数学

By i207M
Graduated from SJZEZ
Studying @THU

Powered by Marp

学长先唠叨几句

志存高远，脚踏实地。

研究整数的理论。
研究模意义下运算的理论。
研究约数、倍数、质数的理论。

质数

最基本的性质

我们说，如果存在一个整数 k ，使得 $a = kd$ ，则称 d 整除 a ，记做 $d \mid a$ ，称 a 是 d 的倍数，如果 $d > 0$ ，称 d 是 a 的约数。特别地，任何整数都整除 0 。

显然大于 1 的正整数 a 可以被 1 和 a 整除，如果除此之外 a 没有其他的约数，则称 a 是素数，又称质数。任何一个大于 1 的整数如果不是素数，也就是有其他约数，就称为是合数。 1 既不是合数也不是素数。

唯一分解定理&标准分解式

$8=2^3$ $24=2^3\times 3$ $60=2^2\times 3\times 5$

素数计数函数：小于或等于 x 的素数的个数，用 $\pi(x)$ 表示。随着 x 的增大，有这样的近似结果： $\pi(x) \sim \frac{x}{\ln(x)}$

$\pi(100) \approx \frac{100}{\ln 100}$

结论需要记忆。有时可以用来计算复杂度。

10^6

10^9

质数判定

最暴力的做法是从$2-(n-1)$枚举。

反证： $n = p \times q > n$

$> \sqrt{n} \quad > \sqrt{n}$

合数一定有小于$\sqrt{n}$的因子，于是可以优化！

```
bool isPrime(a)
{
    if (a < 2) return 0;
    for (int i = 2; i * i <= a; ++i)
        if (a % i == 0) return 0;
    return 1;
}
```

$O(\sqrt{n})$

简单的加速：

如果预处理出$\sqrt{n}$以内的质数，则时间复杂度为$O(\sqrt{n}/\log n)$

$$\approx \frac{\sqrt{n}}{\ln\sqrt{n}} = \frac{\sqrt{n}}{\frac{1}{2}\ln n^{\cancel{\frac{1}{2}}}}$$

质因子分解

怎样$O(\sqrt{n})$地分解出质因数?

12

1 2 3 6 12

$n : d \quad \frac{n}{d}$

$12 = 2^2 \times 3^1$

$d(12) = (2+1)(1+1) = 6$

怎样快速枚举一个数的所有因数?

1	2^0
2	2^1
3	3^1
4	
6	$2^1 \times 3^1$
12	$2^2 \times 3^1$

$n = p_1^{k_1} p_2^{k_2} \cdots p_s^{k_s}$

for $0 \ldots k_1$ $\quad k_1+1$

for $0 \ldots k_2$ $\quad k_2+1$

$\ddots$ for $0 \ldots k_s$

$d(n) = \prod_{i=1}^{s} (k_i+1)$

$d = p_1^{i_1} p_2^{i_2} \cdots p_s^{i_s}$

约数个数的上界

$2\times3\times5\times7\times11\times13$

d^3

质因数

因数

$n \le$	10^1	10^2	10^3	10^4	10^5	10^6	10^7	10^8	10^9
$\max\{\omega(n)\}$	2	3	4	5	6	7	8	8	9
$\max\{d(n)\}$	4	12	32	64	128	240	448	768	1344
$n \le$	10^{10}	10^{11}	10^{12}	10^{13}	10^{14}	10^{15}	10^{16}	10^{17}	10^{18}
$\max\{\omega(n)\}$	10	10	11	12	12	13	13	14	15
$\max\{d(n)\}$	2304	4032	6720	10752	17280	26880	41472	64512	103680

d^2

10^5

~~1~~ 2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~

~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~

17 ~~18~~ 19 ~~20~~

筛法

Eratosthenes 筛法

考虑这样一件事情：如果 x 是合数，那么 x 的倍数也一定是合数。利用这个结论，我们可以避免很多次不必要的检测。

如果我们从小到大考虑每个质数，然后同时把当前这个数的所有（比自己大的）倍数记为合数，那么运行结束的时候没有被标记的数就是素数了。

$$\frac{n}{2}+\frac{n}{3}+\frac{n}{4}\cdots\cdots\frac{n}{n}=n\underbrace{\left(\frac{1}{2}+\frac{1}{3}+\cdots\frac{1}{n}\right)}$$

