数论专题

约数

河南省实验中学信息技术组

2026 年 02 月 03 日

数论专题

- 质数
- 约数
- 同余

约数

- 若正整数 n 除以整数 d 的余数为 0，即 d 能整除 n，则称 d 是 n 的约数，n 是 d 的倍数，即为 $d|n$。
- 约数集：求 $1 \sim n$ 的正约数集合。
- 试除法：遍历 $d = 1 \sim \sqrt{N}$ 中的每一个数，尝试 d 能否整除 N，若能整除，则 N/d 也是 N 的约数。
- 证明：若 $d \geq \sqrt{N}$ 是 N 的约数，则 $N/d \leq \sqrt{N}$ 也是 N 的约数。换言之，约数总是成对出现的 (完全平方数除外)。
- 时间复杂度：O($\sqrt{N}$)
- 推论：一个正整数 N 的约数个数上界为 $2\sqrt{N}$。

约数集

- 求 $1 \sim N$ 的每个正整数的正约数集合。
- 试除法：用试除法分别求出 $1 \sim N$ 每个数的正约数集合。
- 时间复杂度：$O(N\sqrt{N})$。
- 倍数法：类似埃氏筛法，对于数 d，在 d 的倍数 $d, 2d, 3d, \cdots, \lfloor \frac{N}{d} \rfloor d$ 中标记该数包含约数 d。
- 时间复杂度：$O(N + \frac{N}{2} + \frac{N}{2} + \cdots + \frac{N}{N}) = O(N \log N)$
- 推论：$1 \sim N$ 每个数的约数个数总和大约为 $N \log N$。

约数集

```
// 2×10^9 以内的自然数最多有 1536 个约数
int f[1600];
int m = 0;
for(int i = 1; i * i <= n; ++i)
    if(n % i == 0)
    {
        f[++m] = i;
        if(i != n / i) f[++m] = n / i;
    }
```

```
//f[i] 表示 i 的约数集
vector<int> f[500010];
for(int i = 1; i <= n; ++i)
    for(int j = 1; j <= n / i; ++j)
        f[i * j].push_back(i);  //i*j 包含约数 i
```

算术基本定理推论

- 在算术基本定理中，正整数 N 被唯一分解为 $N = p_1^{c_1} p_2^{c_2} \cdots p_m^{c_m}$，其中 c_i 都是正整数，p_i 都是质数，且满足 $p_1 < p_2 < \cdots < p_m$，则 N 的正约数集合可写为：

$$\left\{p_1^{b_1} p_2^{b_2} \cdots p_m^{b_m}\right\}, 0 \leq b_i \leq c_i$$

- N 的正约数个数为：

$$\mathrm{C}_{c_1+1}^1 + \mathrm{C}_{c_2+1}^1 \cdots \mathrm{C}_{c_m+1}^1 = (c_1+1)(c_2+1)\cdots(c_m+1) = \prod_{i=1}^{m}(c_i+1)$$

- N 的所有正约数的和为：

$$(1 + p_1 + p_1^2 + \cdots + p_1^{c_1}) \times \cdots \times (1 + p_m + p_m^2 + \cdots + p_m^{c_m}) = \prod_{i=1}^{m}\sum_{j=0}^{c_i}(p_i)^j$$

【例】约数个数与约数和

【题目描述】

给定 $N(1 \le N \le 100)$ 个正整数 $a_i(1 \le a_i \le 2 \times 10^9)$，请你输出这些数的乘积的约数个数和约数和，答案分别对 $10^9 + 7$ 取模。

【输入格式】

一个正整数 n，接下来一行包含 n 个整数 a_i。

【输出格式】

两行，第一行为约数个数，第二行为约数和。

【样例输入】

```
3
2 6 8
```

【样例输出】

```
12
252
```

【样例解释】

$2 \times 6 \times 8 = 96$，它的约数有 $1, 2, 3, 4, 6, 8, 12, 16, 24, 32, 96$。

【例】约数个数与约数和

- **对每个数按照算术基本定理进行分解，并将其合并。**
- **照算术基本定理求解约数个数和约数和，运算过程中取模。**
- 引理：小于等于 2×10^9 的数的不同质因子个数不会超过 *10* 个，且所有质因子的指数总和不超过 *30*。
- **证明：最小的 11 个质数的乘积** $2 \times 3 \times 5 \times 7 \times 11 \times 13 \times 17 \times 19 \times 23 \times 29 \times 31 > 2 \times 10^9$**，所以对于小于等于** 2×10^9 **的数不可能有多于 10 个质因子。那么，即使包含最小的质数，仍然有** $2^{31} > 2 \times 10^9$**，所以对于小于等于** 2×10^9 **的数的指数总和不可能超过 30。**

