数论与积性函数

fuboat

February 4, 2017

由于数论问题常常与计算某个式子的值有关，因此有必要说明一下相关知识.

求和 $\sum$

$$\sum_i f(i) + g(i) \Leftrightarrow \left(\sum_i f(i)\right) + \left(\sum_i g(i)\right)$$

$$\sum_i \sum_j f(i) g(j) \Leftrightarrow \sum_j \sum_i f(i) g(j) \Leftrightarrow \left(\sum_i f(i)\right) \times \left(\sum_i g(i)\right)$$

$$\sum_i f(i) \times k \Leftrightarrow k \times \sum_i f(i)$$

枚举约数转为枚举倍数

$$\sum_{i}^{N}\sum_{d|i}f(d)\Leftrightarrow\sum_{d}\lfloor\frac{N}{d}\rfloor f(d)$$

接下来是对各种基本积性函数的介绍.

如果 $f(x), g(x)$ 为积性函数, 则下列函数也有积性:

1. $h(x) = f(x^p)$
2. $h(x) = f^p(x)$
3. $h(x) = f(x)g(x)$
4. $h(x) = \sum_{i|x} f(i)g(\frac{x}{i})$

莫比乌斯函数 $\mu(x)$

定义:

$$\mu(x)=\prod_{i|x,i\text{为质数}}-1[i^2\nmid x]$$

性质:

$$[x=1]=\sum_{i|x}\mu(i)$$

运用:

$$\sum_i\sum_j[(i,j)=1]=\sum_i\sum_j\sum_{d|i,d|j}\mu(d)$$

欧拉函数 $\varphi(x)$

定义:

$$\varphi(x) = \sum_{i}^{x} [(i, x) = 1]$$

性质:

$$x = \sum_{i|x} \varphi(i)$$

运用:

$$\sum_{i} \sum_{j} (i, j) = \sum_{i} \sum_{j} \sum_{d|i, d|j} \varphi(d)$$

约数和 $\sigma(x)$

定义:

$$\sigma(x) = \sum_{i|x} i$$

性质:

$$\sigma(xy) = \sum_{i|x} \sum_{j|y} [(i, \frac{y}{j}) = 1] ij$$

例题 1

给定 N, 求 $\sum_{i=1}^{N} \sigma(i)$.

例题 1

根据定义，显然有:

$$
\begin{aligned}
\sum_{i=1}^{N} \sigma(i) &= \sum_{i=1}^{N} \sum_{d|i} d \\
&= \sum_{d=1}^{N} \sum_{i=1}^{\lfloor \frac{N}{d} \rfloor} d \\
&= \sum_{d=1}^{N} \lfloor \frac{N}{d} \rfloor d
\end{aligned}
$$

例题 2

给定 N, M, 求 $\sum_{i=1}^{N}\sum_{j=1}^{M}\sigma(ij)$.

例题 2

根据性质，显然有:

$$\begin{aligned}
\sum_{i=1}^{N}\sum_{j=1}^{M}\sigma(ij) &= \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{x|i}\sum_{y|j}[(x,\tfrac{j}{y})=1]xy \\
&= \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{x|i}\sum_{y|j}xy\sum_{d|x,d|\frac{j}{y}}\mu(d) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{d|i,d|j}\mu(d)\sum_{x|\frac{i}{d}}\sum_{y|\frac{j}{d}}xyd \\
&= \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{d|i,d|j}\mu(d)\sigma(\tfrac{i}{d})\sigma(\tfrac{j}{d})d \\
&= \sum_{d=1}^{N}\mu(d)d\left(\sum_{i=1}^{\lfloor\frac{N}{d}\rfloor}\sigma(i)\right)\left(\sum_{j=1}^{\lfloor\frac{M}{d}\rfloor}\sigma(j)\right)
\end{aligned}$$

约数个数 $d(x)$

定义:

$$d(x) = \sum_{i|x} 1$$

性质:

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

莫比乌斯反演

就两个式子:
如果 $F(x) = \sum_{i|x} f(x)$,
则有 $f(x) = \sum_{i|x} \mu(i) F(\frac{x}{i})$.

