数论函数选讲

任之洲

绍兴市第一中学

2016 年 3 月 22 日

Content

- 一些定义与常识
- 杜教筛
- 一种积性函数求和方法

- 各块内容之间关联不大。
- ~~可能引起公式恐惧症患者的轻度不适。~~
- 如何和你了解的不同，请装作没看见。

id=43116906

数论函数

数论函数

定义域为正整数的函数。

数论函数

数论函数

定义域为正整数的函数。

积性函数

对于所有$gcd(a,b)=1$，满足$f(ab)=f(a)f(b)$。

数论函数

数论函数

定义域为正整数的函数。

积性函数

对于所有$gcd(a,b)=1$，满足$f(ab)=f(a)f(b)$。

完全积性函数

对于所有a,b，满足$f(ab)=f(a)f(b)$。

常见的积性函数

Euler函数

$\varphi(n)$表示$[1, n]$中与n互质的数的个数。
$\varphi(n) = n\prod_{p\in P}\frac{p-1}{p}$，其中$P$是$n$的不同质因子集合。

常见的积性函数

Euler函数

$\varphi(n)$表示$[1,n]$中与n互质的数的个数。

$\varphi(n)=n\prod_{p\in P}\frac{p-1}{p}$，其中$P$是$n$的不同质因子集合。

Möbius函数

若n有平方数因子，则$\mu(n)=0$。

否则，若n为k个不同质数之积，则$\mu(n)=(-1)^k$。

常见的积性函数

Euler函数

$\varphi(n)$表示$[1,n]$中与n互质的数的个数。
$\varphi(n)=n\prod_{p\in P}\frac{p-1}{p}$，其中$P$是$n$的不同质因子集合。

Möbius函数

若n有平方数因子，则$\mu(n)=0$。
否则，若n为k个不同质数之积，则$\mu(n)=(-1)^k$。

除数函数

$\sigma_k(n)$表示所有正因子的k次幂之和。
$d(n)=\sigma_0(n)$，表示n的正因子个数。
$\sigma(n)=\sigma_1(n)$，表示n的所有正因子之和。

常见的完全积性函数

幂函数

$Id_k(n) = n^k$

$1(n) = Id_0(n) = 1$

$Id(n) = Id_1(n) = n$

常见的完全积性函数

幂函数

$$Id_k(n) = n^k$$
$$1(n) = Id_0(n) = 1$$
$$Id(n) = Id_1(n) = n$$

单位函数

$$e(n) = \epsilon(n) = \begin{cases} 1 & n = 1 \\ 0 & n > 1 \end{cases}$$

Dirichlet卷积

Dirichlet卷积

定义两个数论函数f, g的Dirichlet卷积

$$(f * g)(n) = \sum_{d|n} f(d)g(\frac{n}{d})$$

Dirichlet卷积

Dirichlet卷积

定义两个数论函数f,g的Dirichlet卷积

$$(f*g)(n)=\sum_{d|n}f(d)g(\frac{n}{d})$$

Dirichlet卷积的性质

交换律：$f*g=g*f$
结合律：$(f*g)*h=f*(g*h)$
分配律：$f*(g+h)=f*g+f*h$
单位元：$f*\epsilon=\epsilon*f$
若f,g均为积性函数，则$f*g$也为积性函数。

常见的Dirichlet卷积

常见的Dirichlet卷积

$d(n)=\sum_{d|n}1$，即$d=1*1$。
$\sigma(n)=\sum_{d|n}d$，即$\sigma=d*1$。
$\varphi(n)=\sum_{d|n}\mu(d)\frac{n}{d}$，即$\varphi=\mu*Id$。
$\epsilon(n)=\sum_{d|n}\mu(d)$，即$\epsilon=\mu*1$。

常见的Dirichlet卷积

常见的Dirichlet卷积

$d(n)=\sum_{d|n}1$，即$d=1*1$。
$\sigma(n)=\sum_{d|n}d$，即$\sigma=d*1$。
$\varphi(n)=\sum_{d|n}\mu(d)\frac{n}{d}$，即$\varphi=\mu*Id$。
$\epsilon(n)=\sum_{d|n}\mu(d)$，即$\epsilon=\mu*1$。

- 设n有$k(k>0)$个不同质因子。

常见的Dirichlet卷积

常见的Dirichlet卷积

$d(n)=\sum_{d|n}1$，即$d=1*1$。
$\sigma(n)=\sum_{d|n}d$，即$\sigma=d*1$。
$\varphi(n)=\sum_{d|n}\mu(d)\frac{n}{d}$，即$\varphi=\mu*Id$。
$\epsilon(n)=\sum_{d|n}\mu(d)$，即$\epsilon=\mu*1$。

- 设n有$k(k>0)$个不同质因子。
- 那么

$$\sum_{d|n}\mu(d) = \sum_{i=0}^{k}(-1)^i\binom{k}{i}$$

常见的Dirichlet卷积

常见的Dirichlet卷积

$d(n) = \sum_{d|n} 1$，即$d = 1 * 1$。
$\sigma(n) = \sum_{d|n} d$，即$\sigma = d * 1$。
$\varphi(n) = \sum_{d|n} \mu(d)\frac{n}{d}$，即$\varphi = \mu * Id$。
$\epsilon(n) = \sum_{d|n} \mu(d)$，即$\epsilon = \mu * 1$。

- 设n有$k(k > 0)$个不同质因子。
- 那么

$$
\begin{aligned}
\sum_{d|n} \mu(d) &= \sum_{i=0}^{k} (-1)^i \binom{k}{i} \\
&= (1-1)^k = 0
\end{aligned}
$$

Möbius反演

Möbius反演

如果有两个函数f,g满足

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

则它们也满足

$$g(n)=\sum_{d|n}\mu(d)f(\frac{n}{d})$$

反之亦然，即

$$f=g*1\Leftrightarrow g=\mu*f$$

Möbius反演

- 设

$$f = g * 1$$

Möbius反演

- 设

$$f = g * 1$$

- 两侧都卷上μ，得

$$\mu * f = \mu * g * 1$$

Möbius反演

- 设

$$f = g * 1$$

- 两侧都卷上μ，得

$$\mu * f = \mu * g * 1$$

- 整理得

$$\mu * f = \epsilon * g = g$$

Möbius反演

- 设

$$f = g * 1$$

- 两侧都卷上μ，得

$$\mu * f = \mu * g * 1$$

- 整理得

$$\mu * f = \epsilon * g = g$$

- 两侧都卷上μ，得

$$g * 1 = \mu * f * 1 = \epsilon * f = f$$

线性筛

- 设pr_i为i的最小质因子。

线性筛

- 设pr_i为i的最小质因子。
- $2 \to n$枚举i

线性筛

- 设pr_i为i的最小质因子。
- $2 \to n$枚举i
  - 假如pr_i还没有被计算出来，那么i为质数，那么$pr_i = i$。

线性筛

- 设pr_i为i的最小质因子。
- $2 \to n$枚举i
  - 假如pr_i还没有被计算出来，那么i为质数，那么$pr_i = i$。
  - 枚举所有不超过pr_i的质数p，使$pr_{ip} = p$。

线性筛

- 设pr_i为i的最小质因子。
- $2 \to n$枚举i
  - 假如pr_i还没有被计算出来，那么i为质数，那么$pr_i = i$。
  - 枚举所有不超过pr_i的质数p，使$pr_{ip} = p$。
- 相当于将每个数的质因子降序分解了。

线性筛

- 设pr_i为i的最小质因子。
- $2 \to n$枚举i
  - 假如pr_i还没有被计算出来，那么i为质数，那么$pr_i = i$。
  - 枚举所有不超过pr_i的质数p，使$pr_{ip} = p$。
- 相当于将每个数的质因子降序分解了。
  - 分解方式是唯一的，于是每个数只会被筛到一次。

线性筛

- 设pr_i为i的最小质因子。
- $2 \to n$枚举i
  - 假如pr_i还没有被计算出来，那么i为质数，那么$pr_i = i$。
  - 枚举所有不超过pr_i的质数p，使$pr_{ip} = p$。

- 相当于将每个数的质因子降序分解了。
  - 分解方式是唯一的，于是每个数只会被筛到一次。
  - 复杂度$O(n)$。

线性筛

- 设pr_i为i的最小质因子。
- $2 \rightarrow n$枚举i
  - 假如pr_i还没有被计算出来，那么i为质数，那么$pr_i = i$。
  - 枚举所有不超过pr_i的质数p，使$pr_{ip} = p$。
- 相当于将每个数的质因子降序分解了。
  - 分解方式是唯一的，于是每个数只会被筛到一次。
  - 复杂度$O(n)$。
- 筛出质数的同时，得到了每个数的最小质因子。

线性筛

- 设pr_i为i的最小质因子。
- $2 \to n$枚举i
  - 假如pr_i还没有被计算出来，那么i为质数，那么$pr_i = i$。
  - 枚举所有不超过pr_i的质数p，使$pr_{ip} = p$。
- 相当于将每个数的质因子降序分解了。
  - 分解方式是唯一的，于是每个数只会被筛到一次。
  - 复杂度$O(n)$。
- 筛出质数的同时，得到了每个数的最小质因子。
  - 进一步可以得到每个数去除所有最小质因子后的结果。

线性筛

- 设pr_i为i的最小质因子。
- $2 \to n$枚举i
  - 假如pr_i还没有被计算出来，那么i为质数，那么$pr_i = i$。
  - 枚举所有不超过pr_i的质数p，使$pr_{ip} = p$。
- 相当于将每个数的质因子降序分解了。
  - 分解方式是唯一的，于是每个数只会被筛到一次。
  - 复杂度$O(n)$。
- 筛出质数的同时，得到了每个数的最小质因子。
  - 进一步可以得到每个数去除所有最小质因子后的结果。
  - 方便地计算积性函数。

Sums of totients of powers
Project Euler 512

题目大意

设

$$f(n) = \left(\sum_{i=1}^{n} \varphi(n^i)\right) mod\ (n+1)$$

$$g(n) = \sum_{i=1}^{n} f(i)$$

如$g(100) = 2007$。
求$g(5*10^8)$。

- 设n有m个不同质因子$p_1 \sim p_m$，那么有

$$\varphi(n) = n\prod_{i=1}^{m}\frac{p_i - 1}{p_i}$$

- 设n有m个不同质因子$p_1 \sim p_m$，那么有

$$\varphi(n) = n\prod_{i=1}^{m}\frac{p_i - 1}{p_i}$$

- 易得

$$\varphi(n^k) = n^{k-1}\varphi(n)$$

Sums of totients of powers

Project Euler 512

- 设n有m个不同质因子$p_1 \sim p_m$，那么有

$$\varphi(n) = n\prod_{i=1}^{m}\frac{p_i - 1}{p_i}$$

- 易得

$$\varphi(n^k) = n^{k-1}\varphi(n)$$

- 由于$n \equiv -1 \pmod{n+1}$，所以

$$f(n) = \varphi(n)\sum_{i=1}^{n}(-1)^{i-1}$$

- 设n有m个不同质因子$p_1 \sim p_m$，那么有

$$\varphi(n) = n\prod_{i=1}^{m}\frac{p_i - 1}{p_i}$$

- 易得

$$\varphi(n^k) = n^{k-1}\varphi(n)$$

- 由于$n \equiv -1 \ (mod \ \ n+1)$，所以

$$f(n) = \varphi(n)\sum_{i=1}^{n}(-1)^{i-1}$$

- 当n为奇数时，$f(n) = \varphi(n)$，否则$f(n) = 0$。利用线性筛可以$O(n)$计算。

Sums of totients of powers
Project Euler 512

- 设n有m个不同质因子$p_1 \sim p_m$，那么有

$$\varphi(n) = n \prod_{i=1}^{m} \frac{p_i - 1}{p_i}$$

- 易得

$$\varphi(n^k) = n^{k-1}\varphi(n)$$

- 由于$n \equiv -1 \pmod{n+1}$，所以

$$f(n) = \varphi(n) \sum_{i=1}^{n} (-1)^{i-1}$$

- 当n为奇数时，$f(n) = \varphi(n)$，否则$f(n) = 0$。利用线性筛可以$O(n)$计算。
  - 数组开不下？

- 设n有m个不同质因子$p_1 \sim p_m$，那么有

$$\varphi(n) = n\prod_{i=1}^{m}\frac{p_i - 1}{p_i}$$

- 易得

$$\varphi(n^k) = n^{k-1}\varphi(n)$$

- 由于$n \equiv -1 \pmod{n+1}$，所以

$$f(n) = \varphi(n)\sum_{i=1}^{n}(-1)^{i-1}$$

- 当n为奇数时，$f(n) = \varphi(n)$，否则$f(n) = 0$。利用线性筛可以$O(n)$计算。
  - 数组开不下？省去偶数位置。

题目大意

设

$$
\begin{aligned}
f(n) &= \sum_{d|n} gcd(d, \frac{n}{d}) \\
F(n) &= \sum_{i=1}^{n} f(i)
\end{aligned}
$$