【例】Sumdiv

【题目描述】

求 A^B 的所有约数之和，结果对 9901 取余。

【输入格式】

一行，两个数 $A, B (1 \le A, B \le 5 \times 10^7)$。

【输出格式】

一行，一个数，A^B 的所有约数之和，结果对 9901 取余。

【样例输入】

```
2 3
```

【样例输出】

```
15
```

【样例解释】

$2^3 = 8$，它的约数有 $1, 2, 4, 8$，总和为 15。

【例】Sumdiv

- 把 A 分解质因数，表示为 $p_1^{c_1}p_2^{c_2}\cdots p_m^{c_m}$，则 A^B 表示为 $p_1^{B\times c_1}p_2^{B\times c_2}\cdots p_m^{B\times c_m}$，它的所有约数和为

$$(1+p_1+p_1^2+\cdots+p_1^{B\times c_1})\times\cdots\times(1+p_m+p_m^2+\cdots+p_m^{B\times c_m})$$

- 显然只需要求解形如 $S(p,n)=p^0+p^1+p^2+\cdots+p^n$ 对 9901 取余的结果，然后累乘取余即可。
  - 当 n 为奇数时，$S(p,n)=p^0+p^1+\cdots+p^{\frac{n}{2}}+p^{\frac{n+1}{2}}\times(p^0+p^1+\cdots+p^{\frac{n}{2}})=S(p,\frac{n}{2})+p^{\frac{n+1}{2}}\times S(p,\frac{n}{2})$，例如 $p^0+p^1+p^2+p^3+p^4+p^5=p^0+p^1+p^2+p^3\times(p^0+p^1+p^2)$。
  - 当 n 为偶数时，$S(p,n)=S(p,n-1)+p^n$。
- 利用上述分治算法和快速幂即可求出问题的解。

【例】余数之和

【题目描述】

给定正整数 n 和 k，计算 $(k \mod 1) + (k \mod 2) + \cdots + (k \mod n)$ 的值。其中 $1 \le n, k \le 10^9$。

【输入格式】

一行，包含两个正整数 n, k。

【输出格式】

一行，即余数之和。

【样例输入】

```
5 3
```

【样例输出】

```
7
```

【样例解释】

$(3 \mod 1) + (3 \mod 2) + (3 \mod 3) + (3 \mod 4) + (3 \mod 5) = 0 + 1 + 0 + 3 + 3 = 7$

【例】余数之和

- 方法 1：直接求解，时间复杂度为 O(K)，超时。
- 方法 2：观察 $k \mod i$ 的规律，可以发现在某一个区间内，$k \mod i$ 的值是一个等差数列。
  - 从后向前看，可以看出每个区间的公差每次加 1，设区间的公差为 a，则区间可以写为 $[\lfloor \frac{k}{(a+1)} \rfloor + 1, \lfloor \frac{k}{a} \rfloor]$。例如，当 $a = 1$ 时，区间为 $[51, 100]$；当 $a = 2$ 时，区间为 $[34, 50]$。
  - 但是当 i 较小 (公差 a 很大) 时，此规律便不再适用。

【例】余数之和

- 当 $n = 96, k = 100$ 时：

i	$k \% i$
1	0
2	0
3	1
4	0
5	0
6	4
7	2
8	4
9	1
10	0
11	1
12	4

i	$k \% i$
13	9
14	2
15	10
16	4
17	15
18	10
19	5
20	0
21	16
22	12
23	8
24	4

i	$k \% i$
25	0
26	22
27	19
28	16
29	13
30	10
31	7
32	4
33	1
34	32
35	30
36	28

i	$k \% i$
37	26
38	24
39	22
40	20
41	18
42	16
43	14
44	12
45	10
46	8
47	6
48	4

i	$k \% i$
49	2
50	0
51	49
52	48
53	47
54	46
55	45
56	44
57	43
58	42
59	41
60	40

i	$k \% i$
61	39
62	38
63	37
64	36
65	35
66	34
67	33
68	32
69	31
70	30
71	29
72	28

i	$k \% i$
73	27
74	26
75	25
76	24
77	23
78	22
79	21
80	20
81	19
82	18
83	17
84	16

i	$k \% i$
85	15
86	14
87	13
88	12
89	11
90	10
91	9
92	8
93	7
94	6
95	5
96	4

【例】余数之和

- 当 $n = 96, k = 100$ 时:

