数论入门选讲

天吾 (krydom)

清华大学，交叉信息院

3 Aug 2020

课前闲谈

整除

整除：若 $a = bk$，其中 a, b, k 都是整数，则 b 整除 a，记做 $b|a$。
也称 b 是 a 的约数（因数），a 是 b 的倍数

整除

整除：若 $a = bk$，其中 a, b, k 都是整数，则 b 整除 a，记做 $b|a$。
也称 b 是 a 的约数（因数），a 是 b 的倍数

显而易见的性质：
1 整除任何数，任何数都整除 0
若 $a|b, a|c$，则 $a|(b+c), a|(b-c)$
若 $a|b$，则对任意整数 c，$a|bc$
传递性：若 $a|b, b|c$，则 $a|c$

[CF 762A] k-th divisor

求 n 的第 k 小的约数。如果不存在输出 -1
$1 \le n \le 10^{15}, 1 \le k \le 10^9$

[CF 762A] k-th divisor

求 n 的第 k 小的约数。如果不存在输出 -1
$1 \leq n \leq 10^{15}, 1 \leq k \leq 10^9$

分析：注意到约数总是成对出现：
若 k 是 n 的约数, 则 (n/k) 也是 n 的约数。
在一对约数中，必有一个不大于 $\sqrt{n}$，另一个不小于 $\sqrt{n}$。
因此枚举 $1..\sqrt{n}$ 就能求出 n 的所有约数。

质数和合数

若大于 1 的正整数 p 仅有两个因子 1 和 p，则称 p 是一个质数（素数）。
否则，若 $p > 1$，则称 p 是一个合数。
1 不是质数也不是合数

质数和合数

若大于 1 的正整数 p 仅有两个因子 1 和 p，则称 p 是一个质数（素数）。
否则，若 $p > 1$，则称 p 是一个合数。
1 不是质数也不是合数

若 n 是一个合数，则 n 至少有 1 个质因子。因此其中最小的质因子一定不大于 $\sqrt{n}$
质数有无穷多个。不大于 n 的质数约有 $n/\ln n$ 个。

质数和合数

若大于 1 的正整数 p 仅有两个因子 1 和 p，则称 p 是一个质数（素数）。
否则，若 $p > 1$，则称 p 是一个合数。
1 不是质数也不是合数

若 n 是一个合数，则 n 至少有 1 个质因子。因此其中最小的质因子一定不大于 $\sqrt{n}$
质数有无穷多个。不大于 n 的质数约有 $n/\ln n$ 个。

唯一分解定理：把正整数 n 写成质数的乘积
（即 $n = p_1 p_2 p_3 ... p_k$，其中 p_i 为质数且单调不减），
这样的表示是唯一的。

[CF 776B] Sherlock and his girlfriend

n 个点，标号 $2 ... n+1$，
给这些点染色，要求若 a 是 b 的质因子，则 a 和 b 的颜色不同。
求一种颜色数最少的方案
$n \leq 1000$

[CF 776B] Sherlock and his girlfriend

n 个点，标号 $2...n+1$，
给这些点染色，要求若 a 是 b 的质因子，则 a 和 b 的颜色不同。
求一种颜色数最少的方案
$n \leq 1000$

分析：注意到这是二分图，一边是质数，一边是合数。
把质数都染成 1，合数都染成 2 即可。

质因数分解

利用 n 最多只有 1 个 $> \sqrt{n}$ 的质因子，
可以得到一个 $O(\sqrt{n})$ 的质因数分解算法。

```
vector<int> factor(int x) {
    vector<int> ret;
    for (int i = 2; i * i <= x; ++i)
        while (x % i == 0) {
            ret.push_back(i);
            x /= i;
        }
    if (x > 1) ret.push_back(x);
    return ret;
}
```

带余除法、同余

对于整数 $a, b, b > 0$，则存在唯一的整数 q, r，满足 $a = bq + r$，
其中 $0 \leq r < b$。
其中称 q 为商、r 为余数。

带余除法、同余

对于整数 $a, b, b > 0$，则存在唯一的整数 q, r，满足 $a = bq + r$，
其中 $0 \le r < b$。
其中称 q 为商、r 为余数。

余数用 $a \ mod \ b$ $(a\%b)$ 表示。
若两数 a, b 除以 c 的余数相等，则称 a,b 模 c 同余，记做 $a \equiv b(mod \ c)$。