$$\approx n\log n$$

$2i \quad 3i \quad \cdots \quad (i-1)i \quad i \times i \quad (i+1) \cdot i$

```
int Eratosthenes(int n) {
  int p = 0;
  for (int i = 0; i <= n; ++i) is_prime[i] = 1;
  is_prime[0] = is_prime[1] = 0;
  for (int i = 2; i <= n; ++i) {
    if (is_prime[i]) {
      prime[p++] = i;  // 把i加入质数表里,后置自增运算代表当前素数数量
      if ((long long)i * i <= n)
        for (int j = i * i; j <= n; j += i)
          // 因为从 2 到 i - 1 的倍数我们之前筛过了，这里直接从
          // i 的倍数开始，提高了运行速度
          is_prime[j] = 0;  // 是i的倍数的均不是素数
    }
  }
  return p;
}
```

$12 \begin{cases} 2 \\ 3 \end{cases} \quad 14 \begin{cases} 2 \\ 7 \end{cases} \quad \sum_{prime} \frac{n}{p}$

注意复杂度是$O(n \log \log n)$

循环从i^2开始可以显著加速（但是不影响复杂度）

$12 \leftarrow 2$

$9 \leftarrow 3$

$57 \leftarrow 57$

$5 \times 7 = 35$

$2 \times 5 \times 7 = 70$

$3 \times 5 \times 7 =$

5×7

$\mathrm{mindiv}[n] = P_i$

$$\mathrm{mindiv}\begin{bmatrix} n \cdot P_1 \\ P_2 \\ \vdots \\ P_{i-1} \end{bmatrix} = \begin{matrix} P_1 \\ P_2 \\ \vdots \\ P_{i-1} \end{matrix}$$

线性筛

核心思想：在每个数的最小质因子处遍历它。

n 会被谁标记？

$(5 \times 7)^2$

$\frac{n}{\mathrm{mindiv}[n]}$ 标记！

$\mathrm{mindiv}(n \cdot P_i) = P_i$

$\mathrm{mindiv}[n]$

$= n$

$\mathrm{mindiv}[n \cdot P_{i+1}] = P_i$

prime: 2, 3, 5, 7

~~1~~ 2 3 ~~4~~ 5 ~~6~~

7 ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~

13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~

$\frac{6}{mindiv(6)} = \frac{6}{2} = 3$

$\frac{12}{mindiv(12)} = 6$

```
bool notp[N];
int pri[N], cnt;
void sieve(int n)
{
    notp[1] = 1;
    for(int i = 2; i <= n; ++i)
    {
        if(!notp[i]) pri[++cnt] = i;
        for(int j = 1, t; j <= cnt && (t = i * pri[j]) <= n; ++j)
        {
            notp[t] = 1;
            if(i % pri[j] == 0) break;
        }
    }
}
```

关于线性筛一定要记住的性质：线性筛在每个数的最小质因子处遍历它。

所以我们稍加修改就可以顺便筛出每个数的最小质因子。

线性筛还可以筛φ, μ, σ等一切积性函数。
至于什么是积性函数，留待以后学习。

剩余系运算

2.1 取模

因为取模具有很好的性质，比如

$$(a+b) \bmod p = ((a \bmod p) + (b \bmod p)) \bmod p$$

$$(a-b) \bmod p = ((a \bmod p) - (b \bmod p)) \bmod p$$

$$(a \times b) \bmod p = ((a \bmod p) \times (b \bmod p)) \bmod p$$

$a - b = a + p - b$

对于除法，还可以乘法逆元。

所以如果最终的答案比较大，通过要求取模是很常见的做法。

$n = \frac{n}{m} \cdot m + (n \bmod m)$

$x = k \cdot m + r$

a^{2^0}

$a \cdot a = a^2$

$a^2 \cdot a^2 = a^4$

a^8

快速幂

求$a^b \mod p$, $\underline{b \le 10^{18}}$ 2^{64}

$1 0 1 0 1 0$

$a^b = a^{2^1} \cdot a^{2^3}$ a^{2^0} a^{2^1} a^{2^2} a^{2^3}

$b = 1010_{(2)}$

结合律 $a^{14} = (a^2)(a^4)(a^8)$

二进制分组的思想。

```
int qpow(int a, int b)
{
    int res = 1;
    for(; b; b >>= 1, a = (LL)a * a % md) if(b & 1) res = (LL)res * a % md;
    return res;
}
```