公差a	区间左界	区间右界	区间起点值	区间结束值
1	51	100	0	49
2	34	50	0	32
3	26	33	1	22
4	21	25	0	16
5	17	20	0	15
6	15	16	4	10
7	13	14	2	9
8	12	12	4	4
9	11	11	1	1
10	10	10	0	0
11	9	9	1	1
12	8	8	4	4
13	8	7	2	4
14	7	7	2	2
15	7	6	4	2
16	6	6	4	4

【例】余数之和

- 方法 3：观察 $k \mod i = k - \lfloor\frac{k}{i}\rfloor \times i$，故原式的计算可以转化为

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

  - 我们注意观察 $\lfloor\frac{k}{i}\rfloor$ 的规律，可以发现，如果将 $[1, n]$ 划分成若干个区间，在每个区间里 $\lfloor\frac{k}{i}\rfloor$ 的值是相等的。例如 60 除以 [21,30] 内的所有数的结果都是 2。
  - 此时 $\sum_{i=1}^{n}\lfloor\frac{k}{i}\rfloor \times i$ 在任何一个区间内都是等差数列，可以直接求和。
  - 从某种意义上来看，方法 3 和方法 2 有相似之处，只是方法 3 中的 $\lfloor\frac{k}{i}\rfloor$ 是单调 (增/减) 的，这就可以方便计算。

【例】余数之和

- 当 $n = 60, k = 60$ 时：

i	$k \% i$	$\lfloor k/i \rfloor$
1	0	60
2	0	30
3	0	20
4	0	15
5	0	12
6	0	10
7	4	8
8	4	7
9	6	6
10	0	6
11	5	5
12	0	5

i	$k \% i$	$\lfloor k/i \rfloor$
13	8	4
14	4	4
15	0	4
16	12	3
17	9	3
18	6	3
19	3	3
20	0	3
21	18	2
22	16	2
23	14	2
24	12	2

i	$k \% i$	$\lfloor k/i \rfloor$
25	10	2
26	8	2
27	6	2
28	4	2
29	2	2
30	0	2
31	29	1
32	28	1
33	27	1
34	26	1
35	25	1
36	24	1

i	$k \% i$	$\lfloor k/i \rfloor$
37	23	1
38	22	1
39	21	1
40	20	1
41	19	1
42	18	1
43	17	1
44	16	1
45	15	1
46	14	1
47	13	1
48	12	1

i	$k \% i$	$\lfloor k/i \rfloor$
49	11	1
50	10	1
51	9	1
52	8	1
53	7	1
54	6	1
55	5	1
56	4	1
57	3	1
58	2	1
59	1	1
60	0	1

【例】余数之和

- 经过发现规律，我们可以得出在 $[1, N]$ 的每一段区间 $[x, \lfloor \frac{k}{\lfloor k/x \rfloor} \rfloor]$ 的 i，$\lfloor \frac{k}{i} \rfloor$ 的值和 $\lfloor \frac{k}{x} \rfloor$ 的值相等。例如，区间 $[21, 30]$ 可以写为 $[21, \lfloor \frac{60}{\lfloor 60/21 \rfloor} \rfloor]$
- 对于 $i = 1 \sim k$，$\lfloor \frac{k}{i} \rfloor$ 的不同值最多有 $2\sqrt{k}$ 个，所以最多有 $2\sqrt{k}$ 个区间。
- 时间复杂度：$O(\sqrt{K})$。

```
long long ans = n * k;
// gx 为区间最后一项
for(long long x = 1, gx; x <= n; x = gx + 1)
{
    gx = k / x ? min(k / (k / x), n) : n;
    // (x+gx)-> 首项 + 末项
    // (gx-x+1)-> 项数
    ans -= (k / x) * (x + gx) * (gx - x + 1) / 2;
}
```

最大公约数

- 若自然数 d 同时时自然数 a 和 b 的约数，则称 d 是 a 和 b 的公约数。在所有 a 和 b 的公约数中最大的一个，称为 a 和 b 的最大公约数，记为 $\gcd(a, b)$。
- 若自然数 m 同时是自然数 a 和 b 的倍数，则称 m 是 a 和 b 的公倍数。在所有 a 和 b 的公倍数中最小的一个，称为 a 和 b 的最小公倍数，记为 $\operatorname{lcm}(a, b)$。
- 定理：$\forall a, b \in \mathbf{N}, \gcd(a, b) \times \operatorname{lcm}(a, b) = a \times b$。

