数论专题

石门中学 罗梓璋

目录

说明

$[exp]$表示如果中间的布尔表达式exp为真，则为1，否则为0。
如无特殊说明，任意字母表示正整数，其中p专门表示质数。
f(n)等表示数论函数，有说明时也可以直接写为f。
除法默认下取整。
枚举默认从1开始。

数论基本知识

数论函数

数论函数
定义域为正整数，陪域（≈值域）是复数的函数。
这里值域默认正整数。
有时也可以理解为多项式，自变量是项数，因变量是系数。

常见数论函数

- 元函数
  $e(n)=[n=1]$
  表示元函数e当且仅当n=1时为1，其他时候都为0。
  不要和自然对数的底数搞混了。（虽然机会不多）

- 恒等函数
  $I(n)=1$

  无论什么时候都是1。

- 单位函数
  $ID(n)=n$
  因变量和自变量一样。

- 幂函数
  $ID^k(n)=n^k$
  其中
  $ID^0(n)=I(n)=1$
  $ID^1(n)=ID(n)=n$

- 约数函数（又叫除数函数）

$$\sigma_k(n)=\sum_{d|n}d^k$$

  其中$\sigma_0(n)$就是约数个数函数。
  $\sigma_1(n)$就是约数和函数。

- 莫比乌斯函数
  这里写成$\mu(n)$。

- 欧拉函数
  这里写成$\phi(n)$表示n以内有$\phi(n)$个正整数和n互质。

积性函数

如果一个数论函数f(n)满足 $f(ab) = f(a) \cdot f(b)$ 其中a、b互质。
那么这个数论函数f(n)具有积性，称为积性函数f(n)。
如果一个函数是积性函数，我们只需要求出$f(p^k)$的公式，就可以用欧拉筛法简单地求出10^7以内的每一个值。

完全积性函数

如果一个积性函数f(n)去掉乘数互质的条件，即
$f(ab) = f(a) \cdot f(b)$其中a、b为任意正整数，则f(n)为完全积性函数。

狄利克雷卷积基本知识

符号

这里用$a \times b$表示a卷积b，$a \cdot b$表示a直接乘b。

常见数论函数的计算

- 狄利克雷卷积

$$(a \times b)(n) = \sum_{d|n} a(d)b(\frac{n}{d})$$

- 乘法

$$(a \cdot b)(n) = a(n)b(n)$$

- 加减法

$$(a \pm b)(n) = a(n) \pm b(n)$$

狄利克雷卷积的性质

这里的字母表示任意数论函数。

- 交换律$a \times b = b \times a$
- 结合律$(a \times b) \times c = a \times (b \times c)$
- 分配律$(a + b) \times c = a \times c + b \times c$
- 单位元是元函数$a \times e = a$，即任意数论函数卷积元函数结果不变。
- **特别注意：**$a \cdot e = e$，即任意数论函数直接乘元函数都是元函数。不要弄混了。
- 存在逆元 如果$a \times b = e$则a和b互为逆元。

常见卷积式

这里的字母表示任意数论函数。

- 莫比乌斯反演$f = g \times I \Leftrightarrow g = f \times \mu$
- 莫比乌斯函数的性质 $\mu \times I = e$ 即μ和I互为逆元。这个式子可以解释莫比乌斯反演。暴力反推就可以得到莫比乌斯函数的定义式。（证明可以用数学归纳法）
- 欧拉函数的性质 $\phi \times I = ID \Leftrightarrow ID \times \mu = \phi$
- 恒等函数的性质 $I \times I = \sigma_0 \Leftrightarrow \sigma_0 \times \mu = I$
- 幂函数的性质 $ID^k \times I = \sigma_k \Leftrightarrow \sigma_k \times \mu = ID^k$

特别性质

如果满足$c(n) = b(x) \cdot b(\frac{n}{x})$
那么我们能推出

$$\left[(a \cdot b^k) \times b^k\right](n) = \sum_{d|n} a(d) \cdot b^k(d) \cdot b^k(\frac{n}{d}) = \sum_{d|n} a(d)c^k(n) = \left[(a \times I) \cdot c^k\right](n)$$

