> **写在最前面：本文部分内容来自网上各大博客或是各类图书，由我个人整理，增加些许见解，仅做学习交流使用，无任何商业用途。因个人实力时间等原因，本文并非完全原创，请大家见谅。**

《算法竞赛中的初等数论》（四）正文 **0x40**反演（**ACM / OI / MO**）（十五万字符数论书）

《算法竞赛中的初等数论》（信奥 / 数竞 / ACM）前言、后记、目录索引（十五万字符的数论书）

全文目录索引链接： https://fanfansann.blog.csdn.net/article/details/113765056

0x40 反演

0x41 整除分块

0x41.1 前置知识

引理 1

$$\forall a,b,c\in\mathbb{Z},\left\lfloor\frac{a}{bc}\right\rfloor=\left\lfloor\frac{\left\lfloor\frac{a}{b}\right\rfloor}{c}\right\rfloor\tag{1}$$

> **证明**
> $$
> \begin{aligned}
> &\frac{a}{b}=\left\lfloor\frac{a}{b}\right\rfloor+r(0\leq r<1)\
> \implies
> &\left\lfloor\frac{a}{bc}\right\rfloor
> =\left\lfloor\frac{a}{b}\cdot\frac{1}{c}\right\rfloor
> =\left\lfloor \frac{1}{c}\left(\left\lfloor\frac{a}{b}\right\rfloor+r\right)\right\rfloor
> =\left\lfloor \frac{\left\lfloor\frac{a}{b}\right\rfloor}{c} +\frac{r}{c}\right\rfloor
> =\left\lfloor \frac{\left\lfloor\frac{a}{b}\right\rfloor}{c}\right\rfloor\
> &&\square
> \end{aligned}

$$**引理2**\tag{2}$$

\forall n \in \mathbb{N}{+}, \left|\left{ \left\lfloor \frac{n}{d} \right\rfloor \mid d \in \mathbb{N}{+},d\leq n \right}\right| \leq \left \lfloor 2\sqrt{n} \right\rfloor

\begin{aligned}
&\left\lfloor\frac{n}{i}\right\rfloor \leq \frac{n}{i}\
\implies
&\left\lfloor\frac{n}{ \left\lfloor\frac{n}{i}\right\rfloor }\right\rfloor
\geq \left\lfloor\frac{n}{ \frac{n}{i} }\right\rfloor
= \left\lfloor i \right\rfloor=i \
\implies
&i\leq \left\lfloor\frac{n}{ \left\lfloor\frac{n}{i}\right\rfloor }\right\rfloor=j\
&&\square
\end{aligned}

$$\text{不妨设\$}k=\left\lfloor\frac{n}{i}\right\rfloor\text{\$，考虑证明当\$}\left\lfloor\frac{n}{j}\right\rfloor=k\text{\$时，\$}j\text{\$的最大值为\$}\left\lfloor\frac{n}{k}\right\rfloor\text{\$:}\tag{3}$$

\left\lfloor\frac{n}{j}\right\rfloor=k
\iff
k\leq\frac{n}{j}<k+1
\iff
\frac{1}{k+1}<\frac{j}{n}\leq\frac{1}{k}
\iff
\frac{n}{k+1}<j\leq\dfrac{n}{k}

又因为j为整数所以$j_{\max} = \left\lfloor\frac{n}{k}\right\rfloor = \left\lfloor\frac{n}{\lfloor\frac{n}{i}\rfloor}\right\rfloor$综上所诉，值相等的区间为$[i,j] = [i,\left\lfloor\frac{n}{\lfloor\frac{n}{i}\rfloor}\right\rfloor]$综上所述，我们每次以$[i,j]$为一块，分块求和即可。

```
\sum_{i=1}^{\min (n,m)}\left\lfloor\frac{n}{i}    \right\rfloor\left\lfloor\frac{m}{i}    \right\rfloor
```