求最大公约数

- 辗转相除法 (欧几里得算法)

$$\forall a, b \in \mathbf{N}, b \neq 0, \gcd(a, b) = \gcd(b, a \mod b)$$

- 更相减损术

$$\forall a, b \in \mathbf{N}, a \geq b, \gcd(a, b) = \gcd(b, a - b) = \gcd(a, a - b)$$
$$\forall a, b \in \mathbf{N}, \gcd(k \times a, k \times b) = k \times \gcd(a, b)$$

- 欧几里得算法是最常用的，时间复杂度为 $O(\log(a + b))$。
- 如果要求两个高精度数的最大公约数时，可以考虑用更相减损术。

```
int gcd(int a, int b)
{
    return b ? gcd(b, a % b) : a;
}
```

【例】Hankson 的趣味题

【题目描述】

有 $n(n \le 2000)$ 个询问，在每个询问中，给定四个自然数 a, b, c, d $(1 \le a, b, c, d \le 2 \times 10^9)$，求有多少个 x 满足 $\gcd(a, x) = b$ 且 $\mathrm{lcm}(c, x) = d$。

【输入格式】

第一行，包含一个正整数 n，接下来有 n 行，每行有四个数字 a, b, c, d。

【输出格式】

共 n 行，对于每一个询问，给出满足条件的个数。

【样例输入】

```
2
41 1 96 288
95 1 37 1776
```

【样例输出】

```
6
2
```

【样例解释】

- 第一组数据，x 可以是 9、18、36、72、144、288，共有 6 个。
- 第二组数据，x 可以是 48、1776，共有 2 个。

【例】Hankson 的趣味题

- 算法 1：从 $\operatorname{lcm}(c, x) = d$ 可以得出 x 一定是 d 的约数。故我们可以用试除法求出 d 的所有约数，逐一判断是否满足条件。
- 时间复杂度：$\mathrm{O}(N\sqrt{d}\log d)$。

【例】Hankson 的趣味题

- 算法 2：因为 x 是 d 的约数，所以 x 的质因子一定是 d 的质因子。对于 d 的每一个质因子 p，计算 x 可能包含多少个 p。
- 具体内容见《算法竞赛进阶指南》P144。

互质与欧拉函数

- $\forall a,b\in\mathbf{N}$，若 $\gcd(a,b)=1$，则称 a,b 互质。
- 欧拉函数：$1\sim N$ 中与 N 互质的数的个数称为欧拉函数，记为 $\varphi(N)$。
- 算术基本定理中 $N=p_1^{c_1}p_2^{c_2}\cdots p_m^{c_m}$，则

$$\varphi(N)=N\times\frac{p_1-1}{p_1}\times\frac{p_2-1}{p_2}\times\cdots\times\frac{p_m-1}{p_m}=N\times\prod_{p|N}(1-\frac{1}{p})$$

- 根据公式，我们在分解质因数过程中，可以求出欧拉函数。
- 证明[1]：容斥原理。令 $N={p_1}^{c_1}{p_2}^{c_2}$，则 $1\sim N$ 中 p 的倍数有 $p_1,2p_1,3p_1,\cdots,\frac{N}{p_1}p_1$ 个，p_2 的倍数有 $p_2,2p_2,3p_2,\cdots,\frac{N}{p_2}p_2$ 个，如果排除这 $\frac{N}{p_1}+\frac{N}{p_2}$ 个与 N 包含相同质因子的数，那么 p_1p_2 的倍数被排除了 2 次，需要加回来。故，

$$\varphi(N)=N-\frac{N}{p_1}-\frac{N}{p_2}+\frac{N}{p_1p_2}=N(1-\frac{1}{p_1})(1-\frac{1}{p_2})=N\times\frac{p_1-1}{p_1}\times\frac{p_2-1}{p_2}$$

[1] 自行了解。

互质与欧拉函数


```
int phi(int n)
{
    int ans = n;
    for(int i = 2; i * i <= n; ++i)
    {
        // 不要写 ans=ans*(i-1)/i; 因为可能运行时溢出
        if(n % i == 0) ans = ans / i * (i - 1);
        while(n % i == 0) n /= i;
    }
    if(n > 1) ans = ans / n * (n - 1);
    return ans;
}
```