矩阵乘法 A^n 无交换律

$123 = 1\times 100 + 2\times 10 + 3\times 1$

求$a^b \mod p,\ b \le 10^{10^6}$

~~高精度转二进制~~

$a \quad a^2 \quad a^4 \quad \cdots \quad a^{2^{\cdots}}$

个 $a^{0 \cdots 9}$

十 $a^{10\ 20 \cdots 90}$

十进制快速幂!

$\vdots$ $a^{0 \cdots 9 \times 10^i}$

位数

$\wedge 10$

$(a^b)^c = a^{bc}$

最大公约数

$\gcd(a,b)$ = gcd(b, a%b)

gcd(a,b)= gcd(a-b, b)

$a = a' \cdot g$ $\quad a-b=(a'-b')g$

$b = b' \cdot g$ $\quad b = b'g$

设 $a > b$

① $b < \frac{a}{2}$ $\quad a\%b < b < \frac{a}{2}$

② $b > \frac{a}{2}$ $\quad a\%b = a-b < \frac{a}{2}$

欧几里得算法：

```
int gcd(int a, int b)
{
    if(b == 0) return a;
    return gcd(b, a % b);
}
```

为什么复杂度是$O(\log v)$?

$a = p_1^{k_1} + p_2^{k_2} \cdots\cdots p_n^{k_n}$

$b = p_1^{l_1} + p_2^{l_2} \cdots\cdots + p_n^{l_n}$

m, n

$m+n = \min(m,n) + \max(m,n)$

最小公倍数

$\mathrm{lcm} \cdot \gcd = ab$

$$\operatorname{lcm}(a,b) = ab/\gcd(a,b)$$

$= (a,b)$

$\gcd(a,b) = \dfrac{p_1^{\min(k_1,l_1)} + p_2^{\min(k_2,l_2)} \cdots\cdots}{p_1^{\max(k_1,l_1)} \quad p_2^{\max(k_2,l_1)} \cdots\cdots}$

$\operatorname{lcm}(a,b) =$

例题：P1029 最大公约数和最小公倍数问题

https://www.luogu.com.cn/problem/P1029

两个数的积等于它们最大公约数和它们最小公倍数的积。所以满足条件的两个数的积一定是读入的两个数的积。

$$a'gx + b'gy = g(a'x + b'y)$$

$$ax + by = z$$

扩展欧几里得算法

扩展欧几里得算法（Extended Euclidean algorithm, EXGCD），常用于求 $ax + by = \gcd(a, b)$ 的一组可行解。

可行性由裴蜀定理证明。

设

$ax_1 + by_1 = \gcd(a, b)$ 要求出

$bx_2 + (a \bmod b)y_2 = \gcd(b, a \bmod b)$ 假设已求

由欧几里得定理可知： $\gcd(a, b) = \gcd(b, a \bmod b)$

所以 $ax_1 + by_1 = bx_2 + (a \bmod b)y_2$

求

(b, a mod b)

又因为 $a \bmod b = a - (\lfloor \frac{a}{b} \rfloor \times b)$

所以 $ax_1 + by_1 = bx_2 + (a - (\lfloor \frac{a}{b} \rfloor \times b))y_2$

$ax_1 + by_1 = ay_2 + bx_2 - \lfloor \frac{a}{b} \rfloor \times by_2 = ay_2 + b(x_2 - \lfloor \frac{a}{b} \rfloor y_2)$