每次右边界取$\min(\left\lfloor\frac{n}{\lfloor\frac{n}{i}\rfloor}\right\rfloor, \left\lfloor\frac{m}{\lfloor\frac{m}{i}\rfloor}\right\rfloor)$，保证了两者分别相等，所以每两项的乘积相等。复杂度并没有变化。$**Code**\backslash\backslash\backslash cppfor(lll=1,r;l<=m$

\begin{aligned}\displaystyle \sum\limits{i=l}^{r}i\left\lfloor \frac{n}{i}\right\rfloor&=\sum\limits{i=l}^{r}i\left\lfloor \frac{n}{l}\right\rfloor&\&=\left\lfloor \frac{n}{l}\right\rfloor\sum\limits_{i=l}^{r}i&\&=\left\lfloor \frac{n}{l}\right\rfloor\times\frac{(r-l+1)(l+r)}{2}\end{aligned}

$$**ExampleB**\text{计算} \tag{6}$$

\sum_{i=1}^{n}i{2}\left \lfloor \frac{n}{i} \right \rfloor

$**Solution**$同样可以整除分块，只不过把等差数列求和公式换成平方数列求和的公式$**Code**\backslash\backslash\backslash cppintcal(intx)returnx*(x+1)*(2*x+1)$

\sum^{n}_{k=1}\sum{n}_{k|x}x

$**Solution**$显然它的实际意义就是求k在$[1,n]$内的所有倍数和。即对于每个k,n里会存在$\left\lfloor\frac{n}{k}\right\rfloor$个$k$的倍数，而且是一个等差数列，公差为

\sum^{n}_{k=1}\frac{k(\left \lfloor \frac{n}{k} \right \rfloor+1)\left \lfloor \frac{n}{k} \right \rfloor}{2}

显然可以直接整除分块：$\backslash\backslash\backslash cppintn, ans1, ans2; signedmain()cin>>n; for(intk=1;k<=n;++k)for(intj=1;j*k<=n;++j)ans1+=j*$

\begin{aligned}\sum\limits{i=1}^{n}\left\lceil\frac{n}{i}\right\rceil&=\sum\limits{i=1}^{n}(\left\lfloor\frac{n}{i}\right\rfloor+1)-d(n)&\\&=n-d(n)+\sum\limits_{i=1}{n}\left\lfloor\frac{n}{i}\right\rfloor
\end{aligned}

[redacted]

\begin{aligned}&\left\lfloor\cfrac{a}{b}\right\rfloor=a\mod b &\&
\left \lfloor\cfrac{a}{b}\right\rfloor=a-\left\lfloor\cfrac{a}{b}\right\rfloor\times b&\&
a=\left\lfloor\cfrac{a}{b}\right\rfloor\times (b+1)&\&
\cfrac{a}{b+1}=\left\lfloor\cfrac{a}{b}\right\rfloor\end{aligned}

[redacted]

\sum_{d\mid n}\mu(d)=
\begin{cases}
1&n=1\
0&n\neq 1\
\end{cases}

[redacted]

\begin{aligned}\sum{d\mid n}\mu(d)F(\frac{n}{d})&=\sum{d\mid n}\mu(d)\sum{k\mid \frac{n}{d}}f(k)&\&=\sum{k\mid n}f(k)\sum_{d\mid \frac{n}{k}}\mu(d)&\&=f(n)\end{aligned}

> **解释：**根据定义$F(n)=\sum_{d|n}f(d)$，我们使用$\sum_{d|n}f(d)$来替换$F(\frac{n}{d})$，再交换求和顺序（$\frac{n}{d}$意味着n的所有因子，$k\mid\frac{n}{d}$则k也是

\begin{aligned}F\ast\mu&=f1\mu &\&= f\varepsilon\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ （1\ast\mu=\varepsilon,\varepsilonf=f) &\&= f \end{aligned}

$$\text{反演公式得证}\$\square\$---**\text{形式二证明}****\text{证明一：}**\text{我们考虑逆推这个式子。} \tag{11}$$

\begin{aligned}
\sum{n|d}{\mu(\frac{d}{n})F(d)}
&= \sum{k=1}^{+\infty}{\mu(k)F(kn)}&\&
= \sum{k=1}^{+\infty}{\mu(k)\sum{kn|d}{f(d)}}&\&
= \sum{n|d}{f(d)\sum{k|\frac{d}{n}}{\mu(k)}}&\&
= \sum{n|d}{f(d)\epsilon(\frac{d}{n})}&\&
= \sum{n|d}{f(d)[\frac{d}{n}=1]}&\&
= \sum_{n|d}{f(d)[n=d]}&\&
= f(n)
\end{aligned}