如$F(10) = 32$，$F(1000) = 12776$。
求$F(10^{15})$。

$$F(n) = \sum_{i=1}^{n} \sum_{d|i} gcd(d, \frac{i}{d})$$

GCD of Divisors

Project Euler 530

$$
\begin{aligned}
F(n) &= \sum_{i=1}^{n} \sum_{d|i} gcd(d, \frac{i}{d}) \\
&= \sum_{d} d \sum_{i=1}^{\left\lfloor \frac{n}{d^2} \right\rfloor} \sum_{d'|i} \epsilon(gcd(d', \frac{i}{d'}))
\end{aligned}
$$

GCD of Divisors
Project Euler 530

$$
\begin{aligned}
F(n) &= \sum_{i=1}^{n} \sum_{d|i} gcd(d, \frac{i}{d}) \\
&= \sum_{d} d \sum_{i=1}^{\left\lfloor \frac{n}{d^2} \right\rfloor} \sum_{d'|i} \epsilon(gcd(d', \frac{i}{d'})) \\
&= \sum_{d} \sum_{d'} d\mu(d') \sum_{i=1}^{\left\lfloor \frac{n}{(dd')^2} \right\rfloor} \sigma_0(i)
\end{aligned}
$$

GCD of Divisors
Project Euler 530

$$
\begin{aligned}
F(n) &= \sum_{i=1}^{n} \sum_{d|i} gcd(d, \frac{i}{d}) \\
&= \sum_{d} d \sum_{i=1}^{\lfloor \frac{n}{d^2} \rfloor} \sum_{d'|i} \epsilon(gcd(d', \frac{i}{d'})) \\
&= \sum_{d} \sum_{d'} d\mu(d') \sum_{i=1}^{\lfloor \frac{n}{(dd')^2} \rfloor} \sigma_0(i) \\
&= \sum_{d=1}^{n} \varphi(d) \sum_{i=1}^{\lfloor \frac{n}{d^2} \rfloor} \sigma_0(i)
\end{aligned}
$$

- 需要计算

$$F(n) = \sum_{d=1}^{n} \varphi(d) \sum_{i=1}^{\left\lfloor \frac{n}{d^2} \right\rfloor} \sigma_0(i)$$

GCD of Divisors
Project Euler 530

- 需要计算

$$F(n) = \sum_{d=1}^{n} \varphi(d) \sum_{i=1}^{\left\lfloor \frac{n}{d^2} \right\rfloor} \sigma_0(i)$$

- $\varphi(d)$只需要前$\sqrt{n}$项，约数个数和可以$O(\sqrt{n})$计算

$$\sum_{i=1}^{n} \sigma_0(i) = \sum_{i=1}^{n} \left\lfloor \frac{n}{i} \right\rfloor$$

- 需要计算

$$F(n)=\sum_{d=1}^{n}\varphi(d)\sum_{i=1}^{\left\lfloor\frac{n}{d^2}\right\rfloor}\sigma_0(i)$$

- $\varphi(d)$只需要前$\sqrt{n}$项，约数个数和可以$O(\sqrt{n})$计算

$$\sum_{i=1}^{n}\sigma_0(i)=\sum_{i=1}^{n}\left\lfloor\frac{n}{i}\right\rfloor$$

- 暴力计算的复杂度为

$$\sum_{i=1}^{\lfloor\sqrt{n}\rfloor}O\left(\sqrt{\frac{n}{i^2}}\right)=\sum_{i=1}^{\lfloor\sqrt{n}\rfloor}O\left(\frac{\sqrt{n}}{i}\right)=O(\sqrt{n}\ln n)$$

题目大意

设$R(M)$为所有满足以下条件的数对p,q的$\frac{1}{pq}$的和。

- $1 \le p < q \le M$
- $p + q \ge M$
- $gcd(p,q) = 1$

设$S(N) = \sum_{i=2}^{N} R(i)$。
如$S(2) = R(2) = \frac{1}{2}$，$S(10) \approx 6.9147$，$S(100) \approx 58.2962$。
求$S(10^7)$，保留小数点后4位。

- 原问题为求

$$S(N)=\sum_{i=1}^{N}\sum_{p=1}^{i}\sum_{q=p+1}^{i}\frac{1}{pq}[p+q\geq i][gcd(p,q)=1]$$

- 原问题为求

$$S(N)=\sum_{i=1}^{N}\sum_{p=1}^{i}\sum_{q=p+1}^{i}\frac{1}{pq}[p+q\geq i][gcd(p,q)=1]$$

- 设

$$F(N)=\sum_{p=1}^{N}\frac{1}{p}$$

The inverse summation of coprime couples
Project Euler 441

- 原问题为求

$$S(N) = \sum_{i=1}^{N} \sum_{p=1}^{i} \sum_{q=p+1}^{i} \frac{1}{pq}[p + q \geq i][gcd(p, q) = 1]$$

- 设

$$\begin{aligned} F(N) &= \sum_{p=1}^{N} \frac{1}{p} \\ G(N) &= \sum_{p=1}^{N} \sum_{q=p+1}^{N} \frac{1}{pq}[p + q = N] \end{aligned}$$

- 原问题为求

$$S(N)=\sum_{i=1}^{N}\sum_{p=1}^{i}\sum_{q=p+1}^{i}\frac{1}{pq}[p+q\geq i][gcd(p,q)=1]$$

- 设

$$\begin{aligned}F(N) &= \sum_{p=1}^{N}\frac{1}{p}\\ G(N) &= \sum_{p=1}^{N}\sum_{q=p+1}^{N}\frac{1}{pq}[p+q=N]\\ T(N) &= \sum_{p=1}^{N}\sum_{q=p+1}^{N}\frac{1}{pq}[p+q\geq N]\end{aligned}$$

The inverse summation of coprime couples
Project Euler 441

$$T(N) = \sum_{p=1}^{N} \sum_{q=p+1}^{N} \frac{1}{pq}[p + q \geq N]$$

The inverse summation of coprime couples

Project Euler 441

$$
\begin{aligned}
T(N) &= \sum_{p=1}^{N} \sum_{q=p+1}^{N} \frac{1}{pq}[p+q \geq N] \\
&= \sum_{p=1}^{N-1} \sum_{q=p+1}^{N-1} \frac{1}{pq}\left([p+q \geq N-1] - [p+q = N-1]\right) \\
&\quad + \sum_{p=1}^{N-1} \frac{1}{Np}
\end{aligned}
$$

$$
\begin{aligned}
T(N) &= \sum_{p=1}^{N}\sum_{q=p+1}^{N}\frac{1}{pq}[p+q\geq N] \\
&= \sum_{p=1}^{N-1}\sum_{q=p+1}^{N-1}\frac{1}{pq}\left([p+q\geq N-1]-[p+q=N-1]\right) \\
&\quad+\sum_{p=1}^{N-1}\frac{1}{Np} \\
&= T(N-1)-G(N-1)+\frac{F(N-1)}{N}
\end{aligned}
$$

$$G(N) = \sum_{p=1}^{N}\sum_{q=p+1}^{N}\frac{1}{pq}[p+q=N]$$

$$
\begin{aligned}
G(N) &= \sum_{p=1}^{N}\sum_{q=p+1}^{N}\frac{1}{pq}[p+q=N] \\
&= \frac{1}{2}\left(\sum_{p=1}^{N-1}\frac{1}{p(N-p)} - \frac{1}{(\frac{N}{2})^2}[N \ mod \ 2 = 0]\right)
\end{aligned}
$$

$$
\begin{aligned}
G(N) &= \sum_{p=1}^{N}\sum_{q=p+1}^{N}\frac{1}{pq}[p+q=N] \\
&= \frac{1}{2}\left(\sum_{p=1}^{N-1}\frac{1}{p(N-p)} - \frac{1}{(\frac{N}{2})^2}[N\ mod\ 2=0]\right) \\
&= \frac{1}{2}\left(\sum_{p=1}^{N-1}\frac{1}{N}(\frac{1}{p}+\frac{1}{N-p}) - \frac{4}{N^2}[N\ mod\ 2=0]\right)
\end{aligned}
$$

$$
\begin{aligned}
G(N) &= \sum_{p=1}^{N}\sum_{q=p+1}^{N}\frac{1}{pq}[p+q=N] \\
&= \frac{1}{2}\left(\sum_{p=1}^{N-1}\frac{1}{p(N-p)} - \frac{1}{(\frac{N}{2})^2}[N\ mod\ 2=0]\right) \\
&= \frac{1}{2}\left(\sum_{p=1}^{N-1}\frac{1}{N}(\frac{1}{p}+\frac{1}{N-p}) - \frac{4}{N^2}[N\ mod\ 2=0]\right) \\
&= \frac{1}{2}\left(2\sum_{p=1}^{N-1}\frac{1}{Np} - \frac{4}{N^2}[N\ mod\ 2=0]\right)
\end{aligned}
$$

$$\begin{aligned}
G(N) &= \sum_{p=1}^{N}\sum_{q=p+1}^{N}\frac{1}{pq}[p+q=N] \\
&= \frac{1}{2}\left(\sum_{p=1}^{N-1}\frac{1}{p(N-p)} - \frac{1}{(\frac{N}{2})^2}[N\ mod\ 2=0]\right) \\
&= \frac{1}{2}\left(\sum_{p=1}^{N-1}\frac{1}{N}(\frac{1}{p}+\frac{1}{N-p}) - \frac{4}{N^2}[N\ mod\ 2=0]\right) \\
&= \frac{1}{2}\left(2\sum_{p=1}^{N-1}\frac{1}{Np} - \frac{4}{N^2}[N\ mod\ 2=0]\right) \\
&= \frac{F(N-1)}{N} - \frac{2}{N^2}[N\ mod\ 2=0]
\end{aligned}$$