所以 $x_1 = y_2, y_1 = x_2 - \lfloor \frac{a}{b} \rfloor y_2$

将 x_2, y_2 不断代入递归求解直至 $\gcd$ （最大公约数，下同）为 0

递归 x=1,y=0 回去求解。

(a, 0) = a

a·1 + 0·0 = a

```cpp
int exgcd(int a, int b, int &x, int &y)
{
    if (!b) {x = 1, y = 0; return a;}
    int d = exgcd(b, a % b, x, y); // 实际做法可以在这里交换x,y
    int t = x;
    x = y;
    y = t - (a / b) * y;
    return d;
}
```

例题：P1082 同余方程

https://www.luogu.com.cn/problem/P1082

$$ax \equiv 1 \mod b$$

$$ax + kb = 1$$

$$(x \% b + b) \% b$$

同余方程→二元一次方程。

费马小定理

若a是一个整数，p是一个质数，则有

$$a^p = a \pmod p$$

（注意，此公式不要求$(a,p)=1$)

$a^{p-1} \equiv 1$

证明?

法一

提示：对a归纳

题 8. (1) 设 p 是素数，$1 \le i \le p-1$，证明：$p \mid \binom{p}{i}$，其中 $\binom{p}{i} = \frac{p!}{i!(p-i)!}$. $= \frac{\boxed{p} \cdot (p-1)!}{i!\,(p-i)!}$ $= p \cdot k$

(2) 设 $a, b \in \mathbb{Z}$，证明 $(a+b)^p \equiv a^p + b^p \mod p$.

$(a+b)^p = \sum_{i=0} \binom{p}{i} a^i \cdot b^{p-i}$

(3) 证明：对任意的正整数 $a \ge 1$，总有 $a^p \equiv a \mod p$.

出自：清华大学-高等线性代数(1)[朱敏娴]-第13周作业

假设 a 成立 $1^p \equiv 1$

$(a+1)^p = a^p + 1^p = a+1$

法二

引理一：质数剩余系是整环，即，若$ac = bc \pmod p$且$c \neq 0 \pmod p$，则$a = b \pmod p$.

$\{1, 2 \cdots p-1\}$

引理二：若$(a, p) = 1$，则$A = \{a, 2a, \ldots, (p-1)a\}$是完全剩余系。

由引理二，$1 \times 2 \times \ldots \times (p-1) = a \times 2a \times \ldots \times (p-1)a \pmod p$
即$(p-1)! = (p-1)! \times a^{p-1} \pmod p$
即$a^{p-1} = 1 \pmod p$。最后再补上$a = 0 \pmod p$的情况。

实数下，$a^{-1}=\frac{1}{a}$

a^{-1}是a的逆元，$a^{-1}\cdot a=a\cdot a^{-1}=1$

剩余系求逆元

称a与b互逆

对于某个a，是否存在b，使得$ab \equiv 1 \pmod m$

首先考虑逆元的存在性和唯一性。

$\frac{b}{a}=b\cdot a^{-1} \mod m$

$\frac{b}{a}\cdot a=b \pmod m$

x

定义$\frac{b}{a}$为满足$ax=b$的唯一x

mod 7

表示 $3\times 2^{-1}=3\times 4=5$

2的逆元为4

$5\times 2=3$

$a\boxed{x} \equiv 1 \bmod b$

存在性：当且仅当$(a, m) = 1$时有逆元。

证明：考虑$ax + my = 1$的解

$(ax + by = 1) \bmod b$

$ax = 1$

唯一性：逆元若存在，一定唯一。

证明：假设$ab = ac = 1 \pmod{m}$，则$b = bac = (ba)c = c \pmod{m}$

$b = b \times 1$

$ba = 1$

~~孔乙己：求逆元有5种写法，你知道吗？~~

$\bmod 7$

$\{1, 8, 15, 22\} = 1 \quad \{7k+1\}$

$\{2, 9, \ldots\} = 2 \quad \{7k+2\}$

$\frac{1}{2} = 4$

$\{\frac{1}{2}, \frac{1}{2} + 7, \ldots\} = 4$ 整数

mod 7 $\quad 2^{-1} = 2^{7-2} = 2^5 = 32 \% 7$

$= 28+4$

$= 4$

法一：费马小定理

$$a \times a^{p-2} = 1 \pmod p$$

要求p是质数。

$a^{p-1} = 1$ $\qquad 2 \times 4 = 8 = 1$

$a \cdot a^{p-2} = 1$

法二：EXGCD

求$ax + my = 1$的解。

法三：线性求逆元

求出 $1, 2, \ldots, n$ 中每个数关于 p 的逆元。

首先，$1^{-1} \equiv 1 \pmod p$；

其次对于递归情况 i^{-1}，我们令 $k = \lfloor \frac{p}{i} \rfloor$，$j = p \bmod i$，有 $p = ki + j$。再放到 $\bmod p$ 意义下就会得到：$ki + j \equiv 0 \pmod p$；