[redacted]

\sum{i=a}^{b}\sum{j=c}^{d}[\gcd(i,j)=k]\qquad

枚举的范围从a, c出发，不好处理，我们考虑转换成从1出发的形式。根据容斥原理，原式可以分成4块来处理：$ans=\text{solve}(1\sim b,1\sim d)-\text{sol}$

\begin{aligned}
\sum{i=1}^{n}\sum{j=1}^{m}[\gcd(i,j)=k]
&=\sum{i=1}^{n}\sum{j=1}^{m}[\gcd(\frac i k,\frac j k) =1]\ &\\&只有 k 的倍数才有贡献，而[1,n]中k的倍数显然有\left\lfloor\frac{n}{k}\right\rfloor 个&\\&所以令\ i=ik\ 即仅枚举k的倍数即可
&\\&=\sum{i=1}^{\left\lfloor\frac{n}{k}\right\rfloor}\sum{j=1}^{\left\lfloor\frac{m}{k}\right\rfloor}[\gcd(i,j)=1]&\\&(所有k的倍数，除去k后互质才有贡献)
&\\&=\sum{i=1}^{\left\lfloor\frac{n}{k}\right\rfloor}\sum{j=1}^{\left\lfloor\frac{m}{k}\right\rfloor}\varepsilon(\gcd(i,j))
&\\&=\displaystyle\sum{i=1}^{\left\lfloor\frac{n}{k}\right\rfloor}\sum{j=1}^{\left\lfloor\frac{m}{k}\right\rfloor}\sum{d\mid \gcd(i,j)}\mu(d)
&\\&=\displaystyle\sum{d=1}^{\min{\left\lfloor\frac n k\right\rfloor,\left\lfloor\frac m k\right\rfloor}}\mu(d)\sum{i=1}^{\left\lfloor\frac{n}{k}\right\rfloor}[d\mid i]\sum{j=1}^{\left\lfloor\frac{m}{k}\right\rfloor}[d\mid j]
&\\&=\displaystyle\sum_{d=1}^{\min{\left\lfloor\frac n k\right\rfloor,\left\lfloor\frac m k\right\rfloor}}\mu(d)\left \lfloor\frac{n}{kd}\right\rfloor\left\lfloor\frac{m}{kd}\right\rfloor
\end{aligned}

[redacted]

\sum_{i=1}^n \operatorname{lcm}(i,n)

多组数据$1\leqslant T\leqslant 3\times 10^5, 1\leqslant n\leqslant 10^6$**$Solution$****方法一：** (13)

\begin{aligned}ans&=\sum{i=1}^n \operatorname{lcm}(i,n)&\\&=\sum{i=1}^n \frac{i\cdot n}{\gcd(i,n)}
&\\&=n\times \sum{i=1}^n \frac{i }{\gcd(i,n)}
&\\&=n\times \sum{d\mid n}\sum{i=1}^n \frac{i }{d}[\gcd(i,n)=d]
&\\&=n\times \sum{d\mid n}\sum{i=1}^n \frac{i }{d}[\gcd(\frac i d,\frac n d)]
&\\&=n\times \sum{d\mid n}\sum_{i=1}^{ \lfloor\frac n d \rfloor} i\cdot [\gcd(i,\frac n d)]&(i=id)
\end{aligned}

设$f(n)=\sum\limits_{i=1}^{\lfloor\frac{n}{d}\rfloor} i\cdot[\gcd(i,\frac{n}{d})]$显然它的实际含义为：$1\sim n$中所有与$n$互质的数的和。显然和为$\frac{\varphi(n)\cdot n}{2}$即：$$f(x)=\frac{\varphi(n)\cdot n}{2}$$即： (14)

\begin{aligned}\displaystyle ans&=n\times \sum{d\mid n}\sum{i=1}^{ \lfloor\frac n d \rfloor} i\cdot [\gcd(i,\frac n d)]&\\&=n\times \sum{d\mid n}\cfrac{\varphi(d)\cdot d} 2&\\&=n\times \sum{d\mid n}f(d)\end{aligned}