更加常见的形式是：

当$g(n)$是完全积性函数，有$(f \cdot g^k) \times g^k = (f \times I) \cdot g^k$。

这里由于完全积性$g(n) = g(d) \cdot g(\frac{n}{d})$，所以满足上式。

左式展开来是：

$$\sum_{d|n} f(d) \cdot g^k(d) \cdot g^k(\frac{n}{d})$$

右式展开来是：

$$g^k(n) \sum_{d|n} f(d)$$

利用这个性质，我们可以化简公式，也可以构造出$f(n)$和$I(n)$的卷积。

如：因为$ID(d) \cdot ID(\frac{n}{d}) = ID(n)$

可以推出：

- $(f \cdot ID^k) \times ID^k = (f \times I) \cdot ID^k$
- $(\phi \cdot ID^k) \times ID^k = (\phi \times I) \cdot ID^k = ID \cdot ID^k = ID^{k+1}$
- $(\mu \cdot ID^k) \times ID^k = (\mu \times I) \cdot ID^k = e \cdot ID^k = e$
- $ID^k \times ID^k = (I \cdot ID^k) \times ID^k = (I \times I) \cdot ID^k = \sigma_0 \cdot ID^k$

上面的公式告诉我们，这个性质最大的用途是让构造特定的卷积，来化简公式。

数论公式推导基本方法

分配率

$$\sum_x k f(x) = k \sum_x f(x)$$

要仔细看，容易看错。

分离变量

$$\sum_{x=1}^{n} \left(f(x) \left(\sum_{y=1}^{m} g(y) \right) \right) = \left(\sum_{x=1}^{n} f(x) \right) \left(\sum_{y=1}^{m} g(y) \right)$$

然后两个括号之间完全无关，二维变一维。

更高维可以类推。

约数转倍数

如果枚举某一个变量的约数，如$\sum_{x \le n} \sum_{d|x} x$

我们可以直接枚举d，让x=dy。

$\sum_{d \le n} \sum_{y \le \frac{n}{d}} dy$

为了进一步推导的方便，熟练后一般用原变量代替倍数变量，即用x代替y，上式变为：

$\sum_{d \le n} \sum_{x \le \frac{n}{d}} dx$

改变变量枚举顺序的要点

在存在多层枚举时，即类似$\sum_{x \le n} \sum_{y \le m}$这样的情况，有时候我们需要将内外层枚举交换顺序，这时枚举范围和公式本身都有可能发生变化。

改变枚举顺序的要点就是：保持枚举结果不变。

原来的每一个变量组(x,y)被枚举了多少次，改变顺序后要枚举一样的次数。

常见改变顺序的情况如下：
情况一：枚举范围没有关系。如$\sum_{x\le n}\sum_{y\le m}$，y的范围和x无关，每一个(x,y)只被枚举一次，我们可以直接把x和y换过来变成$\sum_{y\le m}\sum_{x\le n}$。

情况二：内层是外层的约数。如$\sum_{x\le n}\sum_{y|x}$，我们可以反过来先枚举y，再枚举y的倍数x。这时每一个(x,y)仍然是只被枚举一次，
考虑y的范围，由于x从1到n，所以它的约数也是从1到n。我们用yx代替x得到：
$\sum_{y\le n}\sum_{x\le \frac{n}{y}} yx$

情况三：公式中存在xy。例如$\sum_{x\le n}\sum_{y\le n} f(xy)[xy\le n]$。这时有一种特别的换枚举顺序的方法。
我们设$t=xy$，然后考虑每一个t出现了多少次。x是t的约数，t每有一个约数就会出现一次。化为
$\sum_{t\le n} f(t)\sigma_0(t)$.

尽量让枚举下标连续

连续的枚举下标如$\sum_{x=1}^{x\le n}$，不连续的下标如$\sum_{d|n}$
前者的枚举可以用分块优化，后者不行。所以大部分内层枚举是外层的约数的情况都要转成内层是外层的倍数。

消除gcd、lcm的方法

情况1：下标中存在gcd。
$\sum_{x\le n}\sum_{y\le n}\sum_{d|gcd(x,y)}$
可以化为：
$\sum_{x\le n}\sum_{y\le n}\sum_{d|x,d|y}$
然后转变枚举顺序，枚举d的倍数x、y：
$\sum_{d\le n}\sum_{x\le \frac{n}{d}}\sum_{y\le \frac{n}{d}}$
后面式子中的x和y换为dx和dy。

情况2：公式中存在a和b的gcd，由于$(\phi\times I)(gcd(a,b))=ID(gcd(a,b))$，我们可以构造一个欧拉函数的卷积式，使得gcd转到下标。$\gcd(a,b)=\sum_{d|gcd(a,b)}\phi(d)=\sum_{d|a,d|b}\phi(d)$

情况3：公式中存在a和b的gcd，我们可以设d是a和b的约数，dx=a，dy=b，这时，
$gcd(a,b)=d[gcd(x,y)=1]$，本来枚举a和b，可以改为枚举d、a、b。然后再消$[gcd(x,y)=1]$，这样gcd就会转到下标。（方法见后）