分块

与数论函数没什么关系的优化

如果当 d 在区间 $[l, r]$ 内的和可以快速求，那我们不妨当 $d = l$ 时一次性将 $d \in [l, r]$ 的值全部贡献进答案，然后令 $d = r + 1$，重复前述操作.

分块

与数论函数没什么关系的优化

举个例子:

$$\sum_{d=1}^{N} f(\lfloor \frac{N}{d} \rfloor) \times d$$

假设对于 $d \in [l, r]$, $\lfloor \frac{N}{d} \rfloor$ 全部相等, 则 $d \in [l, r]$ 对答案的贡献是很好算的.

因为这些 d 对答案的贡献之和可以写成 $f(\lfloor \frac{N}{d} \rfloor) \times K$, 其中 K 是一个可以快速求的值. 这里, $K = \sum_{d=l}^{r} d$.

而 $\lfloor \frac{N}{d} \rfloor$ 的值只有 $O(\sqrt{N})$ 种, 所以只要求 $O(\sqrt{N})$ 次, 比直接暴力少很多.

记忆化 / 预处理

当题目为多组询问时，我们可以通过存储或预处理会被重复计算的部分，达到空间换时间的效果.

记忆化 / 预处理

与数论函数没什么关系的优化

给定多个 N, 分别求出 $g(N)=\sum_{d=1}^{N}\sum_{i=1}^{\lfloor\frac{N}{d}\rfloor}f_d(i)$, 其中 $f_d(x)$ 可以 $O(1)$ 求出.

记忆化 / 预处理

与数论函数没什么关系的优化

考虑将 $N = 1, 2, .., \mathrm{MaxN}$ 时 $g(N)$ 的值全部预处理出来.

当 N 从 1 取到 MaxN 时, $\frac{N}{d}$ 一共只会变化 $\sum_{d=1}^{N} \lfloor \frac{N}{d} \rfloor$ 次.
由于 $\sum_{d=1}^{N} \lfloor \frac{N}{d} \rfloor \approx N \ln N$, 所以可以用 $O(N \ln N)$ 的时间和空间做到 $O(1)$ 询问.

记忆化 / 预处理

与数论函数没什么关系的优化

给定多组 N, M (假定 $N < M$), 分别求出
$g(N, M) = \sum_{d=1}^{N} f_d(\lfloor \frac{N}{d} \rfloor) g_d(\lfloor \frac{M}{d} \rfloor)$, 其中 $f_d(x), g_d(x)$ 可以 $O(1)$ 求出.

记忆化 / 预处理

与数论函数没什么关系的优化

如果直接预处理 $F(N, M)$, 复杂度将是 $\sum_{d=1}^{N} \lfloor \frac{N}{d} \rfloor^2$, 实在是不能承受.
考虑到当 d 较大时 $\lfloor \frac{N}{d} \rfloor$ 的取值会变少, 这时预处理就可以接受了.
取临界值 S, 当 $d < S$ 时暴力做, $d \geq S$ 时用预处理的答案分块做, 复杂度约为 $O(TS + \frac{N^2}{S})$, 其中 T 为数据组数.

线性筛

$O(n)$ 预处理 $1..n$ 的积性函数值

令 $x = p_1^{a_1} p_2^{a_2} .. p_k^{a_k}$, 则 $f(x) = f(p_1^{a_1}) \times f(\frac{x}{p_1^{a_1}})$, 其中 $p_{1..k}$ 均为质数.

我们用欧拉筛可以得到 $x > 1$ 时对应的 p_1 和 a_1, 而对于 $f(p^a)$ 可以快速计算的积性函数, 线性筛完全可以胜任.