显然可以先预处理出所有的$f(n)$，然后利用Dirichlet前缀和计算$\sum\limits_{d\mid n}f(d)$即可。时间复杂度$O(n\log n+T)$**方法二：**当然本题还有$O($

\sum_{i=1}^n \frac{i\cdot n}{\gcd(i,n)}

将原式复制一份并且颠倒顺序，然后将n一项单独提出，可得 (16)

\frac{1}{2}\cdot \left(\sum{i=1}^{n-1}\frac{i\cdot n}{\gcd(i,n)}+\sum{i=n-1}^{1}\frac{i\cdot n}{\gcd(i,n)}\right)+n

根据$\gcd(i,n)=\gcd(n-i,n)$，可将原式化为 (17)

\frac{1}{2}\cdot \left(\sum{i=1}^{n-1}\frac{i\cdot n}{\gcd(i,n)}+\sum{i=n-1}^{1}\frac{i\cdot n}{\gcd(n-i,n)}\right)+n

两个求和式中分母相同的项可以合并。 (18)

\frac{1}{2}\cdot \sum_{i=1}^{n-1}\frac{n^2}{\gcd(i,n)}+n

即 (19)

\frac{1}{2}\cdot \sum_{i=1}^{n}\frac{n^2}{\gcd(i,n)}+\frac{n}{2}

可以将相同的$\gcd(i,n)$合并在一起计算，故只需要统计$\gcd(i,n)=d$的个数。当$\gcd(i,n)=d$时，$\gcd(\frac{i}{d},\frac{n}{d})=1$，所以$\gcd(i,n)=d$的个

\frac{1}{2}\cdot\sum_{d\mid n}\frac{n^2\cdot\varphi(\frac{n}{d})}{d}+\frac{n}{2}

变换求和顺序，设$d'=\frac{n}{d}$，合并公因式，式子化为 (21)

\frac{1}{2}n\cdot\left(\sum_{d'\mid n}d'\cdot\varphi(d')+1\right)

我们可以设$\operatorname{g}(n)=\sum\limits_{d\mid n}d\times\varphi(d)$显然函数$g$是一个积性函数，我们可以利用线性筛$O(n)$筛出$g$：**1.考虑**$\operatorname{g}(p_j^k)$显然它的约数只有$p_j^0,p_j^1$

\operatorname g(p_j^k)=\sum_{c=0}^{k}p_j^c\times \varphi(p_j^c)

又有$\varphi(p_j^c)=p_j^{c-1}\cdot(p_j-1)$，则原式可化为 (23)

\sum_{c=0}^{k}p_j^{2c-1}\times (p_j-1)

于是有 (24)

\operatorname g(p_j^{k+1})=\operatorname g(p_j^k)+p_j^{2k+1}\times (p_j-1)

2.考虑$\operatorname{g}(i\times p_j)$，**当**$p_j\mid i$令$i=a\times p_j^c(\gcd(a,p_j)=1)$，可得 (25)

\operatorname g(i\times p_j)=\operatorname g(a)\times \operatorname g(p_j^{c+1})

(26)

\operatorname g(i)=\operatorname g(a)\times \operatorname g(p_j^c)

显然有： (27)

\begin{aligned}\operatorname g(i\times p_j)-\operatorname g(i)&=\text g(a)\times \text g(p_j^{c+1})-\text g(a)\times \text g(p_j^{c})&\\&=\text g(a)\times (\text g(p_j^{c)+p_j}{2c+1}\times (p_j-1))-\text g(a)\times \text g(p_j^{c})&\\&=\operatorname g(a)\times p_j^{2c+1}\times (p_j-1)\end{aligned}

同理有 (28)

\operatorname g(i)-\operatorname g(\frac{i}{p_j})=\operatorname g(a)\times p_j^{2c-1}\times (p_j-1)

因此 (29)

\operatorname g(i\times p_j)=\operatorname g(i)+\left (\operatorname g(i)-\operatorname g(\frac{i}{p_j})\right)\times p_j^2

[redacted]

\sum{i=1}^n\sum{j=1}^m\operatorname{lcm}(i,j)\qquad (n,m\leqslant 10^7)