情况4：有lcm出现。$lcm(a,b)=\frac{ab}{gcd(a,b)}$，转为gcd。

一般来说，出现gcd都用一个变量代替它就好了。

[bool]条件框的消除

如果公式中存在[bool]条件框，有两种处理方法：
1、不管它，暴力枚举。
2、想办法消除它。
大部分情况下，前者都是低效的，所以要尽量消除条件框。
(注意比较罕见的情况下，是消不掉的，不要思维僵化。)

常见消除方法：
1、利用容斥原理分类成符合条件的和不符合的。
如$\sum_x\sum_y x+[x+y=n]y$

这个条件中符合的很容易拆出来，显然每一个x只有一个y符合。
不符合的反而难拆，我们用容斥，减去多余的不符合的即可。
变为$\sum_x nx - \sum_x x + \sum_x (n-x)$

2、等效替代
用一个和条件框等效的式子替代它。
最常见的式子是这个：

$$[gcd(x,y)=1] = \sum_{d|gcd(x,y)} \mu(d)$$

即$[gcd(x,y)=1] = e(gcd(x,y)) = (\mu \times I)(gcd(x,y))$
一个式子要替代条件框，那么它的取值就只有0、1两种，常见的函数之中只有元函数e符合，所以常见的方法就是构造一个有元函数的式子，再转化。

3、直接枚举符合条件的。
常见的是有约数的情况，枚举倍数。

- 如$\sum_{x\le n}[a|x]x = \sum_{d\le \frac{n}{a}} ad$
  即枚举a的倍数。
- 如果有$[x|a]$存在要枚举x，可以直接$O(\sqrt{n})$枚举a的约数。
- 另一种情况是这样：$[gcd(x,y)=a]$，我们想办法把它转化为gcd=1。我们直接枚举a的倍数ax和ay，此时只要gcd(x,y)=1就可以保证gcd(ax,ay)=a了，即
  $\sum_{x\le n}\sum_{y\le n}[gcd(x,y)=a] = \sum_{x\le \frac{n}{a}}\sum_{y\le \frac{n}{a}}[gcd(x,y)=1]$

4、变形为熟悉的形式
如$[d|xy] = [\frac{d}{gcd(x,d)}|y]$，后者有gcd，我们可以用上面消gcd的方法来解决。
我们设$gcd(x,d)=m$，用倍数代替约数，用mx代替x，md代替d，这时条件变为$[d|y][gcd(x,d)=1]$，这两者我们都有标准方法解决。

用数列求和公式化简

基本的数列求和要过关。
$\sum_{i\le n} i = \frac{n(n+1)}{2}$
$\sum_{i\le n} i^2 = \frac{2n^3+3n^2+n}{6}$
设m次方和的公式为S(m)，推m次方和公式的方法如下：
$(n+1)^{m+1} - n^{m+1} = \sum_{i=0}^{m} C_{m+1}^{i} n^i$
将n=1..n加起来得到：
$(n+1)^{m+1} - n^{m+1} + n^{m+1} - (n-1)^{m+1} + \ldots - 1^{m+1} = \sum_{i=0}^{m} C_{m+1}^{i} S(i)$
化简
$(n+1)^{m+1} - 1 = \sum_{i=0}^{m} C_{m+1}^{i} S(i)$
把要求的S(m)取出来得到：
$S(m) = \frac{(n+1)^{m+1} - 1 - \sum_{i=0}^{m-1} C_{m+1}^{i} S(i)}{C_{m+1}^{m}}$
即只要求出S(1..m-1)就可以求出S(m)了。
实际上考试中如果你忘记了公式，用高斯消元直接求系数会比较实际一点。（S(m)是m+1次多项式。）

注意点

- 小心括号，求和式最好整个括号起来，防止和后面的弄混。
- 小心枚举范围，注意是从1开始还是从其他开始，连续的还是枚举约数，到n还是到m。
- 注意整除和普通除法的差别。$\lfloor \frac{a+b}{c} \rfloor \neq \lfloor \frac{a}{c} \rfloor + \lfloor \frac{b}{c} \rfloor$。