两边同时乘 $i^{-1} \times j^{-1}$：

$0 \equiv k\cancel{i}\,i^{-1} \cdot j^{-1} + \cancel{j} \cdot i^{-1} \cdot \cancel{j^{-1}}$

$$kj^{-1} + i^{-1} \equiv 0 \pmod p$$

$$i^{-1} \equiv -kj^{-1} \pmod p$$

再带入 $j = p \bmod i$，有 $p = ki + j$，有：

$$i^{-1} \equiv -\lfloor \frac{p}{i} \rfloor \boxed{(p \bmod i)^{-1}} \pmod p$$

我们注意到 $p \bmod i < i$，而在迭代中我们完全可以假设我们已经知道了所有的模 p 下的逆元 $j^{-1}, j < i$。

故我们就可以推出逆元，利用递归的形式，而使用迭代实现：

```
inv[1] = 1;
for (int i = 2; i <= n; ++i) inv[i] = (LL)(p - p / i) * inv[p % i] % p;
```

使用 $p - \lfloor \frac{p}{i} \rfloor$ 来防止出现负数。

$- p/i \cdot [p\%i]$

另外，根据线性求逆元方法的式子： $i^{-1} \equiv -kj^{-1} \pmod p$

递归求解 j^{-1} , 直到 $j=1$ 返回 1 。

中间优化可以加入一个记忆化来避免多次递归导致的重复，这样求 $1,2,\ldots,n$ 中所有数的逆元的时间复杂度仍是 $O(n)$ 。

注意：如果用以上给出的式子递归进行单个数的逆元求解，目前已知的时间复杂度的上界为 $O(n^{\frac{1}{3}})$ ，具体请看知乎讨论。

$O(\log n)$

法四：线性求逆元2

上面的方法只能求 1 到 n 的逆元，如果需要求任意给定 n 个数（$1 \le a_i < p$）的逆元，就需要下面的方法：

首先计算 n 个数的前缀积，记为 s_i ，然后使用快速幂或扩展欧几里得法计算 s_n 的逆元，记为 sv_n 。

因为 sv_n 是 n 个数的积的逆元，所以当我们把它乘上 a_n 时，就会和 a_n 的逆元抵消，于是就得到了 a_1 到 a_{n-1} 的积逆元，记为 sv_{n-1} 。

$\cancel{a_1^{-1} \cdot a_2^{-1} \cdot a_3^{-1} \cdots a_{i-1}^{-1}} \cdot a_i^{-1}$

同理我们可以依次计算出所有的 sv_i ，于是 a_i^{-1} 就可以用 $s_{i-1} \times sv_i$ 求得。

所以我们就在 $O(n + \log p)$ 的时间内计算出了 n 个数的逆元。

法五：线性筛

逆元是完全积性函数，可以线性筛求。

（其实质数的逆元还得单独求...）

中国剩余定理(Chinese Remainder Theorem)

$$\begin{cases} x \equiv b_1 \pmod{a_1} \\ x \equiv b_2 \pmod{a_2} \\ \dots \\ x \equiv b_n \pmod{a_n} \end{cases}$$

$$\begin{cases} x \equiv 3 \bmod 7 \\ x \equiv 1 \bmod 2 \\ x \equiv 2 \bmod 5 \end{cases}$$

其中$\{a_i\}$互质，求x

肯定有无数组解，而且解都是最小的解$+\mathrm{lcm}(a_i)$得到的。如何求出最小的正解?