带余除法、同余

对于整数 $a, b, b > 0$，则存在唯一的整数 q, r，满足 $a = bq + r$，
其中 $0 \leq r < b$。
其中称 q 为商、r 为余数。

余数用 $a \ mod \ b$ $(a\%b)$ 表示。
若两数 a, b 除以 c 的余数相等，则称 a,b 模 c 同余，记做 $a \equiv b(mod\ c)$。

性质：$a \equiv b(mod\ c)$ 与 $c|(a-b)$ 等价
推论：若 $a \equiv b(mod\ c), d|c$，则 $a \equiv b(mod\ d)$

最大公约数

设 a, b 是不都为 0 的整数，c 为满足 $c|a$ 且 $c|b$ 的最大整数，
则称 c 是 a, b 的最大公约数
记为 $GCD(a, b)$ 或 (a, b)

类似地可以定义多个数的最大公约数
GCD=Greatest Common Divisor
求 GCD 的一般公式：质因数分解

$$a = p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$$

$$b = p_1^{b_1} p_2^{b_2} \dots p_k^{b_k}$$

$$(a,b) = p_1^{\min(a_1,b_1)} p_2^{\min(a_2,b_2)} \dots p_k^{\min(a_k,b_k)}$$

最大公约数

一些性质:

$(a, a) = (0, a) = a$

若 $a|b$，则 $(a, b) = a$

$(a, b) = (a, a + b) = (a, ka + b)$

$(ka, kb) = k \cdot (a, b)$

$(a, b, c) = ((a, b), c)$

若 $(a, b) = 1$，则称 a, b 互质（互素）

互质的两个数往往有很好的性质

[CF 664A] Complicated GCD

求 $gcd(a, a+1, a+2, ..., b)$
$1 \leq a \leq b \leq 10^{100}$

[CF 664A] Complicated GCD

求 $gcd(a, a+1, a+2, ..., b)$
$1 \leq a \leq b \leq 10^{100}$

分析：注意到 $gcd(a, a+1) = 1$
因此 $a < b$ 时答案为 1，否则答案为 a

[CF 757B] Bash's Big Day

给定 n 个正整数 $\{a_i\}$
求一个子集 S，满足 $gcd(S_1, ..., S_k) > 1$，同时 $|S|$ 尽可能大。
$1 \leq n, ai \leq 10^5$

[CF 757B] Bash's Big Day

给定 n 个正整数 $\{a_i\}$
求一个子集 S，满足 $gcd(S_1, ..., S_k) > 1$，同时 $|S|$ 尽可能大。
$1 \leq n, ai \leq 10^5$

分析：$gcd > 1$，说明存在一个正整数 $d > 1$，满足 d 整除 S 内的所有元素。
枚举 $d = 2...max\{a_i\}$ 并统计答案
设 $V = max\{a_i\}$ 则复杂度为 $O(V \ln V)$。

欧几里得算法

又称辗转相除法
迭代求两数 gcd 的做法
由 $(a, b) = (a, ka + b)$ 的性质： $gcd(a, b) = gcd(b, a \ mod \ b)$

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

容易证明这么做的复杂度是 $O(\log n)$

裴蜀定理

裴蜀定理：
设 $(a, b) = d$，则对任意整数 x, y，有 $d|(ax + by)$ 成立；
特别地，一定存在 x, y 满足 $ax + by = d$

裴蜀定理

裴蜀定理:
设 $(a, b) = d$，则对任意整数 x, y，有 $d|(ax + by)$ 成立;
特别地，一定存在 x, y 满足 $ax + by = d$

等价的表述：不定方程 $ax + by = c(a, b, c$ 为整数) 有解的充要条件为 $(a, b)|c$
推论：a, b 互质等价于 $ax + by = 1$ 有解

扩展欧几里德算法

考虑如何求得 $ax + by = d$ 的一个解。这里 $d = (a, b)$
考虑使用欧几里德算法的思想，令 $a = bq + r$，其中 $r = a \ mod\ b$;
递归求出 $bx + ry = d$ 的一个解。

扩展欧几里德算法

考虑如何求得 $ax + by = d$ 的一个解。这里 $d = (a, b)$
考虑使用欧几里德算法的思想，令 $a = bq + r$，其中 $r = a \bmod b$;
递归求出 $bx + ry = d$ 的一个解。