- 在纸上写的时候可以一次只推导一部分公式，将不变的部分先放在一边，只推导会变化的部分。（当然自己要记得住）

杜教筛（前缀和问题）

问题

求某数论函数a(n)的前缀和S(n)。

$$S(n)=\sum_{i=1}^{n}a(i)$$

思想

构造卷积形式，用容易求得前缀和来推难求得前缀和。

线性筛

可以处理10^7范围内的积性函数前缀和问题。
最少需要求出的式子：$f(p^k)$

杜教筛

找另外一个数论函数b(n)，把a(n)和b(n)卷积起来。
设$a\times b=c$
那么我们算出c(n)的前缀和：

$$\sum_{i=1}^{n}c(i)=\sum_{i=1}^{n}\sum_{d|i}a(d)b(\frac{i}{d})$$

右边的式子下标是非连续的，我们考虑把右边的式子换一个形式，把约数变为倍数，让a的下标连续。
考虑对于每一个b来说，它乘了哪些a。
考虑有哪些$a(x)b(y)$形式的(x,y)对。
i从1枚举到n，每一个i的每一对约数就有一对(x,y)对。
那么y的范围明显是1到n，对于每一个y，对应的x可以是从1到$\frac{n}{y}$。
在这个范围内，显然每一对(x,y)都满足$xy\leq n$
再考虑1到n的每个数的每一个约数是否都出现了，假如有一个数k的约数q_1和q_2没有出现，说明要么$q_1>n$，要么$q_1q_2>n$，这些显然不可能。
所以上面的枚举方式可以枚举出1到n每一个数的每一对约数。

这种枚举方式可以写为：

$$\sum_{i=1}^{n}c(i)=\sum_{y=1}^{n}\Big(b(y)\sum_{x=1}^{\frac{n}{y}}a(x)\Big)=\sum_{i=1}^{n}b(i)S(\frac{n}{i})$$

注意到i=1的时候，出现了S(n)，我们把S(n)单独提出来得到：

$$S(n)=\frac{\sum_{i=1}^{n}c(i)-\sum_{i=2}^{n}b(i)S(\frac{n}{i})}{b(1)}$$

这时我们得到了一个求S(n)的递归式。

时间复杂度

假设我们能O(1)求出$\sum c(i)$和任意的b(i)。

$\sum b(i)S(\frac{n}{i})$
这个式子用分块处理，这样一层的时间是$O(\sqrt{n})$
设时间复杂度为f(n)，则
$f(n)=O(\sqrt{n})+\sum_{i=2}^{n}f(\frac{n}{i})$

我们再把$n^{\frac{2}{3}}$以内的S(n)通过线性筛预处理出来。
这样$f(\frac{n}{i})$在估算时就只需要考虑大于$n^{\frac{2}{3}}$范围的。

这里补充一个知识，n先整除a再整除b，是n整除ab的结果。也就是说，n整除一个数的结果，再整除一个数还在这些结果之中。
引理：

$$\lfloor\frac{\lfloor\frac{n}{a}\rfloor}{b}\rfloor=\lfloor\frac{n}{ab}\rfloor$$

证明：
设$\lfloor\frac{n}{ab}\rfloor=k$，则$n=kab+c$，其中$c<ab$。
然后有$\lfloor\frac{n}{a}\rfloor=kb+\lfloor\frac{c}{a}\rfloor$，
$\lfloor\frac{\lfloor\frac{n}{a}\rfloor}{b}\rfloor=k+\lfloor\frac{\lfloor\frac{c}{a}\rfloor}{b}\rfloor$
由于$c<ab$，所以$\lfloor\frac{\lfloor\frac{c}{a}\rfloor}{b}\rfloor=0$
所以$\lfloor\frac{\lfloor\frac{n}{a}\rfloor}{b}\rfloor=k=\lfloor\frac{n}{ab}\rfloor$
（这个引理不要乱用，混进加法之后就失效了）

回到时间复杂度的问题，用哈希或者map之类的记忆化方法，让每一个计算过的S(n)不再重复计算。（实际操作中，哈希和map时间差别几乎不存在，建议用map程序简单）
我们注意n的取值，全部是最开始的n整除某个数之后的结果。
这时，所有要计算的S(n)只剩下$\sqrt{n}$个。

这个时间复杂度可以写成

$$\sum_{i=1}^{n^{\frac{1}{3}}}O(\sqrt{\frac{n}{i}})\approx O(\int_{0}^{n^{\frac{1}{3}}}\sqrt{\frac{n}{i}})=O(n^{\frac{2}{3}})$$