[redacted]

f(n)=\sum{i=1}^nt(i)\text g\left(\left\lfloor\frac{n}{i}\right\rfloor\right)\
\iff \\text g(n)=\sum{i=1}^n\mu(i)t(i)f\left(\left\lfloor\frac{n}{i}\right\rfloor\right)

[redacted]

|\overline{A_1} \cap ... \cap \overline{A_n}| = |A| - \sum\limits{i=1}^n |A_i| + \sum\limits{1 \leq i_1 < i_2 \leq n} |A{i_1} \cap A{i_2}| -... + (-1)^n \cdot |A_1 \cap ... \cap A_n|

那么根据德摩根定律显然有 (30)

|\overline{A_1 \cup ...\cup A_n}|+|\overline{A_1}\cap ...\cap \overline{A_n}|=|A|

即 (31)

|A_1\cup A_2\cup...\cup A_n|=\sum\limits{1\le i\le n}|A_i|-\sum\limits{1\le i<j\le n}|A_i\cap A_j|+...+(-1)^{n-1}\times |A_1\cap A_2\cap ...\cap A_n|

那么我们设A_i^c为A_i的补集，变形可得 (32)

|A_1^c\cap A_2^c\cap ...\cap A_n^c|=|S|-\sum\limits{1\le i\le n}|A_i|+\sum\limits{1\le i<j\le n}|A_i\cap A_j|-...+(-1)^n\times |A_1\cap A_2\cap ...\cap A_n|

显然补集的补集就是原集，则有 (33)

|A_1\cap A_2\cap ...\cap A_n|=|S|-\sum\limits{1\le i\le n}|A_i^c|+\sum\limits{1\le i<j\le n}|A_i^c\cap A_j^{c|-...+(-1)}n\times |A_1^c\cap A_2^c\cap ...\cap A_n^c|

设$f(n)$表示n个补集的交集大小，$g(n)$表示n个原集的大小则 (34)

|A_1^c\cap A_2^c\cap ...\cap A_n^c|=|S|-\sum\limits{1\le i\le n}|A_i|+\sum\limits{1\le i<j\le n}|A_i\cap A_j|-...+(-1)^n\times |A_1\cap A_2\cap ...\cap A_n|

即可表示为 (35)

f(n)=\sum\limits_{i=0}^{n(-1)}i{n\choose i}\text g(i)

而第二个等式 (36)

|A_1\cap A_2\cap ...\cap A_n|=|S|-\sum\limits{1\le i\le n}|A_i^c|+\sum\limits{1\le i<j\le n}|A_i^c\cap A_j^{c|-...+(-1)}n\times |A_1^c\cap A_2^c\cap ...\cap A_n^c|

即可表示为 (37)

\text g(n)=\sum\limits_{i=0}^{n(-1)}i{n\choose i}f(i)

即： (38)

f(n)=\sum\limits{i=0}^{n(-1)}i{n\choose i}g(i)\Leftrightarrow \text g(n)=\sum\limits{i=0}^{n(-1)}i{n\choose i}f(i)

我们令$h(n)=(-1)^n g(n)$带入上述公式即可得到 **二项式反演的第一种形式 ** ● **二项式反演的第一种形式 ** (39)

f(n)=\sum\limits{i=0}^n{n\choose i}h(i)\Leftrightarrow \frac{h(n)}{(-1)^n}=\sum\limits{i=0}^{n(-1)}i{n\choose i}f(i)

更一般的形式为： (40)

f(n)=\sum\limits{i=m}^n{n\choose i}\text g(i)\Leftrightarrow \text g(n)=\sum\limits{i=m}^{n(-1)}{n-i}{n\choose i}f(i)

● **二项式反演的第二种形式 ** (41)

f(n)=\sum\limits{i=n}^m{i\choose n}\text g(i)\Leftrightarrow \text g(n)=\sum\limits{i=n}$^{m(-1)}${i-n}{i\choose n}f(i)