The inverse summation of coprime couples

Project Euler 441

$$S(N) = \sum_{i=1}^{N}\sum_{p=1}^{i}\sum_{q=p+1}^{i}\frac{1}{pq}[p+q\geq i][gcd(p,q)=1]$$

$$
\begin{aligned}
S(N) &= \sum_{i=1}^{N}\sum_{p=1}^{i}\sum_{q=p+1}^{i}\frac{1}{pq}[p+q \geq i][gcd(p,q)=1] \\
&= \sum_{i=1}^{N}\sum_{p=1}^{i}\sum_{q=p+1}^{i}\frac{1}{pq}[p+q \geq i]\sum_{d|gcd(p,q)}\mu(d)
\end{aligned}
$$

$$
\begin{aligned}
S(N) &= \sum_{i=1}^{N}\sum_{p=1}^{i}\sum_{q=p+1}^{i}\frac{1}{pq}[p+q\geq i][gcd(p,q)=1] \\
&= \sum_{i=1}^{N}\sum_{p=1}^{i}\sum_{q=p+1}^{i}\frac{1}{pq}[p+q\geq i]\sum_{d|gcd(p,q)}\mu(d) \\
&= \sum_{i=1}^{N}\sum_{d=1}^{i}\mu(d)\sum_{p=1}^{\lfloor\frac{i}{d}\rfloor}\sum_{q=p+1}^{\lfloor\frac{i}{d}\rfloor}\frac{1}{pqd^2}[pd+qd\geq i]
\end{aligned}
$$

$$
\begin{aligned}
S(N) &= \sum_{i=1}^{N}\sum_{p=1}^{i}\sum_{q=p+1}^{i}\frac{1}{pq}[p+q\geq i][gcd(p,q)=1]\\
&= \sum_{i=1}^{N}\sum_{p=1}^{i}\sum_{q=p+1}^{i}\frac{1}{pq}[p+q\geq i]\sum_{d|gcd(p,q)}\mu(d)\\
&= \sum_{i=1}^{N}\sum_{d=1}^{i}\mu(d)\sum_{p=1}^{\lfloor\frac{i}{d}\rfloor}\sum_{q=p+1}^{\lfloor\frac{i}{d}\rfloor}\frac{1}{pqd^2}[pd+qd\geq i]\\
&= \sum_{i=1}^{N}\sum_{d=1}^{i}\frac{\mu(d)}{d^2}\sum_{p=1}^{\lfloor\frac{i}{d}\rfloor}\sum_{q=p+1}^{\lfloor\frac{i}{d}\rfloor}\Bigg(\frac{1}{pq}[p+q\geq\lfloor\frac{i}{d}\rfloor]\\
&\quad -\frac{1}{pq}[p+q=\lfloor\frac{i}{d}\rfloor][i \bmod d\neq 0]\Bigg)
\end{aligned}
$$

The inverse summation of coprime couples

Project Euler 441

$$S(N) = \sum_{i=1}^{N} \sum_{d=1}^{i} \frac{\mu(d)}{d^2} \left(T(\lfloor \frac{i}{d} \rfloor) - G(\lfloor \frac{i}{d} \rfloor)[i \ mod \ d \neq 0] \right)$$

$$
\begin{aligned}
S(N) &= \sum_{i=1}^{N}\sum_{d=1}^{i}\frac{\mu(d)}{d^2}\left(T(\lfloor\frac{i}{d}\rfloor) - G(\lfloor\frac{i}{d}\rfloor)[i \ mod \ d \neq 0]\right) \\
&= \sum_{d=1}^{N}\frac{\mu(d)}{d^2}\sum_{i=d}^{N}\left(T(\lfloor\frac{i}{d}\rfloor) - G(\lfloor\frac{i}{d}\rfloor)[i \ mod \ d \neq 0]\right)
\end{aligned}
$$

$$\begin{aligned} S(N) &= \sum_{i=1}^{N}\sum_{d=1}^{i}\frac{\mu(d)}{d^2}\left(T(\lfloor\frac{i}{d}\rfloor) - G(\lfloor\frac{i}{d}\rfloor)[i \ mod \ d \neq 0]\right) \\ &= \sum_{d=1}^{N}\frac{\mu(d)}{d^2}\sum_{i=d}^{N}\left(T(\lfloor\frac{i}{d}\rfloor) - G(\lfloor\frac{i}{d}\rfloor)[i \ mod \ d \neq 0]\right) \end{aligned}$$

- 按$\lfloor\frac{i}{d}\rfloor$的值分段计算，复杂度$O(n\ln n)$。

- 经过一些复杂的分类讨论可以做到$O(n)$。

- 经过一些复杂的分类讨论可以做到$O(n)$。
- 还有一个奥妙重重的做法：

Sun, 20 Oct 2013, 19:15
plamenko
C/C++

$O(N \log\log N)$ for precomputing Möbius function
$O(N)$ for precomputing Harmonic numbers
$O(N)$ for overall sum after that

$$S(N) = \frac{1}{2}\left(N - 3 + \sum_{g=1}^{N} \mu(g) \frac{1}{g^2} H(\left\lfloor \frac{N}{g} \right\rfloor)^2\right), \quad N \ge 2$$

0.6s in C++

一个实用技巧

杜教筛

设$f(n)$为一个数论函数，$S(n)=\sum_{i=1}^{n} f(i)$。
考虑再找到一个合适的数论函数$g(n)$。

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

可以得到一个$S(n)$的递推式

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

计算$\mu(n)$之和

- 考虑计算$S(n)=\sum_{i=1}^{n}\mu(i)$。

计算$\mu(n)$之和

- 考虑计算$S(n)=\sum_{i=1}^{n}\mu(i)$。
- 由于$\mu * 1=\epsilon$，所以设$g(n)=1$，那么

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

计算$\mu(n)$之和

- 考虑计算$S(n)=\sum_{i=1}^{n}\mu(i)$。
- 由于$\mu*1=\epsilon$，所以设$g(n)=1$，那么

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

- 由于$\lfloor\frac{\lfloor\frac{x}{a}\rfloor}{b}\rfloor=\lfloor\frac{x}{ab}\rfloor$，所以递推时$\lfloor\frac{n}{i}\rfloor$的取值只有$O(\sqrt{n})$种。

计算$\mu(n)$之和

- 考虑计算$S(n)=\sum_{i=1}^{n}\mu(i)$。
- 由于$\mu * 1=\epsilon$，所以设$g(n)=1$，那么

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

- 由于$\lfloor\frac{\lfloor\frac{x}{a}\rfloor}{b}\rfloor=\lfloor\frac{x}{ab}\rfloor$，所以递推时$\lfloor\frac{n}{i}\rfloor$的取值只有$O(\sqrt{n})$种。
- 设$m=\lfloor\sqrt{n}\rfloor$，那么这些取值为

$$1,2,3,\ldots,m-1,m,\lfloor\frac{n}{m}\rfloor,\lfloor\frac{n}{m-1}\rfloor,\ldots,\lfloor\frac{n}{2}\rfloor,\lfloor\frac{n}{1}\rfloor$$

计算$\mu(n)$之和

- 直接计算的复杂度为

$$\sum_{i=1}^{\lfloor\sqrt{n}\rfloor} O(\sqrt{i}) + \sum_{i=1}^{\lfloor\sqrt{n}\rfloor} O\left(\sqrt{\frac{n}{i}}\right)$$

计算$\mu(n)$之和

- 直接计算的复杂度为

$$\sum_{i=1}^{\lfloor\sqrt{n}\rfloor} O(\sqrt{i}) + \sum_{i=1}^{\lfloor\sqrt{n}\rfloor} O\left(\sqrt{\frac{n}{i}}\right)$$

- 只需要考虑后半部分

$$\sum_{i=1}^{\lfloor\sqrt{n}\rfloor} O\left(\sqrt{\frac{n}{i}}\right) \approx O(n^{\frac{3}{4}})$$

计算$\mu(n)$之和

- 直接计算的复杂度为

$$\sum_{i=1}^{\lfloor\sqrt{n}\rfloor} O(\sqrt{i}) + \sum_{i=1}^{\lfloor\sqrt{n}\rfloor} O\left(\sqrt{\frac{n}{i}}\right)$$

- 只需要考虑后半部分

$$\sum_{i=1}^{\lfloor\sqrt{n}\rfloor} O\left(\sqrt{\frac{n}{i}}\right) \approx O(n^{\frac{3}{4}})$$

- 利用积性预处理前$n^{\frac{2}{3}}$项，可以将复杂度降到$O(n^{\frac{2}{3}})$。

Counting Divisors (square)
SPOJ DIVCNT2

题目大意

设$\sigma_0(n)$为n的约数个数，

$$S_2(n)=\sum_{i=1}^{n}\sigma_0(i^2)$$

求$S_2(n)$。

数据范围

$1\le n\le 10^{12}$，20s，1536MB。

Counting Divisors (square)
SPOJ DIVCNT2

- 设$\omega(n)$为n的不同质因子个数，有

$$\sigma_0(n^2) = \sum_{d|n} 2^{\omega(d)}$$

- 设$\omega(n)$为n的不同质因子个数，有

$$\sigma_0(n^2) = \sum_{d|n} 2^{\omega(d)}$$

- 考虑n的每个约数d，可以在d^2中去掉d的一个质因子集合，也就是$2^{\omega(d)}$种。

数论函数选讲

任之洲

数论函数
PE512
PE530
PE441
杜教筛
DIVCNT2
PE439
积性函数求和
DIVCNT3

- 设$\omega(n)$为n的不同质因子个数，有

$$\sigma_0(n^2) = \sum_{d|n} 2^{\omega(d)}$$

- 考虑n的每个约数d，可以在d^2中去掉d的一个质因子集合，也就是$2^{\omega(d)}$种。
  - 这样可以枚举出n^2的所有约数。

Counting Divisors (square)
SPOJ DIVCNT2

- 设$\omega(n)$为n的不同质因子个数，有

$$\sigma_0(n^2) = \sum_{d|n} 2^{\omega(d)}$$

- 考虑n的每个约数d，可以在d^2中去掉d的一个质因子集合，也就是$2^{\omega(d)}$种。
  - 这样可以枚举出n^2的所有约数。
- 于是，有

$$\sum_{i=1}^{n} \sigma_0(i^2) = \sum_{i=1}^{n} 2^{\omega(i)} \left\lfloor \frac{n}{i} \right\rfloor$$

- 设$\omega(n)$为n的不同质因子个数，有

$$\sigma_0(n^2) = \sum_{d|n} 2^{\omega(d)}$$

- 考虑n的每个约数d，可以在d^2中去掉d的一个质因子集合，也就是$2^{\omega(d)}$种。
  - 这样可以枚举出n^2的所有约数。
- 于是，有

$$\sum_{i=1}^{n} \sigma_0(i^2) = \sum_{i=1}^{n} 2^{\omega(i)} \left\lfloor \frac{n}{i} \right\rfloor$$

- 可以按$\left\lfloor \frac{n}{i} \right\rfloor$分段计算。

Counting Divisors (square)

SPOJ DIVCNT2

- 现在需要计算的是$2^{\omega(i)}$之和，$2^{\omega(n)}$相当于n的无平方因子的约数个数。

Counting Divisors (square)

SPOJ DIVCNT2

- 现在需要计算的是$2^{\omega(i)}$之和，$2^{\omega(n)}$相当于n的无平方因子的约数个数。
- 所以，有

$$2^{\omega(n)} = \sum_{d|n} \mu^2(d)$$

Counting Divisors (square)
SPOJ DIVCNT2

- 现在需要计算的是$2^{\omega(i)}$之和，$2^{\omega(n)}$相当于n的无平方因子的约数个数。
- 所以，有

$$2^{\omega(n)} = \sum_{d|n} \mu^2(d)$$

- 即

$$\sum_{i=1}^{n} 2^{\omega(i)} = \sum_{i=1}^{n} \mu^2(i) \left\lfloor \frac{n}{i} \right\rfloor$$

- 现在需要计算的是$2^{\omega(i)}$之和，$2^{\omega(n)}$相当于n的无平方因子的约数个数。
- 所以，有

$$2^{\omega(n)} = \sum_{d|n} \mu^2(d)$$

- 即

$$\sum_{i=1}^{n} 2^{\omega(i)} = \sum_{i=1}^{n} \mu^2(i) \left\lfloor \frac{n}{i} \right\rfloor$$