空间复杂度显然也是$O(n^{\frac{2}{3}})$
大概可以处理10^9左右的前缀和。

例子

求μ的前缀和

$\mu\times I=e$，e的前缀和就是1

$$S(n)=1-\sum_{i=2}^{n}S(\frac{n}{i})$$

求ϕ的前缀和

$\phi\times I=ID$，ID的前缀和是$\frac{n(n+1)}{2}$

$$S(n)=\frac{n(n+1)}{2}-\sum_{i=2}^{n}S(\frac{n}{i})$$

求σ_k的前缀和

$\sigma_k \times \mu = ID^k$，$ID^k$的前缀和可以用数列推出来。

$$S(n) = \sum_{i=1}^{n} i^k - \sum_{i=2}^{n} \mu(i) S(\frac{n}{i})$$

注意这里要先求μ的前缀和。

求$a = f \cdot g$的前缀和

这里的条件是g完全积性。

$$a \times g = (f \cdot g) \times g = (f \times I) \cdot g$$

如求$a = \mu \cdot ID$的前缀和，$a \times ID = e \cdot ID = e$。

$$S(n) = 1 - \sum_{i=2}^{n} S(\frac{n}{i})$$

求$a = \phi \cdot ID$的前缀和，$a \times ID = ID \cdot ID = ID^2$

$$S(n) = \frac{n^3 + 3n^2 + 2n}{6} - \sum_{i=2}^{n} S(\frac{n}{i})$$

求$a = f \times g$的前缀和

可以通过交换律和结合律改变卷积的对象。

$$a \times I = f \times g \times I = (f \times I) \times g$$

如果$f \times I$可以写成另外一个函数，那么计算过程就简化了。

如$a = \mu \times ID$，$a \times I = e \times ID = ID$

如果$f \times I$不能写成另外一个函数，那么我们不能用杜教筛。

这里有另外一种方法，需要知道f(n)的前缀和。

$$\sum_{i=1}^{n} \Big(f \times g\Big)(i) = \sum_{i=1}^{n} \sum_{d|i} g(d) f(\frac{i}{d})$$

考虑哪些$g(x)f(y)$形式的(x,y)对会出现。

这里(x,y)对会出现的条件时$xy \leq n$

那么我们先枚举g(d)

$$\sum_{i=1}^{n} \Big(f \times g\Big)(i) = \sum_{d=1}^{n} g(d) \sum_{i=1}^{\frac{n}{d}} f(i)$$

只要我们知道f(i)的前缀和，我们就可以分块求$a = f \times g$的前缀和

注意点

- 注意杜教筛公式右边的枚举是从2开始的。
- 不要忘记了还要除以b(1)，也就是大部分情况下要求逆元。
- 每乘一次都要模，否则很容易出事故。
- 全部数都用unsigned long long来记不容易出事，但是很慢。另外一种替代的方法是用int来记，乘的时候临时转成unsigned long long，如果乘法不多的话会快一些。

约数问题

约数的前缀和

由于$\sigma_k = ID^k \times I$，我们可以不用杜教筛求σ_k的前缀和。

枚举每一个约数d，它出现的次数就是$\lfloor \frac{n}{d} \rfloor$

$$\sum_{i=1}^{n}\sigma_k(i)=\sum_{d=1}^{n}i^k\lfloor\frac{n}{d}\rfloor$$

可以在$O(\sqrt{n})$的时间内解决。

约数的积性问题

问题和方法

怎么算$\sigma_k(ab)$?
为了解决这个问题，我们要考虑如何枚举$a\cdot b$的约数。

假设约数是k，显然可以分解为$k=x\cdot y$且$x|a$，$y|b$。
即从a中取出一个约数x，再从b中取出一个约数y，k=xy肯定是ab的约数。
ab的每一个约数也肯定能分成xy，因为把ab质因数分解之后，每一个质因数的次幂数是a、b对应的次幂数相加，而k对应的次幂数显然比a、b的和小，显然能分到a、b中。

但是直接枚举a的约数x和b的约数y是有问题的：k会重复。
因为只要k不是质数，就有很多对约数，每一对约数就能分出一对(x,y)。
我们考虑加怎样的条件才能让k不重复。

这个条件是$gcd(x,\frac{b}{y})=1$

先证必要性。
设$gcd(x,\frac{b}{y})=d$,
则$d|x$且$d|\frac{b}{y}$，即$yd|b$
又$xy=\frac{x}{d}(yd)$,
只要$d\neq 1$，那么(x,y)和$(\frac{x}{d},yd)$都是合法的构成k的分解方案。
所以必须满足d=1才能保证每一个k只被枚举一次。