设求出 $bx + ry = d$ 的一个解为 $x = x_0, y = y_0$，考虑如何把它变形成 $ax + by = d$ 的解。
将 $a = bq + r$ 代入 $ax + by = d$，化简得 $b(xq + y) + rx = d$
我们令 $xq + y = x_0, x = y_0$，则上式成立
故 $x = y_0, y = x_0 - y_0 q$ 为 $ax + by = d$ 的解

扩展欧几里德算法

考虑如何求得 $ax + by = d$ 的一个解。这里 $d = (a, b)$
考虑使用欧几里德算法的思想，令 $a = bq + r$，其中 $r = a \bmod b$;
递归求出 $bx + ry = d$ 的一个解。

设求出 $bx + ry = d$ 的一个解为 $x = x_0, y = y_0$，考虑如何把它变形成 $ax + by = d$ 的解。
将 $a = bq + r$ 代入 $ax + by = d$，化简得 $b(xq + y) + rx = d$
我们令 $xq + y = x_0, x = y_0$，则上式成立
故 $x = y_0, y = x_0 - y_0 q$ 为 $ax + by = d$ 的解

边界情况：$b = 0$ 时，令 $x = 1, y = 0$

扩展欧几里德算法

```
// a * x + b * y = gcd(a, b)
void exgcd(int a, int b, int &x, int &y) {
    if (b == 0) {
        x = 1, y = 0;
        return;
    }
    int q = a / b, r = a % b;
    exgcd(b, r, y, x);
    y -= q * x;
}
```

扩展欧几里德算法

怎么求 $ax + by = c$ 的所有解?

扩展欧几里德算法

怎么求 $ax + by = c$ 的所有解?

先用 exgcd 求出任意一个解 $x = x_0, y = y_0$
再求出 $ax + by = 0$ 的最小的解
$x = d_x = b/(a, b), y = d_y = -a/(a, b)$
所有解就是 $x = x_0 + kd_x, y = y_0 + kd_y$, k 取任意整数

线性组合

给定整数 $\{x_1, x_2, x_3, ..., x_n\}$ 和 k

求任意一组整数 $\{a_1, a_2, a_3, ..., a_n\}$
满足 $a_1x_1 + a_2x_2 + a_3x_3 + ... + a_nx_n = k$，或返回无解。

线性组合

分析：
显然仅当 $(x_1, x_2, ..., x_n)|k$ 时有解。

线性组合

分析：
显然仅当 $(x_1, x_2, ..., x_n)|k$ 时有解。

若 $n = 1$ 直接令 $a_1 = k/x_1$ 构造解。
否则用 (x_{n-1}, x_n) 代替这两个数，递归构造解；
回溯时调用 exgcd(x_{n-1}, x_n)，
用 x_{n-1}, x_n 的线性组合替代 (x_{n-1}, x_n)。

逆元

若 $ax \equiv 1(mod\ b)$，则称 x 是 a 关于模 b 的逆元，
常记做 a^{-1}。

逆元

若 $ax \equiv 1(mod\ b)$，则称 x 是 a 关于模 b 的逆元，常记做 a^{-1}。

回忆同余的性质。上式等价于 $ax + by = 1$
如何求逆元？等价于解方程 $ax + by = 1$

逆元

若 $ax \equiv 1(mod\ b)$，则称 x 是 a 关于模 b 的逆元，
常记做 a^{-1}。

回忆同余的性质。上式等价于 $ax + by = 1$
如何求逆元？等价于解方程 $ax + by = 1$

因此逆元不一定存在：
存在的充要条件为 $(a, b) = 1$
推论：p 是质数，p 不整除 a，则 a 模 p 的逆元存在。

逆元

结论：在 $[0, b)$ 的范围内，a 关于模 b 的逆元 (若存在) 是唯一的。

逆元

结论：在 $[0, b)$ 的范围内，a 关于模 b 的逆元 (若存在) 是唯一的。

证明：
反证法，若 a 有两个逆元 $0 < x_1 < x_2 < b$，
即 $ax_1 \equiv ax_2 \equiv 1 (mod\ b)$，
那么有 $b|a(x_2 - x_1)$ 成立
又由于 $(a, b) = 1$，因此 $b|(x_2 - x_1)$。
其中 $0 < x_2 - x_1 < b$，产生了矛盾。

逆元

```
// 利用exgcd求逆元
int inv(int a, int b) {
    int x, y;
    exgcd(a, b, x, y);
    return x;
}
```

线性求逆元

如何 $O(n)$ 求 $1 \sim n$ 模质数 p 的逆元?

