}{4}})$。
- $O(1)$读写f的技巧：使用两个$\sqrt{n}$大小的数组，一个存储$f(1)$~$f(\sqrt{n})$，一个存储$f(\frac{n}{1})$~$f(\frac{n}{\sqrt{n}})$。

解法2

- 考虑优化解法1，定义$g(n) = f(n) - f(n-1)$
- $g(n) = \sum_{i=1}^{n} i \times (\left\lfloor\frac{n}{i}\right\rfloor - \left\lfloor\frac{n-1}{i}\right\rfloor) = \sum_{i|n} i$
- 使用欧拉筛求出g的前$n^{\frac{2}{3}}$项，再求一下前缀和，就可以得到f的前$n^{\frac{2}{3}}$项，这部分的时间复杂度是$O(n^{\frac{2}{3}})$。
- 现在只需要在$\left\lfloor\frac{n}{i}\right\rfloor > n^{\frac{2}{3}}$时暴力计算$f(\left\lfloor\frac{n}{i}\right\rfloor)$，时间复杂度为$O\left(\sum_{i=1}^{n^{\frac{1}{3}}} \sqrt{\left\lfloor\frac{n}{i}\right\rfloor}\right) = O(n^{\frac{2}{3}})$。

莫比乌斯反演

- $\forall n, f(n)=\sum_{d|n} g(d)$与$\forall n, g(n)=\sum_{d|n} \mu(d) f(\frac{n}{d})$等价。
- 即$f=g*I \Leftrightarrow g=\mu*f$。
- 证明：先证明$\mu*I=e$
- $\forall n>1, \sum_{d|n} \mu(d)=\sum_{i=0}^{k} C(k,i)\times(-1)^i=(1-1)^k=0$
- 证明$f=g*I \Rightarrow g=\mu*f$
- 利用狄利克雷卷积的交换律和结合律，由$f=g*I$得$\mu*f=g*(\mu*I)=g*e=g$
- $g=\mu*f \Rightarrow f=g*I$也可以类似地证明。

“莫比乌斯反演”例题

- 求$\sum_{i=1}^{n}\sum_{j=1}^{m}[\gcd(i,j)=质数]$，$T\le 10^4, n,m\le 10^7$（BZOJ2820)
- 解法：
- $ans=\sum_p\sum_{i=1}^{\lfloor\frac{n}{p}\rfloor}\sum_{j=1}^{\lfloor\frac{m}{p}\rfloor}[\gcd(i,j)=1]$
- 容斥原理：

$$|\bigcup_{i=1}^{n}A_i|=\sum_{i=1}^{n}(-1)^{i-1}\sum_{1\le j_1<j_2<\cdots<j_i\le n}|A_{j_1}\bigcap A_{j_2}\bigcap\cdots\bigcap A_{j_i}|$$

- $ans=\sum_p\sum_{g=1}^{\lfloor\frac{\min(n,m)}{p}\rfloor}\mu(g)\left\lfloor\frac{n}{pg}\right\rfloor\left\lfloor\frac{m}{pg}\right\rfloor=\sum_{i=1}^{\min(n,m)}\left\lfloor\frac{n}{i}\right\rfloor\left\lfloor\frac{m}{i}\right\rfloor\sum_{p|i}\mu(\frac{i}{p})$

“莫比乌斯反演”例题

- 定义$f(i)=\sum_{p|i}\mu\left(\frac{i}{p}\right)$，则$ans=\sum_{i=1}^{\min(n,m)}\left\lfloor\frac{n}{i}\right\rfloor\left\lfloor\frac{m}{i}\right\rfloor f(i)$，预处理出$f$就可以$O(\sqrt{n}+\sqrt{m})$地回答每组询问。
- 求f的一个方式是按照定义暴力求和，复杂度$O(n\log\log n)$
- 观察之后可以得到f的一个递推式，设x为n的最小质因子，若$x^2|n$，则$f(n)=\mu\left(\frac{n}{x}\right)$，否则$f(n)=\mu\left(\frac{n}{x}\right)+\mu(x)f\left(\frac{n}{x}\right)=\mu\left(\frac{n}{x}\right)-f\left(\frac{n}{x}\right)$，使用欧拉筛可以做到$O(n)$。

杜教筛

- 设$S(n)=\sum_{i=1}^{n} f(i)$，下面我们将求$S(n)$的过程一般化
- $\forall$数论函数g，设$h=f*g$，有$\sum_{i=1}^{n} h(i)=\sum_{1}^{n} g(i)S(\left\lfloor\frac{n}{i}\right\rfloor)$
- 移项，得$g(1)S(n)=\sum_{i=1}^{n} h(i)-\sum_{2}^{n} g(i)S(\left\lfloor\frac{n}{i}\right\rfloor)$
- 如果我们可以$O(\sqrt{n})$计算$\sum_{i=1}^{n} h(i)$，$O(1)$计算g的前缀和，就可以快速把问题递归为同类子问题，时间复杂度为$O\left(\sum_{i=1}^{\sqrt{n}} \sqrt{\left\lfloor\frac{n}{i}\right\rfloor}\right)\approx O(n^{\frac{3}{4}})$。
- 如果f有一些比较好的性质（如是积性函数），我们可以用欧拉筛求出前$n^{\frac{2}{3}}$项，更后面的项再递归，时间复杂度为$O(n^{\frac{2}{3}})$。