约数

河南省实验中学信息技术组

欧拉函数性质

1. $\forall n > 1, 1 \sim n$ 中与 n 互质的数的和为 $n \times \frac{\varphi(n)}{2}$。
2. 若 a, b 互质，则 $\varphi(ab) = \varphi(a)\varphi(b)$。
3. $\varphi(n) = \varphi(\prod_{i=1}^{m} p_i^{c_i}) = \prod_{i=1}^{m} \varphi(p_i^{c_i})$
4. 设 p 为质数，若 $p|n$ 且 $p^2|n$，则 $\varphi(n) = \varphi(\frac{n}{p}) \times p$。
5. 设 p 为质数，若 $p|n$ 且 $p^2 \nmid n$，则 $\varphi(n) = \varphi(\frac{n}{p}) \times (p-1)$。
6. n 的所有约数的欧拉函数和为 n，$\sum_{d|n} \varphi(d) = n$。

欧拉函数性质证明[2]

1. 因为 $\gcd(n,x)=\gcd(n,n-x)$，所以与 n 互质的数是成对出现的而且它们的和为 n，故而得到性质❶。
2. 令 $a={p_1}^{c_1}{p_2}^{c_2}\cdots, b={q_1}^{d_1}{q_2}^{d_2}\cdots(p_i\neq q_j)$，则 $\varphi(ab)=ab\frac{p_1-1}{p_1}\frac{p_2-1}{p_2}\cdots\frac{q_1-1}{q_1}\frac{q2-1}{q2}\cdots=a\frac{p_1-1}{p_1}\frac{p_2-1}{p_2}\cdots b\frac{q_1-1}{q_1}\frac{q2-1}{q2}\cdots=\varphi(a)\varphi(b)$
3. 按照算数基本定理拆出的 ${p_i}^{c_i}$ 必然互斥，那么按照性质❷即可得证。
4. 令 $N={p_1}^{c_1}{p_2}^{c_2}\cdots$，则 $\frac{N}{p}={p_1}^{c_1-1}{p_2}^{c_2}\cdots$，那么 $\varphi(N)=N\frac{p_1-1}{p_1}\frac{p_2-1}{p_2}\cdots$，$\varphi(N)=\frac{N}{p}\frac{p_1-1}{p_1}\frac{p_2-1}{p_2}\cdots$，显然有 $\varphi(n)=\varphi(\frac{n}{p_1})\times p_1$。
5. 令 $N=p_1{p_2}^{c_2}\cdots$，则 $\frac{N}{p}={p_2}^{c_2}\cdots$，那么 $\varphi(N)=N\frac{p_1-1}{p_1}\frac{p_2-1}{p_2}\cdots$，$\varphi(N)=\frac{N}{p}\frac{p_2-1}{p_2}\cdots$，显然有 $\varphi(n)=\varphi(\frac{n}{p_1})\times(p_1-1)$。

[2] 自行了解。

【例】最大公约数和

【题目描述】

给定一个整数 n，求

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

【输入格式】

第一行，包含一个正整数 $n(n \le 2^{31}-1)$。

【输出格式】

一行一个整数，表示所求的答案。

【样例输入】

```
6
```

【样例输出】

```
15
```

【样例解释】

$\gcd(1,6)=1, \gcd(2,6)=2, \gcd(3,6)=3, \gcd(4,6)=2, \gcd(5,6)=1, \gcd(6,6)=6$，它们的和为 15。

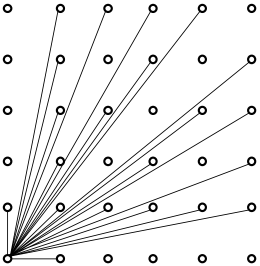
【例】最大公约数和

- 观察到，$\gcd(i,n)=1$ 的数有 $\varphi(n)$ 个，对答案的贡献为 $1\times\varphi(n)$。
- 如果 $\gcd(i,n)=2$，那么有 $\gcd(\frac{i}{2},\frac{n}{2})=1$，这样的数有 $\varphi(\frac{n}{2})$ 个，对答案的贡献为 $2\times\varphi(\frac{n}{2})$。
- 如果 $\gcd(i,n)=3$，那么有 $\gcd(\frac{i}{3},\frac{n}{3})=1$，这样的数有 $\varphi(\frac{n}{3})$ 个，对答案的贡献为 $3\times\varphi(\frac{n}{3})$。
- 依此类推，答案显然为 $\sum\limits_{k|n} k\times\varphi(\frac{n}{k})$。
- 枚举 n 的所有约数，按照上述公式累和即可。
- 时间复杂度：$O(\sqrt{N})$。