线性求逆元

如何 $O(n)$ 求 $1 \sim n$ 模质数 p 的逆元?

方法一：递推
假设现在要求 i 的逆元

线性求逆元

如何 $O(n)$ 求 $1 \sim n$ 模质数 p 的逆元?

方法一：递推
假设现在要求 i 的逆元
考虑带余除法，设 $p = iq + r$，则有 $iq + r \equiv 0(mod\ p)$
注意到 p 是质数，因此 r 不为 0，r 的逆元存在
等式两边乘 $i^{-1}r^{-1}$，得到 $qr^{-1} + i^{-1} \equiv 0(mod\ p)$
因此 $i^{-1} \equiv -qr^{-1} \equiv -(p/i)(p\ mod\ i)^{-1}(mod\ p)$

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

线性求逆元

方法二：倒推

先求 $n!$ 的逆元 (exgcd, 或者后面提到的快速幂)

线性求逆元

方法二：倒推

先求 $n!$ 的逆元 (exgcd, 或者后面提到的快速幂)
然后利用 $((k-1)!)^{-1} \equiv k \cdot (k!)^{-1} (mod\ p)$
倒推求出 $1! \ldots (n-1)!$ 的逆元

线性求逆元

方法二：倒推

先求 $n!$ 的逆元 (exgcd, 或者后面提到的快速幂)
然后利用 $((k-1)!)^{-1} \equiv k \cdot (k!)^{-1} (mod\ p)$
倒推求出 $1! ... (n-1)!$ 的逆元
再利用 $k^{-1} \equiv (k-1)! \cdot (k!)^{-1} (mod\ p)$
就可以求出 $1 ... n$ 的逆元了

组合数取模

回答 T 次询问

每次询问 $C(n, k)$ mod 998244353(一个质数)

$T \leq 10^5, 0 \leq k \leq n \leq 10^7$

组合数取模

回答 T 次询问
每次询问 $C(n, k)$ mod 998244353(一个质数)
$T \leq 10^5, 0 \leq k \leq n \leq 10^7$

分析：$C(n, k) = n!/(k!(n-k)!)$
线性求逆，预处理 $n!$ 以及 $n!$ 的逆元
$O(1)$ 回答询问

线性同余方程

形如 $ax \equiv c(mod\ b)$ 的方程，称为线性同余方程。
等价于 $ax + by = c$；因此有解条件为 $(a, b)|c$

线性同余方程

形如 $ax \equiv c(mod\ b)$ 的方程，称为线性同余方程。
等价于 $ax + by = c$；因此有解条件为 $(a, b)|c$

若 $(a, b) = 1$，则 x 有唯一解 $x \equiv a^{-1}c(mod\ b)$。
否则设 $(a, b) = d, a = a'd, b = b'd, c = c'd$
那么有 $a'x + b'y = c'$，即 $a'x \equiv c'(mod\ b')$
这里 $(a', b') = 1$，因此有 $x \equiv (a')^{-1}c'(mod\ b')$

线性同余方程

形如 $ax \equiv c(mod\ b)$ 的方程，称为线性同余方程。
等价于 $ax + by = c$；因此有解条件为 $(a, b)|c$

若 $(a, b) = 1$，则 x 有唯一解 $x \equiv a^{-1}c(mod\ b)$。
否则设 $(a, b) = d, a = a'd, b = b'd, c = c'd$
那么有 $a'x + b'y = c'$，即 $a'x \equiv c'(mod\ b')$
这里 $(a', b') = 1$，因此有 $x \equiv (a')^{-1}c'(mod\ b')$

综上，任意的线性同余方程总可以判定为无解，或化为 $x \equiv a(mod\ m)$ 的形式。

线性同余方程组

考虑形如 $x \equiv a_i (mod\ m_i)$ 的若干方程联立得到的方程组，如：

$x \equiv 2(mod\ 3)$ .... (1)

$x \equiv 3(mod\ 5)$ .... (2)

$x \equiv 5(mod\ 7)$ .... (3)

线性同余方程组

考虑形如 $x \equiv a_i(mod\ m_i)$ 的若干方程联立得到的方程组，如：
$x \equiv 2(mod\ 3)$ .... (1)
$x \equiv 3(mod\ 5)$ .... (2)
$x \equiv 5(mod\ 7)$ .... (3)