$\big($ n\$，即\$g(i)\$也在\$$f(i)$\$中被计算了\$$\binom{i}{n}$\$次，然后根据容斥原理，可得\$g$(n)=\sum_{i=n}^{m}(-1)^{i-n}\binom{i}{n}f(i)$\$。> ∗∗形式一证明∗∗\$\$

$$
\begin{aligned}
f(n) &= \sum_{i=0}^{n}\binom{n}{i}\sum \\
&= \sum_{i=0}^{n}\sum_{j=0}^{i}(- \\
&= \sum_{j=0}^{n}f(j)\sum \\
&= \sum_{j=0}^{n}\binom{n}{>j} \\
> &= \sum_{j=0}^{n}\binom{n}{>j} \\
> &= \sum_{j=0}^{n}\binom{n}{j}f
\end{aligned}
$$

\begin{aligned}f(k)&={n\choose k}(2^{{2}{n-k}}-1)&\\&=\sum\limits_{i=k}^n{i\choose k}\text g(i)\end{aligned}

进行二项式反演显然有： (43)

\begin{aligned}g(k)&=\sum\limits{i=k}$^{n(-1)}${i-k}{i\choose k}f(i)&\\&=\sum\limits{i=k}$^{n(-1)}${i-k}{i\choose k}{n\choose i}(2^{{2}{n-i}}-1)\end{aligned}

[redacted]

\forall k,[n\mid k]=\frac{1}{n}\sum_{i=0}$^{{n-1}\omega\_n}${ik}

[redacted]

\begin{aligned}\sum{T \in S}(-1)^{|T|+1} \min {T} &=\sum{i=1}^{n} a{i} \sum{j=0}$^{{n-i}(-1)}${j}\left(\begin{array}{c}n-i \j\end{array}\right) \&=\sum{i=1}^{n} a{i}[n-i=0] \&=\sum{i=1}^{n} a{i}[n=i] \&=a_{n} \&=\max {S}\end{aligned}

$Min-Max$容斥在期望意义下也是成立的：，设\$$E(x)$\$表示元素\$$x$\$出现的期望操作次数，则： (44)

E(\max(S))=\sum{T \subseteq S}(-1)^{|T|-1}E(\min(T))\
E(\min(S))=\sum{T \subseteq S}(-1)^{|T|-1}E(\max(T))

∗∗拓展$Min-Max$容斥∗∗设\$$k^{th}\max(S)$\$表示S的第k大元素，则 (45)

k^{th}\max(S)=\sum_{T \subseteq S} (-1)^{|T|-k} {|T|-1 \choose k-1} \min(T)

[redacted]

\displaystyle [x$^{n]F(x)=\frac{1}{n}[x}${-1}](\frac{1}{G(x)})$^{n=\frac{1}{n}[x}${n-1}](\frac{x}{G(x)})^n

扩展拉格朗日反演 (46)

[x$^{n]H(F(x))=\frac{1}{n}[x}${-1}]H'(x)(\frac{1}{G(x)})$^{n=\frac{1}{n}[x}${n-1}]H'(x)(\frac{x}{G(x)})^n

∗∗$Problem A.$大朋友和多叉树∗∗（$[BZOJ3684](https://darkbzoj.tk/problem/3684)$）我们的大朋友很喜欢计算机科学，而且尤其喜欢多叉树。对

H(x)=x+\prod{i\in D}H^i(x)\\Rightarrow H(x)-\prod{i\in D}H^i(x)=x

令\$$F(x)=H(x)$\$,\$$G(x)=x-\prod_{i\in D}x^i$\$，显然有\$$G(F(x))=x$\$因此我们就可以进行拉格朗日反演： (48)

[x$^{n]F(x)=\frac{1}{n}[x}${-1}]\frac{1}{G(x)^n}

[redacted]

\sum{i=1}^n\sum{j=1}$^{mna_ib_j=n\sum_{i=1}}$na_i\sum_{j=1}^mb_j

∗∗技巧$4A.2$：交换枚举顺序∗∗ (49)

\sum{i=1}^na_i\sum{j=1}$^{mb_j=\sum_{j=1}}$mb_j\sum_{i=1}^na_i

(50)

\sum{i=1}^na_i\sum{d|i}b_d=\sum{d=1}^nb_d\sum{i=1}^{\lfloor\frac nd\rfloor}a_{id}