x $\quad x+\mathrm{lcm}$

$M_i \cdot M_i^{-1} \equiv 1 \mod a_i$

$x \% \mathrm{lcm} \% a_i$
$= x \% a_i$

结论： $M_i \cdot M_i^{-1} \equiv 1 \mod \mathrm{lcm}(a_i)$

在lcm$\prod$(a_i)的剩余系下，有唯一解：$\sum b\, a_i \times M_i \times M_i^{-1}$。

其中，$M_i = \prod_{j \neq i} a_j$，$M_i^{-1}$为$M_i$在 $\pmod{a_i}$下的逆元。

令 $M = \prod a_i$

$M_i = \frac{M}{a_i}$

证明：

验证，非常巧妙地凑出了答案。 mod a_i下逆元

$ans = b_1 \cdot a_2 a_3 (a_2 a_3)^{-1} + b_2 \cdot a_1 a_3 (a_1 a_3)^{-1} + b_3 \cdot a_1 a_2 (a_1 a_2)^{-1}$

$\% a_2$

扩展中国剩余定理(EXCRT)

处理模数不互质的情况。

两两合并。 $x = k_1 a_1 + b_1 = k_2 a_2 + b_2$

考虑两个式子：$x \equiv b_1 \pmod{a_1}, x \equiv b_2 \pmod{a_2}$

用两种方法表示x，移项，$a_1 k_1 + a_2 k_2 = b_2 - b_1$，注意到$a_2$的系数的负号被忽略了，因为把负号给了$k_2$；

$b_2 - b_1$

我们可以使用EXGCD解出一组解$a_1 k_1 + a_2 k_2 \equiv \gcd(a_1, a_2)$，**有解的条件为$\gcd(a_1, a_2) | b_2 - b_1$**；

$\times \quad \times \quad \times \frac{b_2 - b_1}{gcd}$

我们可以将k_1带入，解得X，同时满足这两个式子，**所以式子变为了**$x \equiv X \mod (\operatorname{lcm}(a_1, a_2))$

$X = (k_1 a_1 + b_1) \% lcm$

例题：P3868 [TJOI2009]猜数字

https://www.luogu.com.cn/problem/P3868

欧拉函数

欧拉函数（Euler's totient function），即 $\varphi(n)$，表示的是小于等于 n 和 n 互质的数的个数。

$\varphi(1)=1$ $\varphi(3)=2$ $\varphi(6)=2$

$\varphi(2)=1$ $\varphi(8)=$ 1 2 3 4 5 6 7 8

4

$f(xy)=f(x)f(y)$

当 (x,y) 互质时

- 欧拉函数是积性函数。

  特别地，当 n 是奇数时 $\varphi(2n)=\varphi(n)$ 。

证明：

$= \{$与 mn 互质$\}$

设 $(m,n)=1$，即证明 $\varphi(mn)=\varphi(m)\varphi(n)$，构造两个集合 $A=\{a:1\le a\le mn,\gcd(a,mn)=1\}, B=\{(b,c):1\le b\le m,\gcd(b,m)=1,1\le c\le n,\gcd(c,n)=1\}$。只需证明两个集合等势。

b 与 m 互质，c 与 n 互质

大小相等 $\varphi(m)$ $\varphi(n)$

① 与 mn 互质的 $x_1 \neq x_2$

$x \to (x \bmod m,\ x \bmod n)$

$x_1 \to (x_1 \bmod m,\ x_1 \bmod n)$

$x_2 \quad\quad x_2 \quad y_1 \quad\quad x_2 \quad y_2$

$|A| \le |B|$

$$\begin{cases} x \equiv y_1 \% m \\ x \equiv y_2 \% n \end{cases}$$

证明存在单射 $\Leftarrow \forall a_1, a_2 \in A,\ a_1 \neq a_2$，有 $f(a_1) \neq f(a_2)$

证明存在满射（用到中国剩余定理）

$\Leftarrow \forall b \in B,\ \exists a$ 使得 $f(a) = b \quad |B| \le |A|$

②
$$\begin{cases} x \equiv y_1 \% m \\ x \equiv y_2 \% n \end{cases} \Rightarrow \text{唯一 } x$$

$$8 = \varphi(1) + \varphi(2) + \varphi(4) + \varphi(8)$$
$$= 1 + 1 + 2 + 4 = 8$$