下面是一种可行的解法：
由 (1) 设 $x = 3y + 2$, 代入 (2) 得到 $3y + 2 \equiv 3(mod\ 5)$, 解得 $y \equiv 2(mod\ 5)$
设 $y = 5z + 2$, 代入 (3) 得到 $3(5z + 2) + 2 \equiv 5(mod\ 7)$, 解得 $z \equiv 4(mod\ 7)$
设 $z = 7k + 4$, 则 $x = 3(5(7k + 4) + 2) + 2 = 105k + 68$
因此 $x \equiv 68(mod\ 105)$

中国剩余定理

中国剩余定理断言，对于同余方程组 $x \equiv a_i (mod\ m_i)(i = 1...n)$，若 m_i 两两互质，则 x 在 $mod\ M$ 下有唯一解。这里 $M = m_1 m_2 ... m_n$

中国剩余定理

中国剩余定理断言，对于同余方程组 $x \equiv a_i (mod\ m_i)(i = 1...n)$，若 m_i 两两互质，则 x 在 $mod\ M$ 下有唯一解。这里 $M = m_1 m_2 ... m_n$

中国剩余定理同时也给出了构造解的方法：

令 $M = m_1 m_2 ... m_n, M_i = M/m_i$
显然 $(M_i, m_i) = 1$，所以 M_i 关于模 m_i 的逆元存在。把这个逆元设为 t_i
于是有：$M_i t_i \equiv 1(mod\ m_i), M_i t_i \equiv 0(mod\ m_j)(j \neq i)$
进一步：$a_i M_i t_i \equiv a_i (mod\ m_i), a_i M_i t_i \equiv 0(mod\ m_j)(j \neq i)$
因此解为 $x \equiv a_1 M_1 t_1 + a_2 M_2 t_2 + ... + a_n M_n t_n (mod\ M)$

中国剩余定理

```
// Chinese Remainder Theorem
int CRT(const int a[], const int m[], int n) {
    int M = 1, ret = 0;
    for (int i = 1; i <= n; ++i) M *= m[i];
    for (int i = 1; i <= n; ++i) {
        int Mi = M / m[i], ti = inv(Mi, m[i]);
        ret = (ret + a[i] * Mi * ti) % M;
    }
    return ret;
}
```

中国剩余定理

今有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二，问物几何?

$x \equiv 2(mod\ 3)$ ...... (1)
$x \equiv 3(mod\ 5)$ ...... (2)
$x \equiv 2(mod\ 7)$ ...... (3)

中国剩余定理

用中国剩余定理构造解。

$a_i = \{2, 3, 2\}, mi = \{3, 5, 7\}, M = 3 * 5 * 7 = 105$
$M_i = \{105/3, 105/5, 105/7\} = \{35, 21, 15\}$
$m_i = \{inv(35, 3), inv(21, 5), inv(15, 7)\} = \{2, 1, 1\}$
$x \equiv 2 * (35 * 2) + 3 * (21 * 1) + 2 * (15 * 1)$
$x \equiv 233 \equiv 23(mod\ 105)$

组合数取模 2

回答 T 次询问，
每次询问 $C(n,k) \ mod \ 1029471131$
$1029471131 = 13 * 317 * 249811$
$T \leq 10^5, 0 \leq k \leq n \leq 10^{18}$

组合数取模 2

回答 T 次询问，
每次询问 $C(n,k)$ mod 1029471131
$1029471131 = 13 * 317 * 249811$
$T \le 10^5, 0 \le k \le n \le 10^{18}$

分析：分别将 $C(n,k)$ 对 $13, 317, 249811$ 取模，再用中国剩余定理合并。
如何求 $C(n,k)$ mod p，其中 p 为较小的质数?

组合数取模 2

*Lucas 定理：设 p 为质数，则

$$C_n^k \equiv C_{\lfloor n/p \rfloor}^{\lfloor k/p \rfloor} \cdot C_{(n \bmod p)}^{(k \bmod p)} \pmod p$$

组合数取模 2

*Lucas 定理：设 p 为质数，则

$$C_n^k \equiv C_{\lfloor n/p \rfloor}^{\lfloor k/p \rfloor} \cdot C_{(n \bmod p)}^{(k \bmod p)} \pmod p$$