这样可以将枚举的约数变成枚举倍数。**技巧 $4A.3$：暴力破解法**无从下手的时候，多一个暴力枚举判断，多一重枚举，构造出判断式，尝试构造$ε

\begin{aligned}&\ \ \ \ \ \ \ \ \sum{i=1}^n\sum{j=1}^mf(\gcd(i,j))&\&
=\sum{d=1}^{min(n,m)}\sum{i=1}n\sum_{j=1}mf(d)[\gcd(i,j)=d]&\&
=\sum{d=1}^{min(n,m)}f(d)\sum{i=1}n\sum_{j=1}m[\gcd(i,j)=d]&\& \cdots \end{aligned}

$$**技巧 4A.4：减少未知数**\tag{52}$$

\displaystyle \sum{p|k}\sum{d=1}^{\lfloor\frac{n}{p}\rfloor}\mu(d)\lfloor\frac{n}{pd}\rfloor\lfloor\frac{m}{pd}\rfloor

$$可以令 \$k = pd\$（尽量减少未知数，减少枚举）\tag{53}$$

\displaystyle \sum{p|k}\sum{d=1}^{\lfloor\frac{n}{p}\rfloor}\mu(\frac{k}{p})\lfloor\frac{n}{k}\rfloor\lfloor\frac{m}{k}\rfloor

$$**技巧 4A.5：狄利克雷卷积优化**尽力拼凑成积性函数卷积的形式\tag{54}$$

\begin{aligned}&\ \ \ \ \ \sum{d=1}nd3\sum{k=1}^{\left\lfloor\frac n d\right\rfloor}\mu(k)k2(\sum_{i=1}{\left\lfloor\frac n {dk}\right\rfloor}i)2&\\&=\sum_{T=1}{n}T^2\mu(\frac T d)d(\sum{i=1}^{\left\lfloor\frac n {dk}\right\rfloor}i)^2&\&=\sum{T=1}{n}T2\varphi(\frac T d \times d)(\sum{i=1}^{\left\lfloor\frac n {dk}\right\rfloor}i)^2&\&=\sum{T=1}{n}T2\varphi(T)\sum_{i=1}^{\left\lfloor\frac n {dk}\right\rfloor}i^3\end{aligned}

$$**技巧 4A.6：前缀和公式**\tag{55}$$

\begin{aligned}&1+2+3+\cdots+(n-1)\ \ \ \ \ \ =\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \frac 1 2n^2-\frac 1 2n \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ = \frac{n(n-1)} 2&\&12+22+32+\cdots+(n-1)2=\ \ \ \ \ \ \ \ \ \ \ \ \frac 1 3n^3-\frac 1 2 {n^2} +\frac 1 6n \ \ \ \ \ \ \ \ = \frac{n(n-1)(2n-1)} 6&\&13+23+33+\cdots+(n-1)3=\ \ \ \ \ \ \ \ \ \frac 1 4n^4-\frac 1 2 {n^3} +\frac 1 4n^2\ \ \ \ \ \ \ = \frac{n2(n-1)2} 4&\&14+24+34+\cdots+(n-1)4=\ \ \ \ \frac 1 5 n^5-\frac 1 2n^4+\frac 1 3 n^3-\frac n {30} \ \ =\frac{n(n-1)(2n-1)(3n2-3n-1)}{30}&\\&15+25+35+\cdots+(n-1)^5=\frac 1 6n^6-\frac 1 2 n^5+\frac 5 {12}n^4-\frac 1 {12}n2=\frac{n2(n-1)2(2n2-2n-1)}{12}\end{aligned}
$$

> **证明**
>
> 规定 $C(n,k)$ 表示从 n 个数中取 k 个的组合数。
>
> $2^{n+1}=1+C(n+1,1)1^n+C(n+1,2)1^{n-1}+C(n+1,3)1^{n-2}+\cdots+C(n+1,n+1)$
> $3^{n+1}=2^{n+1}+C(n+1,1)2^n+C(n+1,2)2^{n-1}+c(n+1,3)2^{n-2}+\cdots+c(n+1,n+1)$
> ....
> $(n+1)^{n+1}=n^{n+1}+C(n+1,1)n^n+C(n+2,2)n^{n-1}+\cdots+C(n+1,n+1)$
> 上面 n 个式子相加
> $(n+1)^{n+1}=1+(n+1)(1^n+2^n+\ldots+n^n)+C(n+2,2)(1^{n-1}+2^{n-1}+\ldots+n^{n-1})+\ldots+C(n+1,n)(1+2+3+\ldots+n)+n$
> 整理得
> $1^n+2^n+\cdots+n^n=(n+1)^n-\frac{1}{n+1}-(\frac{1}{n+1})(C(n+2,2)(1^{n-1}+2^{n-1}+..+n^{n-1})+\ldots+C(n+1,n)(1+2+3+\cdots+n)+n)$
> 我们知道 1 次的前缀和为: $1+2+\cdots+n=\frac{n(1+n)}{2}$
> 因此我们就可以递推得到 2次，3 次，乃至 n 次前缀和公式