杜教筛例题1

- 求$\sum_{i=1}^{n}\varphi(i)$，$1 \le n \le 10^{11}$，答案取模
- 设$S(n)=\sum_{i=1}^{n}\varphi(i)$，$\mathrm{g(n)}=1$，$h=\varphi * g$
- $g(1)S(n)=\sum_{i=1}^{n}h(i)-\sum_{2}^{n}g(i)S\left(\left\lfloor\frac{n}{i}\right\rfloor\right)$
- 代入g的值，有$h(n)=\sum_{d|n}\varphi(d)=n$
- 代入g,h的值，得$S(n)=\sum_{i=1}^{n}i-\sum_{2}^{n}S\left(\left\lfloor\frac{n}{i}\right\rfloor\right)$
- 用欧拉筛求出欧拉函数的前$n^{\frac{2}{3}}$项，求前缀和得到S的前$n^{\frac{2}{3}}$项。$\left\lfloor\frac{n}{i}\right\rfloor$较大时递归计算，总时间复杂度$O(n^{\frac{2}{3}})$。

杜教筛例题2

- 已知函数f满足$\sum_{d|n} f(d) = n^2 - 3n + 2$，求$\sum_{i=1}^{n} f(i)$，答案取模。$T \le 500, n \le 10^9$，只有5组数据中$n > 10^6$（HDOJ 5608）
- $f(n) = n^2 - 3n + 2 - \sum_{d|n\ and\ d \ne n} f(d)$，从小到大递推$f$的前$n^{\frac{2}{3}}$项，时间复杂度$O(n^{\frac{2}{3}} \log n)$。
- 题设条件实际是是直接告诉了f与$g(n) = n$的狄利克雷卷积，因此剩下的部分直接套用杜教筛即可。唯一的问题可能出在如何快速计算$\sum_{i=1}^{n} i^2 - 3i + 2$，用公式分别求$\sum_{i=1}^{n} i^2$，$\sum_{i=1}^{n} 3i$，$\sum_{i=1}^{n} 2$，即可$O(1)$计算。

Tips

- 杜教筛用到的等式对所有数论函数都成立，只是通常当f为积性函数时，我们才能方便地用欧拉筛求值，从而实现$O(n^{\frac{2}{3}})$的时间复杂度。但正如例题2，有时大于$O(n^{\frac{2}{3}})$的时间复杂度我们仍然可以接受，因此杜教筛仍然可以使用。
- 杜教筛实际上求出来了所有的$S\left(\left\lfloor\frac{n}{i}\right\rfloor\right)$，我们需要的$g$的前缀和也正是$S_g\left(\left\lfloor\frac{n}{i}\right\rfloor\right)$这些项，因此有需要时可以先对$g$使用一次杜教筛以便于快速计算$g$的前缀和。
- 不是所有的函数f（即使它是积性的）都能找到合适的函数g满足g与$f * g$的前缀和都能很快地计算，这种情况下杜教筛的时间复杂度可能非常大，不适合使用杜教筛。

Extended Eratosthenes Sieve

- 给出一个积性函数f，且$f(p)$为关于p的多项式。求$S(n)=\sum_{i=1}^{n} f(i)$。
- $\forall\ 2 \le i \le n$，我们把i分为两类，第一类是最大质因子的幂次=1，则其次大质因子$<\sqrt{n}$；第二类是最大质因子的幂次>1，则其最大质因子$\le\sqrt{n}$。
- 枚举所有质因子$\le\sqrt{n}$的数k，设其最大质因子为L，则$S(n) \mathrel{+}= f(k)\times\sum_{L<p\le\frac{n}{k}} f(p)$，此时每个$k*p$都对应一个第一类数。如果$k$的最大质因子次幂$>1$，$S(n) \mathrel{+}= f(k)$，此时$k$就是一个第二类数。
- 如果我们dfs质因子来得到每个k，通常可以由积性函数的定义与性质简单地计算出$f(k)$。另外需要注意的是，如果$L>\frac{n}{k}$，则继续递归对答案的贡献必然为0，此时需要及时break，否则会影响复杂度。