【例】可见的点

在一个平面直角坐标系内，以 $(0,0)$ 为左下角、(n,n) 为右上角的矩形中，除了 $(0,0)$ 之外，每个坐标上都插着一个钉子。

求在原点 $(0,0)$，能看到多少个钉子。一个钉子能被看到当且仅当它和原点的连线上没有其他钉子。例如，点 $(4,2)$ 就是不可见的，因为它与原点的连线会通过点 $(2,1)$；但是 $(2,3)$ 是可见的。数据范围 $1 \le n \le 10^6$。



【例】可见的点

【输入格式】

第一行，包含一个正整数 n。。

【输出格式】

一行一个整数，表示可见的点数。

【样例 1 输入】

```
4
```

【样例 1 输出】

```
13
```

【样例 2 输入】

```
231
```

【样例 2 输出】

```
32549
```

【例】可见的点

- 除了 (1,0) 、(0,1)、(1,1) 这三个钉子之外，一个钉子要想被看到，当且仅当 $1 \le x, y \le N, x \ne y$ 并且 $\gcd(x, y) = 1$，即纵横坐标互质。
- 通过观察可以发现钉子关于 $(0, 0) \to (N, N)$ 对称，所以我们可以先考虑一半，例如上半部分 $1 \le x < y \le N$。对于每个 $2 \le y \le N$，我们需要统计有多少个 x 满足 $1 \le x < y$ 并且 $\gcd(x, y) = 1$，这样的 x 的数量恰好是 $\varphi(y)$。
- 所以答案就是 $3 + 2 \times \sum\limits_{y=2}^{N} \varphi(y)$，问题就变成求 $2 \sim N$ 每个数的欧拉函数值。
- 利用埃氏筛和欧拉函数计算公式，可以在 $O(N \log\log N)$ 的时间内解决。

```
for(int i = 2; i <= n; ++i) phi[i] = i;
for(int i = 2; i <= n; ++i)
    if(phi[i] == i) // i 是质数
        for(int j = 1; j <= n / i; ++j)
            phi[i * j] = phi[i * j] / i * (i - 1);
s[1] = 0;
for(int i = 2; i <= n; ++i) s[i] = s[i - 1] + phi[i];
cout << 3 + 2 * s[n];
```

【例】可见的点

- 优化：利用线性筛法和欧拉函数性质在 O(N) 的时间内得出 $2 \sim N$ 中每个数的欧拉函数值。
- 欧拉函数性质 4 和 5：
  - 设 p 为质数，若 $p|n$ 且 $p^2|n$，则 $\varphi(n) = \varphi(\frac{n}{p}) \times p$。
  - 设 p 为质数，若 $p|n$ 且 $p^2 \nmid n$，则 $\varphi(n) = \varphi(\frac{n}{p}) \times (p-1)$。
- 线性筛法中，每个数只会被其最小质因子 p 筛一次，我们可以在筛选时从 $\varphi(\frac{n}{p})$ 递推到 $\varphi(n)$。

【例】可见的点

```cpp
int v[N], p[N], phi[N];
memset(v, 0, sizeof(v));
int m = 0;
for(int i = 2; i <= n; ++i)
{
    if(v[i] == 0) v[i] = i, p[++m] = i, phi[i] = i - 1;
    for(int j = 1; j <= m; ++j)
    {
        if(v[i] < p[j]) break; // i 的最小质因子比 p[j] 小则 i*p[j] 被 v[i] 筛
        if(i * p[j] > n) break; // 或者待筛数超出 n 的范围
        v[i * p[j]] = p[j]; // p[j] 是合数 i*p[j] 的最小质因子
        if(i % p[j] == 0) phi[i * p[j]] = phi[i] * p[j]; // 性质 4
        else phi[i * p[j]] = phi[i] * (p[j] - 1); // 性质 5
    }
}
s[1] = 0;
for(int i = 2; i <= n; ++i) s[i] = s[i - 1] + phi[i];
cout << 3 + 2 * s[n];
```

练习

- 反素数 (COGS 3147)
- Sumdiv(COGS 2691)
- 余数之和 (COGS 3149)
- 最大公约数和最小公倍数问题 [NOIP 2001](COGS 1417)
- Hankson 的趣味题 [NOIP 2009](COGS 405)
- 最大公约数和 (COGS 931)
- 可见的点 (COGS 3434)
- GCD(COGS 2602)
- 最大公约数 (COGS 3945)