- $\sum_{i=1}^{n} \mu^2(i)$可以这样暴力计算

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

Counting Divisors (square)
SPOJ DIVCNT2

- 整理一下

- 整理一下

$$\sum_{i=1}^{n} \sigma_0(i^2) = \sum_{i=1}^{n} 2^{\omega(i)} \left\lfloor \frac{n}{i} \right\rfloor$$

- 整理一下

$$\sum_{i=1}^{n} \sigma_0(i^2) = \sum_{i=1}^{n} 2^{\omega(i)} \left\lfloor \frac{n}{i} \right\rfloor$$

$$\sum_{i=1}^{n} 2^{\omega(i)} = \sum_{i=1}^{n} \mu^2(i) \left\lfloor \frac{n}{i} \right\rfloor$$

Counting Divisors (square)
SPOJ DIVCNT2

- 整理一下

$$\begin{aligned}
\sum_{i=1}^{n} \sigma_0(i^2) &= \sum_{i=1}^{n} 2^{\omega(i)} \left\lfloor \frac{n}{i} \right\rfloor \\
\sum_{i=1}^{n} 2^{\omega(i)} &= \sum_{i=1}^{n} \mu^2(i) \left\lfloor \frac{n}{i} \right\rfloor \\
\sum_{i=1}^{n} \mu^2(i) &= \sum_{i=1}^{\lfloor \sqrt{n} \rfloor} \mu(i) \left\lfloor \frac{n}{i^2} \right\rfloor
\end{aligned}$$

- 整理一下

$$\sum_{i=1}^{n}\sigma_0(i^2) = \sum_{i=1}^{n}2^{\omega(i)}\left\lfloor\frac{n}{i}\right\rfloor$$

$$\sum_{i=1}^{n}2^{\omega(i)} = \sum_{i=1}^{n}\mu^2(i)\left\lfloor\frac{n}{i}\right\rfloor$$

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

- 用线性筛预处理$n^{\frac{2}{3}}$以内的函数值，复杂度$O(n^{\frac{2}{3}})$

题目大意

设$\sigma(n)$为n的所有约数之和，

$$S(n) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sigma(ij)$$

如$S(1000) = 563576517282$。
求$S(10^{11})\ mod\ 10^9$。

Sum of sum of divisors
Project Euler 439

$$S(n) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k|ij} k$$

$$
\begin{aligned}
S(n) &= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k|ij} k \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{u|i}\sum_{v|j} uv\epsilon(gcd(v,\frac{i}{u}))
\end{aligned}
$$

$$
\begin{aligned}
S(n) &= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k|ij} k \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{u|i}\sum_{v|j} uv\epsilon(gcd(v,\frac{i}{u})) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{u|i}\sum_{v|j} uv \sum_{d|gcd(v,\frac{i}{u})} \mu(d)
\end{aligned}
$$

数论函数选讲

任之洲

数论函数

PE512

PE530

PE441

杜教筛

DIVCNT2

PE439

积性函数求和

DIVCNT3

$$
\begin{aligned}
S(n) &= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k|ij} k \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{u|i}\sum_{v|j} uv\epsilon(gcd(v,\frac{i}{u})) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{u|i}\sum_{v|j} uv \sum_{d|gcd(v,\frac{i}{u})} \mu(d) \\
&= \sum_{d=1}^{n}\mu(d)\sum_{u=1}^{\lfloor\frac{n}{d}\rfloor}\sum_{v=1}^{\lfloor\frac{n}{d}\rfloor} u(vd)\left\lfloor\frac{n}{ud}\right\rfloor\left\lfloor\frac{n}{vd}\right\rfloor
\end{aligned}
$$

Sum of sum of divisors
Project Euler 439

$$
\begin{aligned}
S(n) &= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k|ij} k \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{u|i}\sum_{v|j} uv\epsilon(gcd(v, \frac{i}{u})) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{u|i}\sum_{v|j} uv \sum_{d|gcd(v,\frac{i}{u})} \mu(d) \\
&= \sum_{d=1}^{n}\mu(d)\sum_{u=1}^{\lfloor\frac{n}{d}\rfloor}\sum_{v=1}^{\lfloor\frac{n}{d}\rfloor} u(vd)\left\lfloor\frac{n}{ud}\right\rfloor\left\lfloor\frac{n}{vd}\right\rfloor \\
&= \sum_{d=1}^{n} d\mu(d)\left(\sum_{i=1}^{\lfloor\frac{n}{d}\rfloor}\sigma(i)\right)^2
\end{aligned}
$$

Sum of sum of divisors

Project Euler 439

- $\sum_{i=1}^{n} \sigma(i)$可以$O(\sqrt{n})$暴力计算

$$\sum_{i=1}^{n} \sigma(i) = \sum_{i=1}^{n} i \left\lfloor \frac{n}{i} \right\rfloor$$

Sum of sum of divisors
Project Euler 439

- $\sum_{i=1}^{n}\sigma(i)$可以$O(\sqrt{n})$暴力计算

$$\sum_{i=1}^{n}\sigma(i)=\sum_{i=1}^{n}i\left\lfloor\frac{n}{i}\right\rfloor$$

- 线性筛预处理前$n^{\frac{2}{3}}$项后，这一部分复杂度为$O(n^{\frac{2}{3}})$。

Sum of sum of divisors
Project Euler 439

- $\sum_{i=1}^{n}\sigma(i)$可以$O(\sqrt{n})$暴力计算

$$\sum_{i=1}^{n}\sigma(i)=\sum_{i=1}^{n} i\left\lfloor\frac{n}{i}\right\rfloor$$

- 线性筛预处理前$n^{\frac{2}{3}}$项后，这一部分复杂度为$O(n^{\frac{2}{3}})$。
- 还需要计算$\sum_{i=1}^{n} i\mu(i)$。

- $\sum_{i=1}^{n}\sigma(i)$可以$O(\sqrt{n})$暴力计算

$$\sum_{i=1}^{n}\sigma(i)=\sum_{i=1}^{n} i\left\lfloor\frac{n}{i}\right\rfloor$$

- 线性筛预处理前$n^{\frac{2}{3}}$项后，这一部分复杂度为$O(n^{\frac{2}{3}})$。
- 还需要计算$\sum_{i=1}^{n} i\mu(i)$。
- 考虑如下等式

$$\begin{aligned} f(n) &= \sum_{k=1}^{n} k\mu(k) g\left(\left\lfloor\frac{n}{k}\right\rfloor\right) \\ \Leftrightarrow \quad g(n) &= \sum_{k=1}^{n} kf\left(\left\lfloor\frac{n}{k}\right\rfloor\right) \end{aligned}$$

- 证明

$$g(n) = \sum_{k=1}^{n} kf\left(\left\lfloor \frac{n}{k} \right\rfloor\right)$$

- 证明

$$\begin{aligned} g(n) &= \sum_{k=1}^{n} kf\left(\left\lfloor\frac{n}{k}\right\rfloor\right) \\ &= \sum_{k=1}^{n} k\sum_{i=1}^{\left\lfloor\frac{n}{k}\right\rfloor} i\mu(i)g\left(\left\lfloor\frac{n}{ki}\right\rfloor\right) \end{aligned}$$

- 证明

$$
\begin{aligned}
g(n) &= \sum_{k=1}^{n} kf\left(\left\lfloor \frac{n}{k} \right\rfloor\right) \\
&= \sum_{k=1}^{n} k \sum_{i=1}^{\lfloor \frac{n}{k} \rfloor} i\mu(i) g\left(\left\lfloor \frac{n}{ki} \right\rfloor\right) \\
&= \sum_{j=1}^{n} g\left(\left\lfloor \frac{n}{j} \right\rfloor\right) \sum_{i|j} \mu(i) i \frac{j}{i}
\end{aligned}
$$

- 证明

$$
\begin{aligned}
g(n) &= \sum_{k=1}^{n} kf\left(\left\lfloor \frac{n}{k} \right\rfloor\right) \\
&= \sum_{k=1}^{n} k \sum_{i=1}^{\lfloor \frac{n}{k} \rfloor} i\mu(i) g\left(\left\lfloor \frac{n}{ki} \right\rfloor\right) \\
&= \sum_{j=1}^{n} g\left(\left\lfloor \frac{n}{j} \right\rfloor\right) \sum_{i|j} \mu(i) i \frac{j}{i} \\
&= \sum_{j=1}^{n} g\left(\left\lfloor \frac{n}{j} \right\rfloor\right) \epsilon(j) j
\end{aligned}
$$

- 证明

$$
\begin{aligned}
g(n) &= \sum_{k=1}^{n} kf\left(\left\lfloor\frac{n}{k}\right\rfloor\right) \\
&= \sum_{k=1}^{n} k \sum_{i=1}^{\lfloor\frac{n}{k}\rfloor} i\mu(i)g\left(\left\lfloor\frac{n}{ki}\right\rfloor\right) \\
&= \sum_{j=1}^{n} g\left(\left\lfloor\frac{n}{j}\right\rfloor\right) \sum_{i|j} \mu(i)i\frac{j}{i} \\
&= \sum_{j=1}^{n} g\left(\left\lfloor\frac{n}{j}\right\rfloor\right) \epsilon(j)j \\
&= g(n)
\end{aligned}
$$

- 证明

$$\begin{aligned} g(n) &= \sum_{k=1}^{n} kf\left(\left\lfloor\frac{n}{k}\right\rfloor\right) \\ &= \sum_{k=1}^{n} k \sum_{i=1}^{\lfloor\frac{n}{k}\rfloor} i\mu(i)g\left(\left\lfloor\frac{n}{ki}\right\rfloor\right) \\ &= \sum_{j=1}^{n} g\left(\left\lfloor\frac{n}{j}\right\rfloor\right) \sum_{i|j} \mu(i) i \frac{j}{i} \\ &= \sum_{j=1}^{n} g\left(\left\lfloor\frac{n}{j}\right\rfloor\right) \epsilon(j) j \\ &= g(n) \end{aligned}$$

- 反之亦然。

- 设

$$f(n) = \sum_{k=1}^{n} k\mu(k)$$

- 设

$$f(n) = \sum_{k=1}^{n} k\mu(k)$$

- 那么

$$\sum_{k=1}^{n} kf(k) = 1$$

- 设

$$f(n) = \sum_{k=1}^{n} k\mu(k)$$

- 那么

$$\sum_{k=1}^{n} kf(k) = 1$$

- 可以得到一个$f(n)$的递推式

$$f(n) = 1 - \sum_{k=2}^{n} kf(k)$$

- 设

$$f(n)=\sum_{k=1}^{n}k\mu(k)$$

- 那么

$$\sum_{k=1}^{n}kf(k)=1$$

- 可以得到一个$f(n)$的递推式

$$f(n)=1-\sum_{k=2}^{n}kf(k)$$

- $f(n)$也可以$O(n^{\frac{2}{3}})$完成计算，总复杂度$O(n^{\frac{2}{3}})$。

积性函数求和

- 设$F(n)$为一个积性函数，n的质因子分解为$\prod_{i=1}^{k} {p_i}^{c_i}$。

积性函数求和

- 设$F(n)$为一个积性函数，n的质因子分解为$\prod_{i=1}^{k} p_i^{c_i}$。
- 考虑这样来描述一个积性函数：

积性函数求和

- 设$F(n)$为一个积性函数，n的质因子分解为$\prod_{i=1}^{k} {p_i}^{c_i}$。
- 考虑这样来描述一个积性函数：
  - 当n为质数时，即$n = p$，$F(p) = G(p)$。