再证充分性。
将a和b分解质因数，为了满足$gcd(x,\frac{b}{y})=1$，每一个b中y取剩下的质因数，在x中都不存在。
那么我们按照这样的方法去取x和y的质因数：
对于每一个约数k，它的每一个质因数优先给y，直到b中该质因数不够为止，再分配给x。
因为k是ab的约数，所以一定分配得完，也就每一个k按照$gcd(x,\frac{b}{y})=1$的条件一定能分为x和y。
所以满足$gcd(x,\frac{b}{y})=1$的(x,y)对可以取到ab的每一个约数。

应用

$$\sigma_k(ab)=\sum_{x|a}\sum_{y|b}(xy)^k[gcd(x,\frac{b}{y})=1]$$

为了方便下一步的演算，我们可以交换$\frac{b}{y}$和y，得到：

$$\sigma_k(ab)=\sum_{x|a}\sum_{y|b}(x\frac{b}{y})^k[gcd(x,y)=1]$$

我们可以求σ_k的前缀和：

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sigma_k(ij)=\sum_{x=1}^{n}\sum_{y=1}^{n}[gcd(x,y)=1]\sum_{i=1}^{\frac{n}{x}}\sum_{j=1}^{\frac{n}{y}}(xij)^k$$

接下来就可以用莫比乌斯反演和数列求和来化简。
设1大n的k次方和为S(n)
最后得到：

$$\sum_{d=1}^{n} d^k \left(\sum_{x=1}^{\frac{n}{d}} x^k S(\lfloor \frac{n}{dx} \rfloor) \right) \left(\sum_{y=1}^{\frac{n}{d}} S(\lfloor \frac{n}{dy} \rfloor) \right)$$

优化枚举的方法

问题

有时推公式时会剩下一些求和的项，要求满足一些只能暴力枚举的条件。我们要试着优化暴力枚举

关键点

无序枚举转有序枚举。

求满足$abc \leq n$的(a,b,c)组数。

我们先做它的简化版，枚举满足$ab \leq n$的(a,b)对。
暴力的方法：

$$\sum_{i=1}^{n} \lfloor \frac{n}{i} \rfloor$$

时间复杂度$O(\sqrt{n})$
这里枚举a和b是无序的。
假如我们规定$a \leq b$，那么问题就分为两部分：
当a<b时，a只用枚举到$\sqrt{n}$，b有$\lfloor \frac{n}{a} \rfloor - a$个。注意到$a > b$和这个情况一样，所以答案要乘以二。
当a=b时，显然只有一种结果。
得到答案：

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

时间复杂度比$O(\sqrt{n})$要小。

那么回到3个的情况，我们同样可以规定$a \leq b \leq c$
那么情况分为以下几种：
a=b=c，有$n^{\frac{1}{3}}$组。
a=b<c，有3种相同情况，每种$\sum_{i=1}^{n^{\frac{1}{3}}} (\lfloor \frac{n}{i^2} \rfloor - i)$组。
a<b=c，有3种相同情况，每种$\sum_{i=1}^{n^{\frac{1}{3}}} (\lfloor \frac{n}{i} \rfloor - i)$组。
a<b<c，有6种相同情况，每种$\sum_{i=1}^{n^{\frac{1}{3}}} \sum_{j=i}^{\sqrt{\frac{n}{i}}} (\lfloor \frac{n}{ij} \rfloor - j)$组。
全部加起来再化简即可。
时间复杂度$O(n^{\frac{2}{3}})$

常见互质问题的解决方法

n以内和n互质的数个数

$\phi(n)$，定义。

n以内和n互质的数的和

引理：已知$a<n$，则$gcd(a,n)=1 \Leftrightarrow gcd(n-a,n)=1$。
证明：设$gcd(n-a,n)=d$，则$d|n-a$且$d|n$，所以$d|a$。因为$gcd(a,n)=1$，所以$d=1$。
这个引理的意义是：如果a和n互质，那么n-a和n也互质。
也就是说，和n互质的数是成对的，而且和是n。
注意到当n是偶数时，$\frac{n}{2}$一定不和n互质，也就是说$a \neq n-a$。
所以当$n \geq 2$的时候，和n互质的数一定有偶数个，也就是$\frac{\phi(n)}{2}$对。总和为：$\frac{\phi(n)}{2}n$。
这个式子对n=1不成立，补上特例写为：$\frac{\phi(n)n+[n=1]}{2}$。

n以内互质的数有多少对

只算x<y的(x,y)对，答案再乘以二。注意(1,1)会重复两次，要减去。

$$2\left(\sum_{i=1}^{n}\phi(i)\right)-1$$

m以内和n互质的数的个数

$$\sum_{i=1}^{m}[gcd(i,n)=1]$$

$$=\sum_{i=1}^{m}\sum_{d|i,d|n}\mu(d)$$

$$=\sum_{d|n}^{min(n,m)}\mu(d)\lfloor\frac{m}{d}\rfloor$$

还有一种办法适合m>>n时用。
引理：$gcd(x,n)=1 \Leftrightarrow gcd(xmodn,n)=1$
证明：设$x=ny+k$，$gcd(k,n)=d$，则$d|k,d|n$，所以$d|x$，所以$d=1$。