- $n = \sum_{d|n} \varphi(d)$ 枚举n的所有因数为d

  证明可以考虑 $\frac{1}{n} \dots \frac{n}{n}$

$$\frac{1}{8} \quad \cancel{\frac{2}{8}} \quad \frac{3}{8} \quad \cancel{\frac{4}{8}} \quad \frac{5}{8} \quad \cancel{\frac{6}{8}} \quad \frac{7}{8} \quad \cancel{\frac{8}{8}}$$

$$\frac{1}{4} \quad \frac{3}{4} \quad \frac{1}{2} \quad 1$$

$\varphi(8)$ $\varphi(4)$ $\varphi(2)$ $\varphi(1)$

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

p是质数

$1 \dots p^k$

$p, 2p \dots p^{k-1} p$

- 由唯一分解定理，设 $n = \prod_{i=1}^{n} p_i^{k_i}$ ，其中 p_i 是质数，则

$\varphi(\prod\ )=$

$$
\begin{aligned}
\varphi(n) &= \prod_{i=1}^{s} \varphi(p_i^{k_i}) \\
&= \prod_{i=1}^{s} (p_i - 1) \times {p_i}^{k_i - 1} \\
&= \prod_{i=1}^{s} {p_i}^{k_i} \times (1 - \frac{1}{p_i}) \\
&= n \prod_{i=1}^{s} (1 - \frac{1}{p_i}) \quad \square
\end{aligned}
$$

积性性质

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

n 为奇

$\varphi(2n) = \varphi(n)$

$= \varphi(2) \cdot \varphi(n)$

$= \varphi(n)$

于是我们可以$O(\sqrt{n})$求一个欧拉函数的值。

应用

求小于等于N，与N互质的数的和

$f(8) = 1 + 3 + 5 + 7$

$(x, n) = 1$

$(n-x, n) = 1$

$\underline{\varphi(n) \times n/2}$ $n=1$ $=1$

$\frac{\varphi(n)}{2} \times n$

$\frac{\varphi+\varepsilon}{2} \times n$ $\underline{\varepsilon(n) = [n=1]}$

欧拉定理

$m-1$

若 $\gcd(a,m)=1$ ，则 $a^{\varphi(m)} \equiv 1 \pmod m$ 。

证明与费马小定理的证明类似，此处从略。

扩展欧拉定理

$$a^b \equiv \begin{cases} a^{b \bmod \varphi(p)}, & \gcd(a,\, p) = 1 \\ a^b, & \gcd(a,\, p) \neq 1,\ b < \varphi(p) \\ a^{b \bmod \varphi(p) + \varphi(p)}, & \gcd(a,\, p) \neq 1,\ b \geq \varphi(p) \end{cases} \pmod p$$

欧拉

$\neq a^{b \bmod \varphi + \varphi}$

注意：扩展欧拉定理只能在指数比$\varphi(p)$大时才能用！

$$2^{2^{2^{2}}} \bmod p = 2^{\boxed{2^{2^{2}} \bmod \varphi(p)} + \varphi(p)} \bmod p$$

$$2^{2^{\cdot^{\cdot^{\cdot}}} \bmod \varphi(\varphi(p)) + \varphi(\varphi(p))}$$

例题：P4139 上帝与集合的正确用法

https://www.luogu.com.cn/problem/P4139

经典老题，可能做过?

$$\varphi(n) = n \prod \left(1 - \frac{1}{p_i}\right)$$

$$= \prod p_i^{k-1} (p_i - 1)$$

分解质因数

$O(\log p)$

p 偶数 $\varphi(2^k \times p') = 2^{k-1}\,\underline{\varphi(p')}$

$\le p'$

$\le \frac{1}{2}p$

奇数 $\varphi(3 \times 5 \times 7) \longrightarrow$ 偶

$\uparrow$

$(3-1)(5-1)\cdots\underset{\sim\sim\sim}{(p_i-1)}$

递归即可。

题

可能会混进去一些奇奇怪怪的题目。

在讲题时一定要积极思考！

求n的所有因数的φ

$n \le 10^{14}$

$O(n)$

如何在预处理后“完美”地求出前n个数的质因子分解式

“完美”指不浪费任何的复杂度。

这种方法可以在一些题目中用来优化掉$O(\sqrt{n})$的复杂度。

$O(n\log\log n)$

借助 `min_div` .