积性函数求和

- 设$F(n)$为一个积性函数，n的质因子分解为$\prod_{i=1}^{k} {p_i}^{c_i}$。
- 考虑这样来描述一个积性函数：
  - 当n为质数时，即$n = p$，$F(p) = G(p)$。
  - 当n为质数的幂时，即$n = p^c$且$c > 1$，$F(p^c) = T(p^c)$。

积性函数求和

- 设$F(n)$为一个积性函数，n的质因子分解为$\prod_{i=1}^{k} {p_i}^{c_i}$。
- 考虑这样来描述一个积性函数：
  - 当n为质数时，即$n = p$，$F(p) = G(p)$。
  - 当n为质数的幂时，即$n = p^c$且$c > 1$，$F(p^c) = T(p^c)$。
  - 剩下的情况根据积性，$F(n) = \prod_{i=1}^{k} F({p_i}^{c_i})$。

积性函数求和

- 设$F(n)$为一个积性函数，n的质因子分解为$\prod_{i=1}^{k} {p_i}^{c_i}$。
- 考虑这样来描述一个积性函数：
  - 当n为质数时，即$n = p$，$F(p) = G(p)$。
  - 当n为质数的幂时，即$n = p^c$且$c > 1$，$F(p^c) = T(p^c)$。
  - 剩下的情况根据积性，$F(n) = \prod_{i=1}^{k} F({p_i}^{c_i})$。
- $\varphi(n)$中，$G(p) = p - 1$，$T(p^c) = (p - 1)p^{c-1}$。

积性函数求和

- 设$F(n)$为一个积性函数，n的质因子分解为$\prod_{i=1}^{k} {p_i}^{c_i}$。
- 考虑这样来描述一个积性函数：
  - 当n为质数时，即$n = p$，$F(p) = G(p)$。
  - 当n为质数的幂时，即$n = p^c$且$c > 1$，$F(p^c) = T(p^c)$。
  - 剩下的情况根据积性，$F(n) = \prod_{i=1}^{k} F({p_i}^{c_i})$。
- $\varphi(n)$中，$G(p) = p - 1$，$T(p^c) = (p - 1)p^{c-1}$。
- $\mu(n)$中，$G(p) = -1$，$T(p^c) = 0$。

积性函数求和

- 设$F(n)$为一个积性函数，n的质因子分解为$\prod_{i=1}^{k} {p_i}^{c_i}$。
- 考虑这样来描述一个积性函数：
  - 当n为质数时，即$n = p$，$F(p) = G(p)$。
  - 当n为质数的幂时，即$n = p^c$且$c > 1$，$F(p^c) = T(p^c)$。
  - 剩下的情况根据积性，$F(n) = \prod_{i=1}^{k} F({p_i}^{c_i})$。
- $\varphi(n)$中，$G(p) = p - 1$，$T(p^c) = (p - 1)p^{c-1}$。
- $\mu(n)$中，$G(p) = -1$，$T(p^c) = 0$。
- 当$G(p)$和$T(p^c)$是项数比较少的关于p, p^c的多项式时怎么做？

积性函数求和

一个简单的例子

定义一个欧拉函数的变种$\phi(n,d)$，设$n=\prod_{k=1}^{m} {p_k}^{c_k}$。
其中p_k为互不相同的质因子（$c_k>0$，即把n质因子分解），那么

$$\phi(n,d)=\prod_{k=1}^{m}({p_k}^{c_k}+d)$$

特别地，定义$\phi(1,d)=1$。
对于给定的n,d，求

$$\left(\sum_{i=1}^{n}\phi(i,d)\right)\ mod\ 10^9+7$$

积性函数求和

- 在这个函数中

$$
\begin{aligned}
G(p) &= p + d \\
T(p^c) &= p^c + d
\end{aligned}
$$

积性函数求和

- 在这个函数中

$$\begin{aligned} G(p) &= p+d \\ T(p^c) &= p^c+d \end{aligned}$$

- 对于一个正整数$x(\leq n)$。

积性函数求和

- 在这个函数中

$$\begin{aligned} G(p) &= p + d \\ T(p^c) &= p^c + d \end{aligned}$$

- 对于一个正整数$x(\leq n)$。
  - 最多拥有一个$> \sqrt{n}$的质因子。

积性函数求和

- 在这个函数中

$$\begin{aligned} G(p) &= p + d \\ T(p^c) &= p^c + d \end{aligned}$$

- 对于一个正整数$x(\leq n)$。
  - 最多拥有一个$> \sqrt{n}$的质因子。
  - 如果存在，则幂次一定为1。

积性函数求和

- 在这个函数中

$$\begin{aligned} G(p) &= p+d \\ T(p^c) &= p^c+d \end{aligned}$$

- 对于一个正整数$x(\leq n)$。
  - 最多拥有一个$>\sqrt{n}$的质因子。
  - 如果存在，则幂次一定为1。
- 所以，有

$$\sum_{i=1}^{n} F(i) = \sum_{\substack{x \leq n \\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x)\left(1+\sum_{\substack{\sqrt{n} < p \leq \lfloor \frac{n}{x} \rfloor \\ p\text{为质数}}} G(p)\right)$$

积性函数求和

$$\sum_{i=1}^{n} F(i) = \sum_{\substack{x \leq n \\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x)\left(1 + \sum_{\substack{\sqrt{n} < p \leq \lfloor \frac{n}{x} \rfloor \\ p\text{为质数}}} G(p)\right)$$

积性函数求和

$$\sum_{i=1}^{n} F(i) = \sum_{\substack{x \le n \\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x)\left(1 + \sum_{\substack{\sqrt{n} < p \le \lfloor \frac{n}{x} \rfloor \\ p\text{为质数}}} G(p)\right)$$

- 分为两部分计算

积性函数求和

$$\sum_{i=1}^{n} F(i) = \sum_{\substack{x \le n \\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x) \left(1 + \sum_{\substack{\sqrt{n} < p \le \lfloor \frac{n}{x} \rfloor \\ p\text{为质数}}} G(p) \right)$$

- 分为两部分计算

$$\sum_{\substack{\sqrt{n} < p \le \lfloor \frac{n}{x} \rfloor \\ p\text{为质数}}} G(p)$$

积性函数求和

$$\sum_{i=1}^{n} F(i) = \sum_{\substack{x \le n \\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x)\left(1 + \sum_{\substack{\sqrt{n} < p \le \lfloor \frac{n}{x} \rfloor \\ p\text{为质数}}} G(p)\right)$$

- 分为两部分计算

$$\sum_{\substack{\sqrt{n} < p \le \lfloor \frac{n}{x} \rfloor \\ p\text{为质数}}} G(p)$$

$$\sum_{\substack{x \le n \\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x)$$

$G(p)$的计算

- 需要对每种$\left\lfloor \frac{n}{x} \right\rfloor$计算

$$\sum_{\substack{\sqrt{n} < p \le \lfloor \frac{n}{x} \rfloor \\ p\text{为质数}}} G(p)$$

$G(p)$的计算

- 需要对每种$\left\lfloor \frac{n}{x} \right\rfloor$计算

$$\sum_{\substack{\sqrt{n} < p \le \lfloor \frac{n}{x} \rfloor \\ p\text{为质数}}} G(p)$$

- 相当于求$\le \left\lfloor \frac{n}{x} \right\rfloor$的质数和及质数个数。

$G(p)$的计算

- 设$p_1 \sim p_m$为$\leq \sqrt{n}$的质数，且$p_i < p_{i+1}$。

$G(p)$的计算

- 设$p_1 \sim p_m$为$\le \sqrt{n}$的质数，且$p_i < p_{i+1}$。
- 设$P_k[i][j]$为$[1, j]$范围内与前i个质数互质的数的k次方之和。

$G(p)$的计算

- 设$p_1 \sim p_m$为$\leq \sqrt{n}$的质数，且$p_i < p_{i+1}$。
- 设$P_k[i][j]$为$[1, j]$范围内与前i个质数互质的数的k次方之和。
  - 需要求的即为$P_0[m][j]$和$P_1[m][j]$。

$G(p)$的计算

- 设$p_1 \sim p_m$为$\le \sqrt{n}$的质数，且$p_i < p_{i+1}$。
- 设$P_k[i][j]$为$[1,j]$范围内与前i个质数互质的数的k次方之和。
  - 需要求的即为$P_0[m][j]$和$P_1[m][j]$。
- 设$l = \lfloor \frac{j}{p_i} \rfloor$，容易得到以下递推式

$$P_k[i][j] = P_k[i-1][j] - {p_i}^k P_k[i-1][l]$$

$G(p)$的计算

- 设$p_1 \sim p_m$为$\le \sqrt{n}$的质数，且$p_i < p_{i+1}$。
- 设$P_k[i][j]$为$[1,j]$范围内与前i个质数互质的数的k次方之和。
  - 需要求的即为$P_0[m][j]$和$P_1[m][j]$。
- 设$l = \lfloor \frac{j}{p_i} \rfloor$，容易得到以下递推式

$$P_k[i][j] = P_k[i-1][j] - {p_i}^k P_k[i-1][l]$$

  - 即去掉$1 \sim j$中还没有被筛掉的p_i的倍数。

$G(p)$的计算

- 设$p_1 \sim p_m$为$\leq \sqrt{n}$的质数，且$p_i < p_{i+1}$。
- 设$P_k[i][j]$为$[1,j]$范围内与前i个质数互质的数的k次方之和。
  - 需要求的即为$P_0[m][j]$和$P_1[m][j]$。
- 设$l = \lfloor \frac{j}{p_i} \rfloor$，容易得到以下递推式

$$P_k[i][j] = P_k[i-1][j] - {p_i}^k P_k[i-1][l]$$

  - 即去掉$1 \sim j$中还没有被筛掉的p_i的倍数。
  - 递推时只需要计算$\lfloor \frac{n}{x} \rfloor$这些状态就能完成转移。

$G(p)$的计算

- 设$p_1 \sim p_m$为$\leq \sqrt{n}$的质数，且$p_i < p_{i+1}$。
- 设$P_k[i][j]$为$[1,j]$范围内与前i个质数互质的数的k次方之和。
  - 需要求的即为$P_0[m][j]$和$P_1[m][j]$。
- 设$l = \lfloor \frac{j}{p_i} \rfloor$，容易得到以下递推式

$$P_k[i][j] = P_k[i-1][j] - {p_i}^k P_k[i-1][l]$$

  - 即去掉$1 \sim j$中还没有被筛掉的p_i的倍数。
  - 递推时只需要计算$\lfloor \frac{n}{x} \rfloor$这些状态就能完成转移。
  - 递推完成后，也就计算出了我们需要所有值。

$G(p)$的计算

- 估计暴力递推的复杂度

$$P_k[i][j] = P_k[i-1][j] - {p_i}^k P_k[i-1][l]$$

$G(p)$的计算

- 估计暴力递推的复杂度

$$P_k[i][j] = P_k[i-1][j] - {p_i}^k P_k[i-1][l]$$

- 考虑对于每个$\lfloor \frac{n}{x} \rfloor$计算有多少质数需要转移。

$G(p)$的计算

- 估计暴力递推的复杂度

$$P_k[i][j] = P_k[i-1][j] - {p_i}^k P_k[i-1][l]$$

- 考虑对于每个$\lfloor \frac{n}{x} \rfloor$计算有多少质数需要转移。
  - 设质数密度为$O(\frac{1}{\log n})$。

$G(p)$的计算

- 估计暴力递推的复杂度

$$P_k[i][j] = P_k[i-1][j] - {p_i}^k P_k[i-1][l]$$

- 考虑对于每个$\lfloor \frac{n}{x} \rfloor$计算有多少质数需要转移。
  - 设质数密度为$O(\frac{1}{\log n})$。
  - 那么计算量可估计为

$$\sum_{i=1}^{\lfloor\sqrt{n}\rfloor} \frac{i}{\log i} + \sum_{i=1}^{\lfloor\sqrt{n}\rfloor} \frac{\lfloor min(\frac{n}{i}, \sqrt{n}) \rfloor}{\log n} \approx O\left(\frac{n}{\log n}\right)$$