Extended Eratosthenes Sieve

- 假设我们可以$O(1)$地求出$\sum_{L<p\le\frac{n}{k}}f(p)$，那么上面过程的时间复杂度就是$O$(满足$k\times L\le n$的$k$的个数)，当$n\le 10^{13}$时，时间复杂度为$O(\frac{n^{\frac{3}{4}}}{\log n})$。
- 设$g(i)=\sum_{1\le p\le i}f(p)$，现在问题只剩下了求$\sum_{L<p\le\frac{n}{k}}f(p)=g\left(\lfloor\frac{n}{k}\rfloor\right)-g(L)$。$\lfloor\frac{n}{k}\rfloor$只有$O(\sqrt{n})$种，$L\le\sqrt{n}$也只有$O(\sqrt{n})$种，因此我们只需要计算$g$的$O(\sqrt{n})$项。

Extended Eratosthenes Sieve

- 在题设里提到了$f(p)$是一个关于p的多项式，即$f(p)=\sum a_i p^{k_i}$，我们对于每个i，假设$f(p)=p^{k_i}$，最后乘上系数累加就可以得到真正的$f(p)$。
- 我们现在需要求的是$g(i)=\sum_{1\le p\le i} f(p)$，因此$f$的非质数项的结果是不影响答案的，我们强行规定$f(n)=n^{k_i}$使得$f$成为一个完全积性函数。

Extended Eratosthenes Sieve

```
for i:
    g[i] = \sum_{j=2}^i {j^k}

for p <= \sqrt{n}:
    for i:
        if i < p*p:
            break
        g[i] -= (g[i/p] - g[p-1]) * p^k
```

Extended Eratosthenes Sieve

- 对于每个我们可能用到的$g(i)$，我们只会在遍历不超过$\sqrt{n}$的质数时访问到，因此每个i贡献的时间复杂度为$O\left(\frac{\sqrt{i}}{\log\sqrt{i}}\right) = O(\frac{\sqrt{i}}{\log i})$。
- 时间复杂度为$O\left(\sum_{i=1}^{\sqrt{n}} \frac{i}{\log i} + \sum_{i=1}^{\sqrt{n}} \frac{\sqrt{\frac{n}{i}}}{\log\sqrt{\frac{n}{i}}}\right) \approx O(\frac{n^{\frac{3}{4}}}{\log n})$。
- 虽然Extended Eratosthenes Sieve依赖于函数的积性，不过因为杜教筛能做到较低复杂度时，函数通常也是积性的，所以总体来说Extended Eratosthenes Sieve的适用范围更广。

EES例题

- 定义$d(n)=n$的因数个数。求$S(n)=\sum_{i=1}^{n} d(i^3)$。$n \le 10^{11}$。(SPOJ DIVCNT3)
- 考虑因数个数的计算方式：取每个质因子的幂次+1乘起来。因此$f(i)=d(i^3)$是一个积性函数，且$f(p)=4=4p^0$是一个关于p的多项式。
- 先强行假设$f(p)=p^0=1$，求出$\sum_{1 \le p \le i} f(p)$后乘4，得到真正的$\sum_{1 \le p \le i} f(p)$。一定要保证系数为1，不能是4的原因是我们要保证$f$的每一项是同一个多项式，且$f$是一个完全积性函数，如果这里我们直接令$f(p)=4$，则$f(p_1p_2)=16$，与$f(p_1p_2)=4$矛盾。
- 之后就按照EES的方法求值就可以了。

非套路线性求和方法

- 定义$\omega(n) = n$的质因子个数，$g(n) = 2^{\omega(n)}$，求$S(n) = \sum_{i=1}^{n} g(i)$，答案取模。$n \le 10^{12}$。（CCPC杭州2016 J）
- $g(n) = \sum_{d|n} \mu^2(d)$
- $S(n) = \sum_{i=1}^{n} \sum_{d|i} \mu^2(d) = \sum_{i=1}^{n} i \times \sum_{j=1}^{\lfloor \frac{n}{i} \rfloor} \mu^2(j)$
- 容斥得$\sum_{i=1}^{n} \mu^2(i) = \sum_{i=1}^{n} \left\lfloor \frac{n}{i^2} \right\rfloor \times \mu(i)$
- $S(n) = \sum_{i=1}^{n} i \times \sum_{j=1}^{\lfloor \frac{n}{i} \rfloor} \left\lfloor \frac{n}{ij^2} \right\rfloor \times \mu(j) = \sum_{i=1}^{n} \mu(i) \times \sum_{j=1}^{\lfloor \frac{n}{i^2} \rfloor} j \times \lfloor \frac{n}{i^2 j} \rfloor$