线性筛

$O(n)$ 预处理 $1..n$ 的积性函数值

举例:

$$\sigma(p^a) = p^0 + p^1 + p^2 + .. + p^a$$
$$d(p^a) = a + 1$$
$$\mu(p^a) = -1[a = 1]$$
$$\varphi(p^a) = p^a \times \frac{p-1}{p}$$

例题 3

给定 n, 求 $S(n)=\sum_{i=1}^{n} d(i^2)$, 共 T 组数据.

$T \leq 10000, n \leq 10^{12}$. 时间限制 20s, 空间限制 1.5GB.

例题 3

记 $f(x)$ 为 x 不同质因子的个数, 显然有:

$$d(i^2) = \sum_{d|i} 2^{f(d)}$$

原式化为:

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

考虑怎么组合出 $2^{f(x)}$. 我们可以认为, $2^{f(x)}$ 是 x 的无平方因子的因数的个数, 即:

$$2^{f(x)} = \sum_{d|x} \mu^2(d)$$

原式化为:

例题 3

$$\begin{aligned}\sum_{i=1}^{n} 2^{f(i)} &= \sum_{i=1}^{n} \sum_{d|i} \mu^2(d) \\ &= \sum_{d=1}^{n} \mu^2(d) \lfloor \frac{n}{d} \rfloor\end{aligned}$$

这里可以分块做，但是要求出 $\mu^2(d)$ 的前缀和.

例题 3

由 $\mu(x)$ 的容斥含义可得:

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

所以可以暴力做.

由于 $d(i^2)$ 是积性函数, 因此 n 较小时还是可以线性筛出 $S(n)$ 的, 所以只要算 i 较大时 $d(i^2)$ 的和. 这样做可以将复杂度做到 $O(n^{\frac{2}{3}})$.

杜教筛

利用 Dirichlet 卷积低于线性地计算积性函数前缀和

对于一个积性函数 $f(x)$，记 $S(x)=\sum_{i=1}^{x} f(i)$. 如果可以找到一个合适的积性函数 $g(x)$ 进行卷积，显然有：

$$\sum_{i=1}^{n}\sum_{d|i} f(\frac{i}{d})g(d)=\sum_{d=1}^{n} g(d)\sum_{i=1}^{\lfloor \frac{n}{d}\rfloor} f(i)=\sum_{d=1}^{n} g(d)S(\lfloor\frac{n}{d}\rfloor)$$

记 $f*g=h$，则有：

$$g(1)S(n)=\sum_{i=1}^{n}\sum_{d|i} f(\frac{i}{d})g(d)-\sum_{d=2}^{n} g(d)S(\lfloor\frac{n}{d}\rfloor)=\sum_{i=1}^{n} h(i)-\sum_{d=2}^{n} g(d)S(\lfloor\frac{n}{d}\rfloor)$$

如果 $g(x), h(x)$ 的前缀和都能快速求出，$S(n)$ 也能快速求出. 利用前述技巧，递归 + 记忆化即可.

例题 4

给定 n, 求 $S(n)=\sum_{i=1}^{n}\mu(i)$.

$n\leq 10^{12}$.

例题 4

由于:

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

又有:

$$\mu*1=\epsilon$$

可得:

$$S(n)=1(1)S(n)=\sum_{d=1}^{n}\epsilon(d)-\sum_{d=2}^{n}1(d)S(\lfloor\frac{n}{d}\rfloor)=1-\sum_{d=2}^{n}S(\lfloor\frac{n}{d}\rfloor)$$

基于质因数分解的方法

利用质因数分解组合出积性函数前缀和的值

定义积性函数 $F(x)$, 对给定的 n 求出 $S(n)=\sum_{d=1}^{n} F(d)$.

不妨将 d 分为两类: 含有大于 $\sqrt{n}$ 的质因子的; 不含大于 $\sqrt{n}$ 的质因子的.
由于 $d \le n$, d 包含的大于 $\sqrt{n}$ 的质因子不会超过 1 个. 不妨对于这一类 d, 将 $F(d)$ 分离为两部分.