$G(p)$的计算

- 当$p_{i+1} > j$时，一定满足$P_k[i][j] = 1$。

$G(p)$的计算

- 当$p_{i+1} > j$时，一定满足$P_k[i][j] = 1$。
- 所以，当${p_i}^2 > j \geq p_i$时

$$P_k[i][j] = P_k[i-1][j] - {p_i}^k$$

$G(p)$的计算

- 当$p_{i+1} > j$时，一定满足$P_k[i][j] = 1$。
- 所以，当${p_i}^2 > j \geq p_i$时

$$P_k[i][j] = P_k[i-1][j] - {p_i}^k$$

- 因此，不必重新计算${p_i}^2 > j$的情况。

$G(p)$的计算

- 当$p_{i+1} > j$时，一定满足$P_k[i][j] = 1$。
- 所以，当$p_i{}^2 > j \geq p_i$时

$$P_k[i][j] = P_k[i-1][j] - p_i{}^k$$

- 因此，不必重新计算$p_i{}^2 > j$的情况。
  - 记录j最近一次被更新时的i的值，设为pre_j。

$G(p)$的计算

- 当$p_{i+1} > j$时，一定满足$P_k[i][j] = 1$。
- 所以，当${p_i}^2 > j \geq p_i$时

$$P_k[i][j] = P_k[i-1][j] - {p_i}^k$$

- 因此，不必重新计算${p_i}^2 > j$的情况。
  - 记录j最近一次被更新时的i的值，设为pre_j。
  - 在调用$P_k[i][j]$时，将编号$pre_j + 1 \sim i$间的质数计入。

$G(p)$的计算

- 当$p_{i+1} > j$时，一定满足$P_k[i][j] = 1$。
- 所以，当${p_i}^2 > j \geq p_i$时

$$P_k[i][j] = P_k[i-1][j] - {p_i}^k$$

- 因此，不必重新计算${p_i}^2 > j$的情况。
  - 记录j最近一次被更新时的i的值，设为pre_j。
  - 在调用$P_k[i][j]$时，将编号$pre_j + 1 \sim i$间的质数计入。
- 考虑对于每个$\lfloor \frac{n}{x} \rfloor$计算有多少质数需要转移。

$G(p)$的计算

- 当$p_{i+1} > j$时，一定满足$P_k[i][j] = 1$。
- 所以，当${p_i}^2 > j \geq p_i$时

$$P_k[i][j] = P_k[i-1][j] - {p_i}^k$$

- 因此，不必重新计算${p_i}^2 > j$的情况。
  - 记录j最近一次被更新时的i的值，设为pre_j。
  - 在调用$P_k[i][j]$时，将编号$pre_j + 1 \sim i$间的质数计入。
- 考虑对于每个$\lfloor \frac{n}{x} \rfloor$计算有多少质数需要转移。
  - 那么计算量可估计为

$$\sum_{i=1}^{\lfloor\sqrt{n}\rfloor} \frac{\sqrt{i}}{\log i} + \sum_{i=1}^{\lfloor\sqrt{n}\rfloor} \frac{\lfloor\sqrt{\frac{n}{i}}\rfloor}{\log n} \approx O\left(\frac{n^{\frac{3}{4}}}{\log n}\right)$$

$F(x)$的计算

- 需要对每种$\left\lfloor \frac{n}{x} \right\rfloor$计算

$$\sum_{\substack{x \le n \\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x)$$

$F(x)$的计算

- 需要对每种$\left\lfloor \frac{n}{x} \right\rfloor$计算

$$\sum_{\substack{x \leq n \\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x)$$

- 由于$\left\lfloor \frac{n}{x} \right\rfloor$值相同的$F(x)$需要乘的系数是相同的，所以可以直接用$\left\lfloor \frac{n}{x} \right\rfloor$来表示状态。

$F(x)$的计算

- 需要对每种$\left\lfloor \frac{n}{x} \right\rfloor$计算

$$\sum_{\substack{x \le n \\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x)$$

- 由于$\left\lfloor \frac{n}{x} \right\rfloor$值相同的$F(x)$需要乘的系数是相同的，所以可以直接用$\left\lfloor \frac{n}{x} \right\rfloor$来表示状态。
  - 状态数$O(\sqrt{n})$。

$F(x)$的计算

- 需要对每种$\lfloor \frac{n}{x} \rfloor$计算

$$\sum_{\substack{x \le n \\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x)$$

- 由于$\lfloor \frac{n}{x} \rfloor$值相同的$F(x)$需要乘的系数是相同的，所以可以直接用$\lfloor \frac{n}{x} \rfloor$来表示状态。
  - 状态数$O(\sqrt{n})$。
  - 设$y = \lfloor \frac{n}{x} \rfloor$，那么转移$x \to xp$，等价于$y \to \lfloor \frac{y}{p} \rfloor$。

$F(x)$的计算

- 需要对每种$\lfloor \frac{n}{x} \rfloor$计算

$$\sum_{\substack{x \le n \\ x\text{没有大于}\sqrt{n}\text{质因子}}} F(x)$$

- 由于$\lfloor \frac{n}{x} \rfloor$值相同的$F(x)$需要乘的系数是相同的，所以可以直接用$\lfloor \frac{n}{x} \rfloor$来表示状态。
  - 状态数$O(\sqrt{n})$。
  - 设$y = \lfloor \frac{n}{x} \rfloor$，那么转移$x \to xp$，等价于$y \to \lfloor \frac{y}{p} \rfloor$。
- 枚举每一个不超过$\sqrt{n}$的质数进行转移。

$F(x)$的计算

- 同样考虑对于每个$\lfloor \frac{n}{x} \rfloor$计算有多少质数需要转移。

$F(x)$的计算

- 同样考虑对于每个$\lfloor \frac{n}{x} \rfloor$计算有多少质数需要转移。
  - 转移时需要枚举这种质因子的幂次，看上去计算量是这样的

$$\sum_{i=1}^{\lfloor \sqrt{n} \rfloor} \frac{i}{\log i} \log i + \sum_{i=1}^{\lfloor \sqrt{n} \rfloor} \frac{\lfloor min(\frac{n}{i}, \sqrt{n}) \rfloor}{\log n} \log n \approx O(n)$$

$F(x)$的计算

- 同样考虑对于每个$\lfloor \frac{n}{x} \rfloor$计算有多少质数需要转移。
  - 转移时需要枚举这种质因子的幂次，看上去计算量是这样的

$$\sum_{i=1}^{\lfloor\sqrt{n}\rfloor} \frac{i}{\log i} \log i + \sum_{i=1}^{\lfloor\sqrt{n}\rfloor} \frac{\lfloor min(\frac{n}{i}, \sqrt{n})\rfloor}{\log n} \log n \approx O(n)$$

  - 考虑有哪些质因子贡献1次、2次、3次...

$$h(n) = \sum_{2 \le i \le \log n} \lfloor n^{\frac{1}{i}} \rfloor \approx O(\sqrt{n})$$

$$\sum_{i=1}^{\lfloor\sqrt{n}\rfloor} \frac{i + h(i)}{\log i} + \sum_{i=1}^{\lfloor\sqrt{n}\rfloor} \frac{\lfloor min(\frac{n}{i}, \sqrt{n})\rfloor + h(\frac{n}{i})}{\log n} \approx O\left(\frac{n}{\log n}\right)$$

$F(x)$的计算

- 回想前一部分是如何优化的。

$F(x)$的计算

- 回想前一部分是如何优化的。
- 考虑如何省去${p_i}^2 > y = \lfloor \frac{n}{x} \rfloor$时的运算。

$F(x)$的计算

- 回想前一部分是如何优化的。
- 考虑如何省去${p_i}^2 > y = \lfloor \frac{n}{x} \rfloor$时的运算。
  - ${p_i}^2 > y \Rightarrow \lfloor \frac{y}{p_i} \rfloor < p_i \le \sqrt{n}$。

$F(x)$的计算

- 回想前一部分是如何优化的。
- 考虑如何省去${p_i}^2 > y = \lfloor \frac{n}{x} \rfloor$时的运算。
  - ${p_i}^2 > y \Rightarrow \lfloor \frac{y}{p_i} \rfloor < p_i \le \sqrt{n}$。
  - 当$\lfloor \frac{n}{x} \rfloor \le \sqrt{n}$时

$$1 + \sum_{\substack{\sqrt{n} < p \le \lfloor \frac{n}{x} \rfloor \\ p\text{为质数}}} G(p) = 1$$

$F(x)$的计算

- 回想前一部分是如何优化的。
- 考虑如何省去$p_i^2 > y = \lfloor \frac{n}{x} \rfloor$时的运算。
  - $p_i^2 > y \Rightarrow \lfloor \frac{y}{p_i} \rfloor < p_i \leq \sqrt{n}$。
  - 当$\lfloor \frac{n}{x} \rfloor \leq \sqrt{n}$时

$$1 + \sum_{\substack{\sqrt{n} < p \leq \lfloor \frac{n}{x} \rfloor \\ p\text{为质数}}} G(p) = 1$$

  - 于是不需要区分转移后的$\lfloor \frac{y}{p_i} \rfloor$具体是多少，只要计算对应的质数和累计入答案即可。

$F(x)$的计算

- 同样考虑对于每个$\lfloor \frac{n}{x} \rfloor$计算有多少质数需要转移。

$$h(n) = \sum_{2 \le i \le \log n} \lfloor n^{\frac{1}{i}} \rfloor \approx O(\sqrt{n})$$

$$\sum_{i=1}^{\lfloor \sqrt{n} \rfloor} \frac{h(i)}{\log i} + \sum_{i=1}^{\lfloor \sqrt{n} \rfloor} \frac{h(\frac{n}{i})}{\log n} \approx O\left(\frac{n^{\frac{3}{4}}}{\log n}\right)$$

Counting Divisors (cube)
SPOJ DIVCNT3

题目大意

设$\sigma_0(n)$为n的约数个数，

$$S_3(n) = \sum_{i=1}^{n} \sigma_0(i^3)$$

求$S_3(n)$。

数据范围

$1 \leq n \leq 10^{11}$，20s，1536MB。

Counting Divisors (cube)
SPOJ DIVCNT3

- 设$F(n)=\sigma_0(n^3)$，那么

$$\begin{aligned} G(p) &= 4 \\ T(p^c) &= 3c+1 \end{aligned}$$

Counting Divisors (cube)

SPOJ DIVCNT3

- 设$F(n) = \sigma_0(n^3)$，那么

$$
\begin{aligned}
G(p) &= 4 \\
T(p^c) &= 3c+1
\end{aligned}
$$

- 可以直接计算，复杂度$O(\frac{n^{\frac{3}{4}}}{\log n})$。

To Be Continued

id=31433449

题目大意

对于一张$2n$个点的无向完全图G。
要求将其拆为m个边集不相交的子图，其中第i张子图的每个点的度数都必须恰好为a_i。
构造一组可行解。

数据范围

$n \leq 100$，$m \leq 2n-1$，$\sum a_i = 2n-1$

题目大意

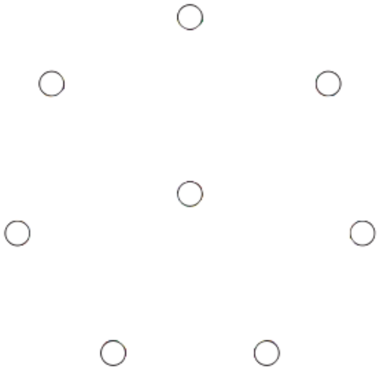
对于一张$2n$个点的无向完全图G。
要求将其拆为m个边集不相交的子图，其中第i张子图的每个点的度数都必须恰好为a_i。
构造一组可行解。