非套路线性求和方法

- $S(n)=\sum_{i=1}^{n} i\times\sum_{j=1}^{\lfloor\frac{n}{i}\rfloor}\left\lfloor\frac{n}{ij^2}\right\rfloor\times\mu(j)=\sum_{i=1}^{n}\mu(i)\times\sum_{j=1}^{\lfloor\frac{n}{i^2}\rfloor} j\times\lfloor\frac{n}{i^2 j}\rfloor$
- 设$f(n)=\sum_{i=1}^{n} i\times\lfloor\frac{n}{i}\rfloor$，则$f(\mathrm{n})$可以$O(\sqrt{n})$计算
- $S(n)=\sum_{i=1}^{n}\mu(i)\times f(\left\lfloor\frac{n}{i^2}\right\rfloor)=\sum_{i=1}^{\sqrt{n}}\mu(i)\times f(\left\lfloor\frac{n}{i^2}\right\rfloor)$
- 时间复杂度$O(\sum_{i=1}^{\sqrt{n}}\sqrt{\left\lfloor\frac{n}{i^2}\right\rfloor})=O(\sqrt{n}\log n)$

rng_58-clj等式

- $\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{c}d(ijk)=\sum_{\gcd(i,j)=\gcd(j,k)=\gcd(k,i)=1}\left\lfloor\frac{a}{i}\right\rfloor\left\lfloor\frac{b}{j}\right\rfloor\left\lfloor\frac{c}{k}\right\rfloor$
- 证明：设$f(a,b,c)=\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{c}d(ijk)$，$g(a,b,c)=\sum_{\gcd(i,j)=\gcd(j,k)=\gcd(k,i)=1}\left\lfloor\frac{a}{i}\right\rfloor\left\lfloor\frac{b}{j}\right\rfloor\left\lfloor\frac{c}{k}\right\rfloor$
- 由数学归纳法，只需要证$d(abc)=f(a,b,c)-f(a-1,b,c)-f(a,b-1,c)-f(a,b,c-1)+f(a-1,b-1,c)+f(a-1,b,c-1)+f(a,b-1,c-1)-f(a-1,b-1,c-1)=g(a,b,c)-g(a-1,b,c)-g(a,b-1,c)-g(a,b,c-1)+g(a-1,b-1,c)+g(a-1,b,c-1)+g(a,b-1,c-1)-g(a-1,b-1,c-1)=\sum_{\gcd(i,j)=\gcd(j,k)=\gcd(k,i)=1}[i|a][j|b][k|c]$

rng_58-clj等式

- $d(abc)=\sum_{\gcd(i,j)=\gcd(j,k)=\gcd(k,i)=1}[i|a][j|b][k|c]$
- 设x_p,y_p,z_p分别为质数p在a,b,c中的幂次
- 则$d(abc)=\prod_p x_p+y_p+z_p+1=\sum_{\gcd(i,j)=\gcd(j,k)=\gcd(k,i)=1}[i|a][j|b][k|c]$
- 由证明方法可知这个等式可扩展至任意维

rng_58-clj等式例题1

- 求$\sum_{i=1}^{a}\sum_{j=1}^{b} d(ij)$。$T,a,b \le 50000$。（SDOI2015 约数个数和）
- $\sum_{i=1}^{a}\sum_{j=1}^{b} d(ij) = \sum_{\gcd(i,j)=1} \left\lfloor\frac{a}{i}\right\rfloor \left\lfloor\frac{b}{j}\right\rfloor = \sum_{g=1}^{\min(n,m)} \mu(g) \sum_{i=1}^{n}\sum_{j=1}^{m} \left\lfloor\frac{n}{ig}\right\rfloor \left\lfloor\frac{m}{jg}\right\rfloor$
- 定义$f(n) = \sum_{i=1}^{n} \lfloor\frac{n}{i}\rfloor$
- $\sum_{i=1}^{a}\sum_{j=1}^{b} d(ij) = \sum_{g=1}^{\min(n,m)} \mu(g) f\left(\left\lfloor\frac{n}{g}\right\rfloor\right) f\left(\left\lfloor\frac{m}{g}\right\rfloor\right)$
- 预处理时间复杂度$O(n)$，每组时间复杂度$O(\sqrt{n}+\sqrt{m})$。

rng_58-clj等式例题2

- 求$\sum_{i=1}^{n}\sum_{j=1}^{n}d(ij)$。$n\le 10^9$。（BZOJ4176）
- 类似地推出$\sum_{i=1}^{n}\sum_{j=1}^{n}d(ij)=\sum_{g=1}^{n}\mu(g)\sum_{i=1}^{n}\sum_{j=1}^{n}\left\lfloor\frac{n}{ig}\right\rfloor\left\lfloor\frac{n}{jg}\right\rfloor=\sum_{g=1}^{n}\mu(g)f^2\left(\left\lfloor\frac{n}{g}\right\rfloor\right)$
- 注意到我们需要求的μ的前缀和总是$\lfloor\frac{n}{g}\rfloor$变化处，正好是杜教筛求$\mu$的前$n$项和时求出来的值。
- 时间复杂度$O(n^{\frac{2}{3}})$