由此, 我们可以将所求的前缀和的式子转为:

$$\sum_{1 \le d \le n, d \text{ 不包含大于} \sqrt{n} \text{ 的质因子}} F(d) \left(1 + \sum_{x, \sqrt{n} < x \le \lfloor \frac{n}{d} \rfloor, x \text{ 为质数}} F(x) \right)$$

式子的正确性是显然的.

基于质因数分解的方法

利用质因数分解组合出积性函数前缀和的值

注意到 $\lfloor \frac{n}{d} \rfloor$ 的取值只有 $O(\sqrt{n})$ 种.
不妨枚举 $\lfloor \frac{n}{d} \rfloor$, 记为 y.

对于每一种 y, 我们只需求出 $\sum_{1\le d\le n, d \text{ 不包含大于}\sqrt{n}\text{ 的质因子}, \lfloor \frac{n}{d} \rfloor = y} F(d)$ 以及 $\left(1+\sum_{x,\sqrt{n}<x\le y, x\text{ 为质数}} F(x)\right)$ 即可.

基于质因数分解的方法

利用质因数分解组合出积性函数前缀和的值

~~很好求对不对?~~
~~好求个头...~~

首先, y 的上界是 n, 所以后一部分不能暴力求, 否则复杂度将会达到 $O(n)$ (还不如打暴力).

考虑到质数分布的密度约为 $O(\frac{1}{\ln n})$, 因此理论上, 后一部分的复杂度可以做到 $O(\frac{n}{\ln n})$.
不妨考虑筛去对应范围内的合数, 由于 $[1, n]$ 之间的合数最多只会有 1 个大于 $\sqrt{n}$ 的质因子, 因此 $[1, n]$ 的合数都能被 $[1, \sqrt{n}]$ 中的质数筛去.
这样复杂度就能保证了.

前一部分也可以通过枚举 $[1, \sqrt{n}]$ 内的质因子来组合出对应的和.

直接通过例题来理解上述步骤.

例题 5

给定 n, 求 $S(n) = \sum_{i=1}^{n} d(i^3)$, 共 T 组数据.

$T \leq 10000, n \leq 10^{11}$. 时间限制 20s, 空间限制 1.5GB.

例题 5

记 $f(x) = d(x^3)$.
当 x 为质数时，$f(x) = 4$.
当 x 是形如 p^a 的数时，$f(x) = 3a + 1$.

我们不妨记 $g_r(k, i)$ 为 $[1, i]$ 中与前 k 个质数互质的数的 r 次幂之和，$k > 0$ 则有：

$$g_r(k, i) = g_r(k, i) - p^r \times g_r(k - 1, \lfloor \frac{i}{p} \rfloor)$$

$k = 0$ 时套公式.
这里，对于每一个 y，只要求出 $g_0(t, y)$ 即可，其中第 t 个质数为满足小于等于 $\sqrt{n}$ 的最大的质数.

$$\left(1 + \sum_{x, \sqrt{n} < x \leq y, x \text{ 为质数}} F(x)\right) = 4(g_0(t, y) - 1^0) + 1$$

例题 5

记 $h(k,i)$ 为 $[1,i]$ 中没有超过 P_k 的质因子的数的 $f(x)$ 的函数值之和, 其中 P_k 表示第 k 个质数的值, 由于 $f(x)$ 为积性函数, 有:

$$h(k,i) = h(k-1,i) + \sum_c f(P_k^c) h(k-1, \lfloor \frac{n}{p_k^c} \rfloor)$$

显然有:

$$\left(\sum_{1 \le d \le n, d \text{ 不包含大于} \sqrt{n} \text{ 的质因子}, \lfloor \frac{n}{d} \rfloor = y} f(d) \right) = h(t, \lfloor \frac{n}{y} \rfloor) - h(t, \lfloor \frac{n}{y+1} \rfloor)$$

然后求的时候记忆化一下就好.