数据范围

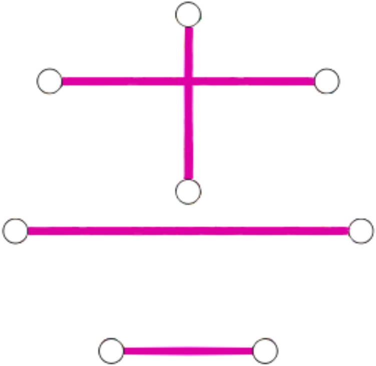
$n \le 100$，$m \le 2n-1$，$\sum a_i = 2n-1$

- 由于m可以取到$2n-1$，所以问题等价于$a_i = 1$。

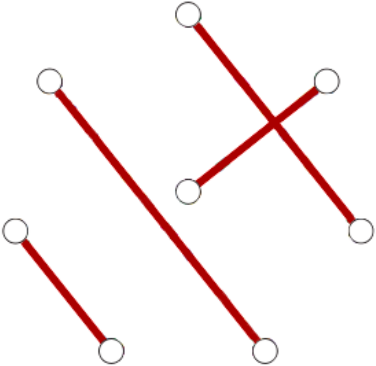
- 考虑将$2n-1$个点排成一个环，剩下一个点放中间。



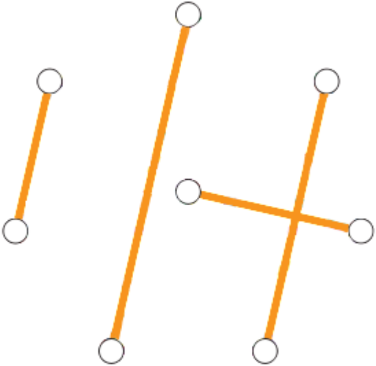
- 考虑将$2n-1$个点排成一个环，剩下一个点放中间。



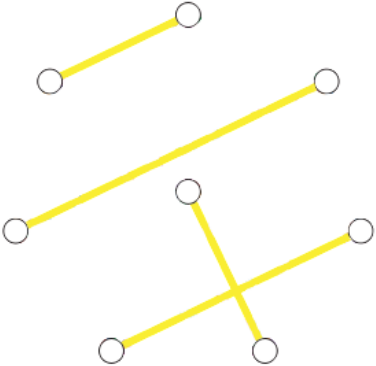
- 考虑将$2n-1$个点排成一个环，剩下一个点放中间。



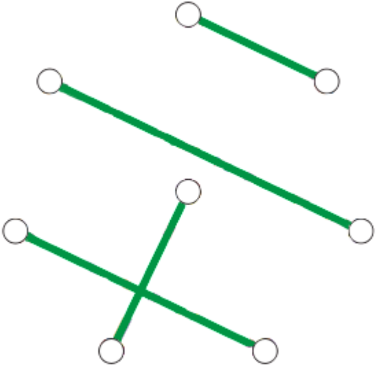
- 考虑将$2n-1$个点排成一个环，剩下一个点放中间。



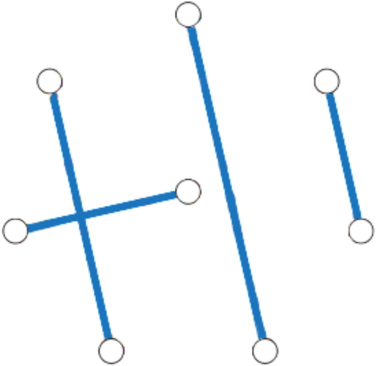
- 考虑将$2n-1$个点排成一个环，剩下一个点放中间。



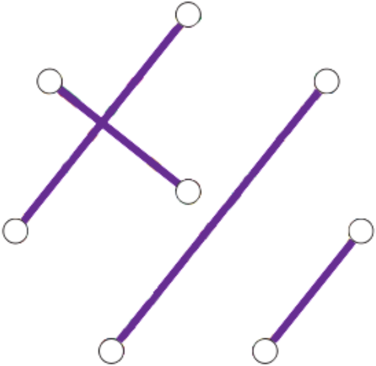
- 考虑将$2n-1$个点排成一个环，剩下一个点放中间。



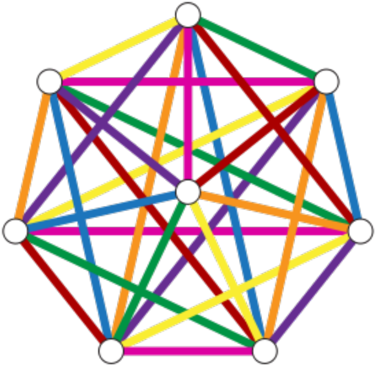
- 考虑将$2n-1$个点排成一个环，剩下一个点放中间。



- 考虑将$2n-1$个点排成一个环，剩下一个点放中间。



- 最终整张图会被划分成这样，问题解决。



Drunkard's Walk

Andrew Stankevich Contest 45 D

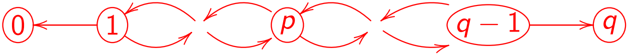
题目大意

构造一张n个点的图，除了点$n-1$和点n外均有两条出边。
一开始的位置在点1，每次等概率选择一条出边走，最终必须到达$n-1$或n，并且到$n-1$的概率为$\frac{p}{q}$。

数据范围

$1 \le p < q \le 100$
要求构造出的$n \le 1000$

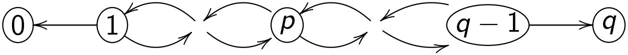
- 构造一条$q+1$个点的链



Drunkard's Walk

Andrew Stankevich Contest 45 D

- 构造一条$q+1$个点的链

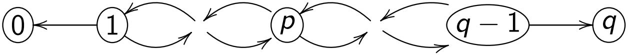


  - 起点设为p，两个终点分别为0和q。

Drunkard's Walk

Andrew Stankevich Contest 45 D

- 构造一条$q+1$个点的链

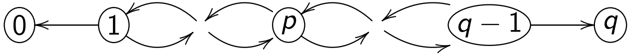


    - 起点设为p，两个终点分别为0和q。
- 设这条链上第$i(1 \le i < q)$个点最终走到0的概率为g_i。

Drunkard's Walk

Andrew Stankevich Contest 45 D

- 构造一条$q+1$个点的链



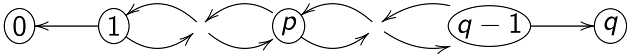
  - 起点设为p，两个终点分别为0和q。
- 设这条链上第$i(1 \le i < q)$个点最终走到0的概率为g_i。
- 那么有

$$g_i = \frac{g_{i-1} + g_{i+1}}{2}$$

Drunkard's Walk

Andrew Stankevich Contest 45 D

- 构造一条$q+1$个点的链



  - 起点设为p，两个终点分别为0和q。
- 设这条链上第$i(1 \le i < q)$个点最终走到0的概率为g_i。
- 那么有

$$g_i = \frac{g_{i-1} + g_{i+1}}{2}$$

- 易得$g_i = \frac{i}{q}$，$g_p = \frac{p}{q}$，问题解决。

题目大意

给出一个长度为n的串，要求构造两个的DFA自动机。

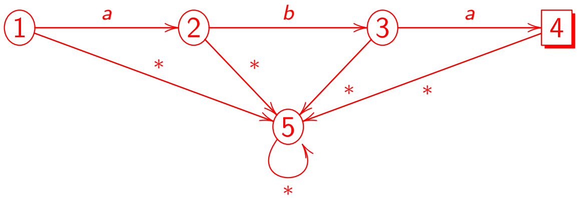
- 状态数不超过$n+1$。
- 1为起始态，可以自由确定若干接收态。
- 对于字符集（小写字母）中的每种字符都有对应转移。
- 这两个DFA自动机能接受的串的集合的交为给定串。

数据范围

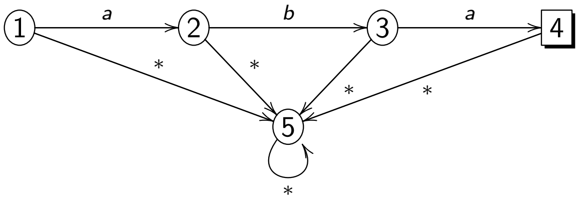
$1 \leq n \leq 50$

- 为了帮助理解题意，再给出一些说明。

- 为了帮助理解题意，再给出一些说明。
- 容易构造一个状态数为$n+2$的DFA自动机使得它只能接收给定的串。

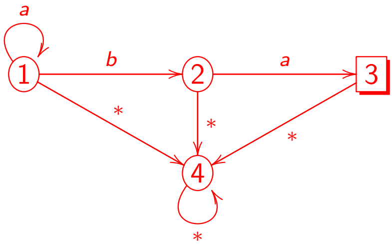


- 为了帮助理解题意，再给出一些说明。
- 容易构造一个状态数为$n+2$的DFA自动机使得它只能接收给定的串。

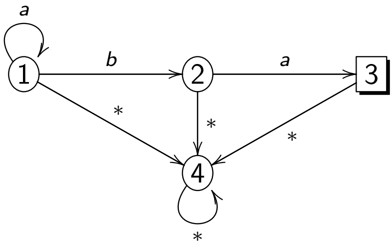


- 整个串由同一种字符构成时，无解。

- 可以构造出一个点数不超过$n+1$的DFA来接收将串首字符无限复制的串集。



- 可以构造出一个点数不超过$n+1$的DFA来接收将串首字符无限复制的串集。



- 同理可以对串尾字符构造，将这个两个DFA取交即为给定串。

题目大意

对于一个$1 \sim n$的排列a，定义a/i为在该排列中去掉i后，剩下数相对大小和位置不变构成一个$1 \sim n-1$的排列。
如$(1,3,5,2,6,4)/2 = (1,2,4,5,3)$。
所有$1 \le i \le n$的a/i顺序打乱后给出。
要求还原一个合法的排列a。

数据范围

$5 \le n \le 300$

- 移除一个$i(\geq 2)$后，排列中1的大小一定不变，位置不变或前移一格。

Permutation Reconstruction
Andrew Stankevich Contest 24 E

- 移除一个$i(\geq 2)$后，排列中1的大小一定不变，位置不变或前移一格。
  - 1在原排列中的位置最多只有3种可能。

- 移除一个$i(\geq 2)$后，排列中1的大小一定不变，位置不变或前移一格。
  - 1在原排列中的位置最多只有3种可能。
- 枚举哪个读入的排列是删1的，将1补在对应位置。

- 移除一个$i(\geq 2)$后，排列中1的大小一定不变，位置不变或前移一格。
  - 1在原排列中的位置最多只有3种可能。
- 枚举哪个读入的排列是删1的，将1补在对应位置。
  - 用*hash*判断是否和输入符合。

- 移除一个$i(\geq 2)$后，排列中1的大小一定不变，位置不变或前移一格。
  - 1在原排列中的位置最多只有3种可能。
- 枚举哪个读入的排列是删1的，将1补在对应位置。
  - 用$hash$判断是否和输入符合。
- 复杂度$O(n^3)$

Robots' Art

Andrew Stankevich Contest 35 H

题目大意

给出两个由$\leq k$的正整数构成的数列A, B，长度为n, m。
求出一个由$\leq k$的正整数构成的数列C，使得C既不是A的子序列也不是B的子序列。
要求输出一个最小长度的可行方案。