0x4B. Dirichlet 前缀和

0x4B.1 Dirichlet 前缀和

Template Problem Dirichlet 前缀和 （P5495）

给定一个长度为 n 的数列 $a_1,a_2,a_3,\ldots,a_n$ 。

现在你要求出一个长度为 n 的数列 $b_1,b_2,b_3,\ldots,b_n$ ，满足

$b_d=\sum_{i|d}a_i$

Solution

如果我们直接枚举倍数的话显然可以做到 $O(\displaystyle\sum_{i=1}^{n}\left\lfloor\frac{n}{i}\right\rfloor)=O(n\log n)$ ，考虑优化。

根据唯一分解定理，我们可以设 $i=\prod p_d^{\alpha_d},j=\prod p_d^{\beta_d}$ ，则 a_i 贡献到 b_j 当且仅当 $\forall d,\alpha_d\le\beta_d$。

发现这个实际上就是一个关于质因子分解后的指数的高维前缀和（例如FMT），即每个数字会被它除以所有质因子的那个数转移过来，所有的质因子构成了一组正交基，核心代码：

```
for(int i = 1; i <= cnt && primes[i] <= n; ++ i)
    for(int j = 1; j * primes[i] <= n; ++ j)
        a[j * primes[i]] += a[j];
```

显然时间复杂度同埃氏筛：$O(n\log\log n)$

Code

```
#define uint unsigned int
uint seed;
inline uint getnext() {
    seed ^= seed << 13;
    seed ^= seed >> 17;
    seed ^= seed << 5;
    return seed;
}
uint a[N], b[N];
uint ans;
int n, m;
bool vis[N];
int primes[N], cnt;
void get_primes(int n)
{
    for(int i = 2; i <= n; ++ i) {
        if(vis[i] == 0)
            primes[ ++ cnt] = i;
        for(int j = 1; j <= cnt && i * primes[j] <= n; ++ j) {
            vis[i * primes[j]] = true;
            if(i % primes[j] == 0) break;
        }
    }
}
int main()
{
    get_primes(N - 1);
    scanf("%d%u", &n, &seed);
    for(int i = 1; i <= n; ++ i)
        a[i] = getnext();
    for(int i = 1; i <= cnt && primes[i] <= n; ++ i)
        for(int j = 1; j * primes[i] <= n; ++ j)
            a[j * primes[i]] += a[j];
    ans = a[1];
    for(int i = 2; i <= n; ++ i)
        ans ^= a[i];
    printf("%u\n", ans);
    return 0;
}
```

0x4B.2 Dirichlet 后缀和

$b[i]=\sum_{i|d}a[d]$

我们这里是从一个数字本身 转移到 这个数字除以所有质因子的数。

```
    for(int i = 1; i <= cnt && primes[i] <= n; ++ i)
        for(int j = n / primes[i]; j ; -- j)
            a[j] += a[j * primes[i]];
```

0x4B.2 倒推 Dirichlet 前缀和

$b[i]=\sum_{d|i}a[d]$

这里是我们知道数组 b ，求数组 a

```
    for(int i = cnt; i ; -- i)
        for(int j = n / primes[i]; j ; -- j)
            a[j * primes[i]] -= a[j];
```

0x4B.3 倒推 Dirichlet 后缀和

$b[i]=\sum_{i|d}a[d]$

同上，我们知道数组 b ，求数组 a

```
for(int i = cnt; i ; -- i)
    for(int j = 1; j * primes[i] <= n; ++ j)
        a[j] -= a[j * primes[i]];
```