递归使用 Lucas 定理，把 $C(n,k)$ 中的 n,k 缩小到 $< p$
注意特判 $n < k$ 时 $C(n,k) = 0$

使用 $C(n,k) = n!/(k!(n-k)!)$
预处理阶乘、逆元即可

欧拉函数

欧拉函数 φ (Euler's totient function)
$\varphi(n)$ 定义为 $[1, n]$ 中与 n 互质的数的个数
例：$\varphi(1) = 1, \varphi(2) = 1, \varphi(6) = 2, \varphi(8) = 4$

欧拉函数

欧拉函数 φ (Euler's totient function)
$\varphi(n)$ 定义为 $[1, n]$ 中与 n 互质的数的个数
例：$\varphi(1) = 1, \varphi(2) = 1, \varphi(6) = 2, \varphi(8) = 4$
推论：若 p 为质数，则 $\varphi(p) = p - 1$

欧拉函数

欧拉函数 φ (Euler's totient function)
$\varphi(n)$ 定义为 $[1, n]$ 中与 n 互质的数的个数
例：$\varphi(1) = 1, \varphi(2) = 1, \varphi(6) = 2, \varphi(8) = 4$
推论：若 p 为质数，则 $\varphi(p) = p - 1$

欧拉函数是积性函数：
若 $(a, b) = 1$，则 $\varphi(ab) = \varphi(a)\varphi(b)$

欧拉函数

证明：令 $S(n)$ 为 $[1, n]$ 中与 n 互质的数的集合。
分别任取 $S(a), S(b)$ 中的元素 a_0, b_0；

考虑线性同余方程组 $\{x \equiv a_0 (mod\ a), x \equiv b_0 (mod\ b)\}$
由中国剩余定理，有唯一解 $x \equiv c_0 (mod\ ab)$
$(c_0, a) = (c_0\ \ mod\ a, a) = (a_0, a) = 1$; 同理 $(c_0, b) = 1$
因此 $(c_0, ab) = 1$， $c_0 \in S(ab)$
反过来，任取 $S(ab)$ 中的元素 c_0，那么令 $a_0 = c_0\ \ mod\ a, b_0 = c_0\ \ mod\ b,$
有 $a_0 \in S(a), b_0 \in S(b)$

以上是 $S(a) \times S(b)$ 到 $S(ab)$ 的一个双射
因此 $|S(ab)| = |S(a)||S(b)|$，即 $\varphi(ab) = \varphi(a)\varphi(b)$

欧拉函数

若 $n = p^k$，p 为质数，则 $\varphi(n) = n(1 - 1/p)$

欧拉函数

若 $n = p^k$，p 为质数，则 $\varphi(n) = n(1 - 1/p)$
证明：若 $(x, p^k) > 1$，则有 $p|x$ 成立。
x 共有 n/p 个，因此 $\varphi(n) = n - n/p = n(1 - 1/p)$

欧拉函数

若 $n = p^k$，p 为质数，则 $\varphi(n) = n(1 - 1/p)$
证明：若 $(x, p^k) > 1$，则有 $p|x$ 成立。
x 共有 n/p 个，因此 $\varphi(n) = n - n/p = n(1 - 1/p)$

若 n 所有不同的质因子为 $p_1, p_2, ..., p_k$，
则 $\varphi(n) = n(1 - 1/p_1)(1 - 1/p_2)...(1 - 1/p_k)$
证明：φ 是积性函数。把 n 拆成 $p_i^{a_i}$ 的乘积可立即得证。

欧拉函数

因此得到一种基于质因数分解求欧拉函数的算法。

```
int phi(int x) {
    int ret = x;
    for (int i = 2; i * i <= x; ++i)
        if (x % i == 0) {
            while (x % i == 0) x /= i;
            ret = ret / i * (i - 1);
        }
    if (x > 1) ret = ret / x * (x - 1);
    return ret;
}
```

[bzoj2705][SDOI2012]Longge 的问题

求 $\sum_{i=1}^{n} gcd(i, n), n \leq 2^{32}$

[bzoj2705][SDOI2012]Longge 的问题

求 $\sum_{i=1}^{n} gcd(i,n), n \le 2^{32}$

分析：注意到 $gcd(i,n)$ 一定是 n 的约数，而 n 的约数只有 $O(\sqrt{n})$ 个。
考虑枚举 n 的约数 (设为 d)，再求出满足 $gcd(i,n) = d$ 的 i 有多少个。

[bzoj2705][SDOI2012]Longge 的问题

假设 $gcd(i, n) = d$，则 $gcd(i/d, n/d) = 1$
注意到 $i/d \leq n/d$。因此共有 $\varphi(n/d)$ 个 i 满足条件。