UVA11327 Enumerating Rational Numbers

以下程序可以输出任何有理数：

```
for d = 1 to infinity do
  for n = 0 to d do
    if gcd(n,d) = 1 then print n/d
```

求输出的第k个有理数。$1 \leq k \leq 14,000,000,000$

样例：

```
1              0/1
2              1/1
3              1/2
12158598919    199999/200000
0
```

2×10^5

注意到大样例的分子分母都较小（10^5），因此直接枚举即可。

要用心观察题目。

P2158 [SDOI2008]仪仗队

https://www.luogu.com.cn/problem/P2158

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

$$=2\times\sum_{i=1}^{n-1}\varphi(i)-\sum_{i=1}^{n-1}[\gcd(i,j)=1]$$

$$=2\times\sum_{i=1}^{n-1}\varphi(i)-1$$

P1072 [NOIP2009 提高组] Hankson 的趣味题

https://www.luogu.com.cn/problem/P1072

直接枚举因子。

例题：P1516 青蛙的约会

https://www.luogu.com.cn/problem/P1516

$$x \bmod m$$

求一个不定方程$(n-m)t+kl=x-y\ (k\in\mathbb{Z})$中$t$的最小非负整数解。

如何从特解扩展出去?

$+lcm$ $-lcm$

$ax+by=z$

$ax+by=\gcd$

$lcm(a,b)$

$x+=\frac{lcm}{a}=m$

$y-=\frac{lcm}{b}$

[SDOI2009]SuperGCD

求$\gcd(a, b)$

$a, b \leq 10^{10000}$

欧几里得算法处理大整数需要高精度。

Stein 算法更相减损

$\gcd(a-b, b) = \gcd(a, b)$

当这两个数都是2的倍数的时候, `++cnt` ,都除以2

如果任意一个是2的倍数,除以2,然后更相减损即可

答案是 结果×(2^幂乘器)

cnt

在大整数时显著优于欧几里得算法。

$\gcd(2n, 2m) = \boxed{2}\gcd(n, m)$

$\gcd(偶, 奇) = \gcd(\frac{偶}{2}, 奇)$

$\gcd(\overset{2}{奇-奇}, 奇)$

BZOJ 2818

给定整数N，求$1 \leq x, y \leq N$且$\gcd(x, y)$为素数的数对(x,y)有多少对.

枚举质数p, 将问题转化为求$1 \le x, y \le N$且$\gcd(x, y) = 1$的数对数目.
用线性筛法预处理欧拉函数前缀和.

CF27E

求因子数一定的最小数。

$d \le 1000$，$ans \le 10^{18}$

爆搜，策略?

指数下降，最高64；质数有限，最多15。

及时剪枝

区间素数筛

求出$[l,r]$内的素数。

$l,r <= 10^{12}, r-l <= 10^6$

很明显我们要从$r-l$入手;

每个合数一定有一个小于$\sqrt{n}$的因子，我们先筛出10^6以内的素数，然后模拟埃拉托色尼筛法。

```cpp
bool isp[N];
void sol(int l, int r) {
	for (int i = l; i <= r; ++i) isp[i - l] = 1;
	if (l == 0) isp[0] = isp[1] = 0;
	else if (l == 1) isp[0] = 0;
	for (int i = 1; i <= cnt; ++i) {
		int x = pri[i];
		if (x * x > r) break;
		for (int j = (l + x - 1) / x * x; j <= r; j += x) {
			if (j == x) continue;
			isp[j - l] = 0;
		}
	}
}
```

$O(n \log n)$

BZOJ某题

求:

$$\left(\sum_{i=l}^{r} d(i^k)\right) \bmod 998244353$$

$$l, r <= 10^{12}, r - l <= 10^6$$

利用到埃氏筛求出每个质因数

Thank you

祝大家CSP/NOIP取得好成绩!

感谢 `OI-Wiki` 的信息

如果你很强，可以尝试以下几个题目:

P3747 [六省联考 2017] 相逢是问候