把m以内的数，每n个分成一组。
除了最后一组外，每一组mod n的值都包括0到n-1，共$\lfloor\frac{m}{n}\rfloor$组。显然每一组有$\phi(n)$个数和n互质。共$\lfloor\frac{m}{n}\rfloor\phi(n)$。
最后一组暴力或者用上一个方法算，再加入答案里。

n以内和m以内互质的数有多少对

$$\sum_{i=1}^{n}\sum_{j=1}^{m}[gcd(i,j)=1]$$

$$=\sum_{d=1}^{min(n,m)}\mu(d)\lfloor\frac{n}{d}\rfloor\lfloor\frac{m}{d}\rfloor$$

积性函数的递归计算

方法

当要求的函数有积性时，我们可以每次拆出第一个质因数（为了简便取一般最小的），这样剩下的部分和拆开的部分可以直接相乘。剩下的部分可以递归处理。
这种方法多见于有多个变量时的求和式。

例子

设

$$Ans(n,m)=\sum_{i=1}^{m}\phi(i\cdot n)$$

ϕ是积性函数，n是变量，所以我们试着拆出n的质因数。
这里有一个技巧：我们把n的每一个质因数取出一个，设为n'。
即$n=p_1^{k_1}p_2^{k_2}\dots p_t^{k_t}$，则$n'=p_1p_2\dots p_t$
注意$\phi(p^k\cdot p)=p^k\cdot\phi(p)$,
则

$$Ans(n,m)=\left(\frac{n}{n'}\right)Ans(n',m)=\sum_{i=1}^{m}\left(\frac{n}{n'}\right)\phi(i\cdot n')$$

现在我们只需要处理Ans(n',m)，此时的n'每一个质因数的次幂都是1。
我们每次取出n最小的质因数p，注意我们不能保证p和i互质，所以要分情况。
如果p和i互质，那么$\phi(i\cdot n')=(p-1)\phi(i\cdot\frac{n'}{p})$。
如果p和i不互质，那么$\phi(i\cdot n')=p\phi(i\cdot\frac{n'}{p})$。
注意到它们的差别只是p和p-1，由于和p不互质的部分好处理（都是p的倍数），互质的部分不好处理，我们可以先假设所有i都和p互质，也就是乘p-1，再给所有互质的部分补上1。
得到：

$$Ans(n',m)=(p-1)\sum_{i=1}^{m}\phi(i\cdot\frac{n'}{p})+\sum_{i=1}^{\lfloor\frac{m}{p}\rfloor}\phi(pi\cdot\frac{n'}{p})$$

也就是：

$$Ans(n',m)=(p-1)Ans(\frac{n'}{p},m)+Ans(n',\lfloor\frac{m}{p}\rfloor)$$

。
时间复杂度$O(log^2n)$。（要记忆化）

欧拉定理的应用

方法

欧拉定理

如果a和n互质，那么$a^{\phi(n)}\equiv 1\pmod n$

降幂

求$a^b\pmod n$
如果a和n互质，因为$a^b=a^{\phi(n)}a^{\phi(n)}\dots a^{b\bmod\phi(n)}$，所以$a^b\equiv a^{b\bmod\phi(n)}\pmod n$。
如果a和n不互质，且$b>\phi(n)$，则$a^b\equiv a^{b\bmod\phi(n)+\phi(n)}\pmod n$
证明太繁琐，这里不给出。

例子 BZOJ3884

求

$$2^{2^{2^{2^{\cdots}}}}\pmod n$$

设f(n)是模n的结果，设$2^{2^{2^{\cdots}}}=k$。
则有：$f(n)\equiv 2^{kmod\phi(n)+\phi(n)}\pmod n$。
即$f(n)\equiv 2^{f(\phi(n))+\phi(n)}\pmod n$。
递归直到n=1时，f(1)=0。
时间复杂度$O(log^2n)$。

组合数取模

问题

求$C_n^k\pmod m$

简单情况

如果n和k在10000以内，暴力杨辉三角递推即可。
如果n和k大概10^6，m是质数，暴力求阶乘和逆元即可。

卢卡斯定理

如果m是不大的质数，那么有$C_n^k\equiv C_{\lfloor\frac{n}{m}\rfloor}^{\lfloor\frac{k}{m}\rfloor}C_{nmodm}^{kmodm}\pmod m$
递归至可以直接算的范围即可。