质因数分解求出每个 $\varphi(n/d)$

[bzoj 2186][Sdoi2008] 沙拉公主的困惑

给定一质数 p，回答 T 组询问
每组询问 $[1, n!]$ 中与 $m!$ 互质的数的个数，结果对 p 取模。
$1 \le m \le n \le 10^7, n < p < 10^9, T \le 10^4$

[bzoj 2186][Sdoi2008] 沙拉公主的困惑

分析：注意到 $gcd(a,b) = gcd(a \ mod \ b, b)$，又 $m!|n!$
则有 $ans = (n!/m!) \cdot \varphi(m!)$

[bzoj 2186][Sdoi2008] 沙拉公主的困惑

分析：注意到 $gcd(a,b) = gcd(a \ mod \ b, b)$，又 $m!|n!$
则有 $ans = (n!/m!) \cdot \varphi(m!)$

由 $\varphi(n) = n(1-1/p_1)(1-1/p_2)...(1-1/p_k)$
$ans = n!(1-1/p_1)(1-1/p_2)...(1-1/p_k)$
这里 $p_1...p_k$ 为 $m!$ 的所有质因子，即不大于 m 的所有素数。

设 $f(n) = n!$，$g(m) = (1-1/p_1)(1-1/p_2)...(1-1/p_k)$
都能在 $O(n)$ 时间内预处理 (利用线性筛和线性求逆元)
则 $ans = f(n)g(m)$，$O(1)$ 回答询问。

欧拉定理

欧拉定理：若 $(a, n) = 1$，则 $a^{\varphi(n)} \equiv 1(mod\ n)$

欧拉定理

欧拉定理：若 $(a, n) = 1$，则 $a^{\varphi(n)} \equiv 1(mod\ n)$

证明：任取 $\varphi(n)$ 个与 n 互质、且互不同余的整数构成一个集合
$S = \{x_1, x_2, ..., x_{\varphi(n)}\}$（这样的集合称为模 n 的简化剩余系)。
考虑集合 $S' = \{ax_1, ax_2, ..., ax_{\varphi(n)}\}$，$S'$ 也是一个简化剩余系。
若将 S 和 S' 的每个元素对 n 取模，则有 $S = S'$。
所以 $x_1x_2...x_{\varphi(n)} \equiv ax_1ax_2...ax_{\varphi(n)}(mod\ n)$
将所有 x 约去即得 $a^{\varphi(n)} \equiv 1(mod\ n)$

欧拉定理

用欧拉定理求逆元：$a \cdot a^{\varphi(n)-1} \equiv 1(mod\ n)$
特别地，若 n 是质数：$a \cdot a^{n-2} \equiv 1(mod\ n)$
快速幂即可

```
int powermod(int a, int b, int n) {
    int ret = 1;
    while (b) {
        if (b & 1) ret = (long long)ret * a % n;
        a = (long long)a * a % n;
        b >>= 1;
    }
    return ret;
}
```

欧拉定理的应用

求 $a^b \ mod \ m$ 时，用欧拉定理缩小指数 b

求 a^b *mod* m 时，用欧拉定理缩小指数 b

例题：求 7^{222} *mod* 10
分析：$(7,10)=1$, 故 $7^{\varphi(10)}=7^4\equiv 1(mod\ 10)$
所以 $7^{222}\equiv 7^{222\ mod\ 4}=7^2=49\equiv 9(mod\ 10)$
即 7^{222} *mod* $10=9$

欧拉定理的应用

求 a^b *mod* m 时，用欧拉定理缩小指数 b

例题：求 7^{222} *mod* 10
分析：$(7,10)=1$, 故 $7^{\varphi(10)}=7^4\equiv 1(mod\ 10)$
所以 $7^{222}\equiv 7^{222\ mod\ 4}=7^2=49\equiv 9(mod\ 10)$
即 7^{222} *mod* $10=9$

$(a,m)>1$ 怎么办?