数据范围

$n, m, k \leq 5000$

Robots' Art
Andrew Stankevich Contest 35 H

- 对于一个已知数列，如何判断它是否是A, B的子序列？

- 对于一个已知数列，如何判断它是否是A, B的子序列？
  - 在A, B上贪心逐个匹配，尽可能匹配靠前位置。

- 对于一个已知数列，如何判断它是否是A, B的子序列？
  - 在A, B上贪心逐个匹配，尽可能匹配靠前位置。
- 预估一下答案的上界。

- 对于一个已知数列，如何判断它是否是A, B的子序列？
  - 在A, B上贪心逐个匹配，尽可能匹配靠前位置。

- 预估一下答案的上界。
- 设$nextA_{i,c}$为在数列A中，i之后第一次出现c的位置。

- 对于一个已知数列，如何判断它是否是A, B的子序列？
  - 在A, B上贪心逐个匹配，尽可能匹配靠前位置。
- 预估一下答案的上界。
- 设$nextA_{i,c}$为在数列A中，i之后第一次出现c的位置。
  - 这个数组可以$O(nk)$递推得到。

Robots' Art

Andrew Stankevich Contest 35 H

- 对于一个已知数列，如何判断它是否是A, B的子序列？
  - 在A, B上贪心逐个匹配，尽可能匹配靠前位置。
- 预估一下答案的上界。
- 设$nextA_{i,c}$为在数列A中，i之后第一次出现c的位置。
  - 这个数组可以$O(nk)$递推得到。
  - 对于一个i，$Max\{nextA_{i,c}\} - i \geq k - 1$。

- 对于一个已知数列，如何判断它是否是A, B的子序列？
  - 在A, B上贪心逐个匹配，尽可能匹配靠前位置。
- 预估一下答案的上界。
- 设$nextA_{i,c}$为在数列A中，i之后第一次出现c的位置。
  - 这个数组可以$O(nk)$递推得到。
  - 对于一个i，$Max\{nextA_{i,c}\} - i \geq k - 1$。
    - 一定能找到一个c，使得c下次出现在k个位置之后。

- 对于一个已知数列，如何判断它是否是A, B的子序列？
  - 在A, B上贪心逐个匹配，尽可能匹配靠前位置。

- 预估一下答案的上界。
- 设$nextA_{i,c}$为在数列A中，i之后第一次出现c的位置。
  - 这个数组可以$O(nk)$递推得到。
  - 对于一个i，$Max\{nextA_{i,c}\} - i \geq k - 1$。
    - 一定能找到一个c，使得c下次出现在k个位置之后。

- 可以贪心每次选择一个在A, B中下次出现最靠后的数放到C末端，所以答案在$O(\frac{n+m}{k})$范围内。

- 设$F[i][j]$表示长度为i的C数列，在A数列中匹配到第j位时，B数列中匹配的最远位置。

- 设$F[i][j]$表示长度为i的C数列，在A数列中匹配到第j位时，B数列中匹配的最远位置。
  - 可以$O(k)$枚举下一个位置的数。

- 设$F[i][j]$表示长度为i的C数列，在A数列中匹配到第j位时，B数列中匹配的最远位置。
  - 可以$O(k)$枚举下一个位置的数。
  - 利用$nextA_{i,c}$和$nextB_{i,c}$转移。

- 设$F[i][j]$表示长度为i的C数列，在A数列中匹配到第j位时，B数列中匹配的最远位置。
  - 可以$O(k)$枚举下一个位置的数。
  - 利用$nextA_{i,c}$和$nextB_{i,c}$转移。
- 这一部分的复杂度为

$$O(|C_{ans}|nk) \le O(\frac{n+m}{k}nk) = O((n+m)n)$$

ASC选讲
任之洲
ASC35F
ASC45D
ASC47D
ASC24E
ASC35H
ASC45H
ASC18I

- 设$F[i][j]$表示长度为i的C数列，在A数列中匹配到第j位时，B数列中匹配的最远位置。
  - 可以$O(k)$枚举下一个位置的数。
  - 利用$nextA_{i,c}$和$nextB_{i,c}$转移。
- 这一部分的复杂度为

$$O(|C_{ans}|nk) \le O(\frac{n+m}{k}nk) = O((n+m)n)$$

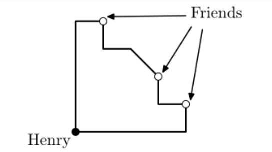
- 时间复杂度$O((n+m)k+(n+m)n)$

题目大意

给出一个n个点的多边形。

- 1号点所在的角是凸的，即两边多边形内部夹角$< 180°$。
- 1号点能在多边形内部看见其他所有点。

求一个最大的顶点集合，使得点集中的点两两不可见。



数据范围

$3 \leq n \leq 500$，$-10^5 \leq x_i, y_i \leq 10^5$

Hide and Seek
Andrew Stankevich Contest 45 H

- 设$F[l][r]$为$l \sim r$这一段顶点中能选取的最大点集。

Hide and Seek

Andrew Stankevich Contest 45 H

- 设$F[l][r]$为$l \sim r$这一段顶点中能选取的最大点集。
  - 不选点l的情况即为$F[l+1][r]$。

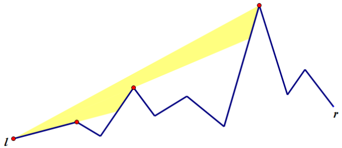
- 设$F[l][r]$为$l \sim r$这一段顶点中能选取的最大点集。
  - 不选点l的情况即为$F[l+1][r]$。
  - 假设选择了点l，设点l能依次看到点$a_1 \sim a_m$。

$$1 + \sum_{i=1}^{m-1} F[a_i + 1][a_{i+1} - 1]$$

- 设$F[l][r]$为$l \sim r$这一段顶点中能选取的最大点集。
  - 不选点l的情况即为$F[l+1][r]$。
  - 假设选择了点l，设点l能依次看到点$a_1 \sim a_m$。

$$1 + \sum_{i=1}^{m-1} F[a_i + 1][a_{i+1} - 1]$$

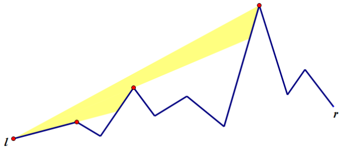
  - 可以参考下图



- 设$F[l][r]$为$l \sim r$这一段顶点中能选取的最大点集。
  - 不选点l的情况即为$F[l+1][r]$。
  - 假设选择了点l，设点l能依次看到点$a_1 \sim a_m$。

$$1 + \sum_{i=1}^{m-1} F[a_i + 1][a_{i+1} - 1]$$

  - 可以参考下图



- 时间复杂度$O(n^3)$或$O(n^2)$

题目大意

给出一个$1\sim k$的排列B，求有多少$1\sim n$的排列A满足

- B是A子序列。
- $A_{A_i}=i$。

数据范围

$1\le k\le n\le 200$，$B_i\le k$

- 因为B是A的子序列，设$A_{b_i} = B_i$，那么$A_{B_i} = b_i$。

- 因为B是A的子序列，设$A_{b_i} = B_i$，那么$A_{B_i} = b_i$。
  - 对于$i < j$，一定满足$A_{B_i} < A_{B_j}$。

- 因为B是A的子序列，设$A_{b_i} = B_i$，那么$A_{B_i} = b_i$。
  - 对于$i < j$，一定满足$A_{B_i} < A_{B_j}$。
- 枚举$0 \le m \le k$，强制对于$i \le m$满足$A_{B_i} \le k$。

- 因为B是A的子序列，设$A_{b_i} = B_i$，那么$A_{B_i} = b_i$。
  - 对于$i < j$，一定满足$A_{B_i} < A_{B_j}$。
- 枚举$0 \le m \le k$，强制对于$i \le m$满足$A_{B_i} \le k$。
  - 对于$i > m$满足$A_{B_i} > k$。

- 因为B是A的子序列，设$A_{b_i} = B_i$，那么$A_{B_i} = b_i$。
  - 对于$i < j$，一定满足$A_{B_i} < A_{B_j}$。
- 枚举$0 \le m \le k$，强制对于$i \le m$满足$A_{B_i} \le k$。
  - 对于$i > m$满足$A_{B_i} > k$。
- 对于$i > m$的那些B_i，需要将b_i填入A_{B_i}。

- 因为B是A的子序列，设$A_{b_i} = B_i$，那么$A_{B_i} = b_i$。
  - 对于$i < j$，一定满足$A_{B_i} < A_{B_j}$。
- 枚举$0 \le m \le k$，强制对于$i \le m$满足$A_{B_i} \le k$。
  - 对于$i > m$满足$A_{B_i} > k$。
- 对于$i > m$的那些B_i，需要将b_i填入A_{B_i}。
- 剩下的位置必须将$i \le m$的B_i依次填入。

- 因为B是A的子序列，设$A_{b_i} = B_i$，那么$A_{B_i} = b_i$。
  - 对于$i < j$，一定满足$A_{B_i} < A_{B_j}$。
- 枚举$0 \le m \le k$，强制对于$i \le m$满足$A_{B_i} \le k$。
  - 对于$i > m$满足$A_{B_i} > k$。
- 对于$i > m$的那些B_i，需要将b_i填入A_{B_i}。
- 剩下的位置必须将$i \le m$的B_i依次填入。
  - 方案是唯一的。

- 考虑$i > m$的这些A_{B_i}，这些$A_{B_i} > k$。

- 考虑$i > m$的这些A_{B_i}，这些$A_{B_i} > k$。
  - 对于$i < j$，一定满足$A_{B_i} < A_{B_j}$。

- 考虑$i > m$的这些A_{B_i}，这些$A_{B_i} > k$。
  - 对于$i < j$，一定满足$A_{B_i} < A_{B_j}$。
  - 方案数为$\binom{n-k}{k-m}$。

- 考虑$i > m$的这些A_{B_i}，这些$A_{B_i} > k$。
  - 对于$i < j$，一定满足$A_{B_i} < A_{B_j}$。
  - 方案数为$\binom{n-k}{k-m}$。
- 考虑$i > k$且$A_i > k$的那些位置，共有$n-k-m$个位置。

- 考虑$i > m$的这些A_{B_i}，这些$A_{B_i} > k$。
  - 对于$i < j$，一定满足$A_{B_i} < A_{B_j}$。
  - 方案数为$\binom{n-k}{k-m}$。
- 考虑$i > k$且$A_i > k$的那些位置，共有$n - k - m$个位置。
  - 可以设计简单的递推式

$$f_N = f_{N-1} + (N-1)f_{N-2}$$

- 考虑$i > m$的这些A_{B_i}，这些$A_{B_i} > k$。
  - 对于$i < j$，一定满足$A_{B_i} < A_{B_j}$。
  - 方案数为$\binom{n-k}{k-m}$。
- 考虑$i > k$且$A_i > k$的那些位置，共有$n-k-m$个位置。
  - 可以设计简单的递推式

$$f_N = f_{N-1} + (N-1)f_{N-2}$$

- 至此，已经完成了所有计算，只需要枚举m判断后累加$\binom{n-k}{k-m}f_{n-m-k}$即可。

- 考虑$i > m$的这些A_{B_i}，这些$A_{B_i} > k$。
  - 对于$i < j$，一定满足$A_{B_i} < A_{B_j}$。
  - 方案数为$\binom{n-k}{k-m}$。
- 考虑$i > k$且$A_i > k$的那些位置，共有$n-k-m$个位置。
  - 可以设计简单的递推式

$$f_N = f_{N-1} + (N-1)f_{N-2}$$

- 至此，已经完成了所有计算，只需要枚举m判断后累加$\binom{n-k}{k-m}f_{n-m-k}$即可。
- 复杂度$O(n+m^2)$。

祝大家省选顺利！

id=44130600