中国剩余定理

如果m不是质数，那么我们要将它分解质因数。
设$C_n^k=a$，我们已经知道在模q_i的时候，a的模为c_i，即$a\equiv c_i\pmod{q_i}$。
所有q_i互质，而且$m=\prod q_i$
也就是说，我们知道了一个线性同余方程组，要求a模m的结果。

我们先考虑简化的问题：
我们选出一个i，保留这一个c_i，其他的c都设为0。
$a\equiv c_i\pmod{q_i}$
设除了c_i以外其他c为0情况下的结果为a_i。
由于a模其他所有q_i都是0，所以a模其他所有q的积肯定是0，即$a_i\equiv 0\pmod{\frac{m}{q_i}}$，等价于$\frac{m}{q_i}|a_i$。
再设$a_i=b_i\frac{m}{q_i}$，则$b_i\frac{m}{q_i}\equiv c_i\pmod{q_i}$，即$b_i\equiv c_i(\frac{m}{q_1})^{-1}\pmod{q_1}$
设t_i是$\frac{m}{q_i}$在模q_i下的逆元。
则这种情况下$a_i=c_it_i\frac{m}{q_i}$。
我们将所有a_i加起来，就得到了最后的答案a。
即

$$a\equiv\sum_{i=1}^{n}c_it_i\frac{m}{q_i}\pmod m$$

注意是模m，所以逆元不会消掉。
代入原来的方程组中每一个$a\equiv c_i\pmod{q_i}$，
对于$a_i=c_it_i\frac{m}{q_i}$这一项肯定是满足的，其他项j由于$q_i|\frac{m}{q_j}$，有$a_j\equiv 0\pmod{q_i}$，所以它们的和满足所有方程。

对p^k的处理

如果m分解质因数之后，每一个质因数的幂都是1，那么m是一个square_free数，可以对每一个质因数用Lucas定理解决再CRT合并。

但是如果某些质因数出现了p^k（k>1），那么不能用Lucas定理解决。这里给出解决方法。

求$C_n^k \pmod{p^k}$。

我们直接拆开$C_n^k = \frac{n!}{k!(n-k)!}$

然后把每一项中所有p拆出来，得到：

$$C_n^k = \frac{p^{k_1}\frac{n!}{p^{k_1}}}{p^{k_2}\frac{k!}{p^{k_2}}p^{k_3}\frac{(n-k)!}{p^{k_3}}}$$

然后我们只需要计算

$$C_n^k = p^{k_1-k_2-k_3}\frac{\frac{n!}{p^{k_1}}}{\frac{k!}{p^{k_2}}\frac{(n-k)!}{p^{k_3}}}$$

即可。

接下来的问题是：怎么从阶乘里取出所有p，并算出剩下的数。

我们把1到n每p个数分为一组，这样除了最后一组外，其他每组模p的结果都是0到p-1。

把模p为0的，即p的倍数单独取出来，剩下每一组模p的结果为1到p-1。

我们取出了$\lfloor \frac{n}{p} \rfloor$个p，剩下的数的乘积为$\Big((p-1)\Big)!^{\lfloor \frac{n}{p} \rfloor}(nmodp)!$。

(p-1)!我们可以预处理，再快速幂。

但是还有另外一种方法：

根据*高斯泛化的威尔逊定理*，1~m中和m互质的数模m的结果分两种情况：

1、乘积为m-1，当且仅当m=4或者p^k或者$2p^k$。

2、乘积为1，除了第一条以外所有情况。

那么在模p^k的时候，只有p=2且$k \neq 2$时，乘积为1，否则乘积为p^k-1。

(nmodp)!就只能预处理了。

注意到如果n!有p^2以上的约数的话，这一次只在每一个数中取出了一个p，剩下的数里面可能还有p。

所以要把剩下的数迭代处理，直到n<p为止。

时间复杂度$O(p+logn)$

这种方法可以处理10^7以内的p。

其他

- $gcd(n^a-1, n^b-1) = n^{gcd(a,b)}-1$
- 模的和

$$\sum_{i=1}^{n} nmodi = \sum_{i=1}^{n}\Big(n - \lfloor \frac{n}{i} \rfloor i\Big) = n^2 - \sum_{i=1}^{n}\lfloor \frac{n}{i} \rfloor i$$

- 威尔逊定理：p是质数的充分必要条件$(p-1)! \equiv -1 \pmod{p}$。