欧拉定理的拓展

定理：$a^{2\varphi(n)} \equiv a^{\varphi(n)} (mod\ n)$

欧拉定理的拓展

定理： $a^{2\varphi(n)} \equiv a^{\varphi(n)}(mod\ n)$

证明：首先我们提出一个引理：
引理一：设 p 为 n 的一个质因子，k 为 p 的次数；
则有 $\varphi(n) \geq k$ 成立。

欧拉定理的拓展

定理：$a^{2\varphi(n)} \equiv a^{\varphi(n)} (mod\ n)$

证明：首先我们提出一个引理：
引理一：设 p 为 n 的一个质因子，k 为 p 的次数；
则有 $\varphi(n) \geq k$ 成立。

引理的证明：
设 $n = p^k \cdot t$；则 $\varphi(n) = \varphi(p^k)\varphi(t) \geq \varphi(p^k) \geq k$

欧拉定理的拓展

定理：$a^{2\varphi(n)} \equiv a^{\varphi(n)} (mod\ n)$

欧拉定理的拓展

定理： $a^{2\varphi(n)} \equiv a^{\varphi(n)} (mod\ n)$

证明：设 $n = n_1 n_2$，其中 $n_1 = (a^{\infty}, n)$
则有 $(a, n_2) = (n_1, n_2) = 1$
由引理一可得 $n_1 | a^{\varphi(n)}$，
因此 $a^{\varphi(n)} \equiv a^{2\varphi(n)} \equiv 0 (mod\ n_1)...(1)$
又由 $(n_1, n_2) = 1$ 得 $\varphi(n) = \varphi(n_1)\varphi(n_2)$,
由欧拉定理 $a^{\varphi(n)} \equiv a^{2\varphi(n)} \equiv 1 (mod\ n_2)...(2)$
中国剩余定理合并 12 两式得证。

推论：当 $b \geq \varphi(n)$ 时， $a^b \equiv a^{b\ mod\ \varphi(n) + \varphi(n)} (mod\ n)$

[bzoj 3884] 上帝与集合的正确用法

令 $a_n = 2^{2^{2^{\cdots}}}$ (n 个 2) 求 $\lim\limits_{n\to\infty}(a_n \bmod m)$

$T(T \le 1000)$ 组询问，$m \le 10^7$

[bzoj 3884] 上帝与集合的正确用法

即解方程 $2^x \equiv x (mod\ m)$
由之前的结论有 $2^x \equiv 2^{x\ mod\ \varphi(m)+\varphi(m)} (mod\ m)$
那就转化成了求 $x\ mod\ \varphi(m)$

[bzoj 3884] 上帝与集合的正确用法

即解方程 $2^x \equiv x (mod\ m)$
由之前的结论有 $2^x \equiv 2^{x\ mod\ \varphi(m)+\varphi(m)} (mod\ m)$
那就转化成了求 $x\ mod\ \varphi(m)$

递归求解，至 $m=1$ 时终止即可。
可以证明递归的层数不大于 $2\log 2m$

线性筛与积性函数

考虑一个经典问题：求不大于 n 的所有素数。

线性筛与积性函数

考虑一个经典问题：求不大于 n 的所有素数。

Eratosthenes 筛法：

1. 初始时令列表 $A = \{2, 3, ..., n\}; p = 2$
2. 枚举所有 p 的倍数 (不包括 p)，并在 A 中删去这些数
3. 令 p 为 A 中的下一个数并跳转至 (2)。如果不存在下一个则结束。
4. 算法结束时，A 中剩下的数为不大于 n 的所有素数。

线性筛与积性函数

```
int sieve(int n, bool isprime[], int prime[]) {
    int tot = 0;
    for (int i = 2; i <= n; ++i) isprime[i] = 1;
    for (int i = 2; i <= n; ++i)
        if (isprime[i]) {
            prime[++tot] = i;
            for (int j = i + i; j <= n; j += i)
                isprime[j] = 0;
        }
    return tot;
}
```

时间复杂度： $O(n \log \log n)$
空间复杂度： $O(n)$

例题：求 $[1, 10^7]$ 内的所有素数。内存限制 1MB

例题：求 $[1, 10^7]$ 内的所有素数。内存限制 1MB

分析：直接做会爆空间。

例题：求 $[1, 10^7]$ 内的所有素数。内存限制 1MB

分析：直接做会爆空间。
把 $[1, 10^7]$ 分成 k 段分别求解；
对于区间 $[l, r]$，枚举不大于 $\sqrt{r}$ 的所有素数 p，在 $[l, r]$ 中筛去 p 的倍数。
需要预处理 $[1, \sqrt{n}]$ 的所有素数
时间复杂度：$O(k \cdot \sqrt{n} + n \log\log n)$
空间复杂度：$O(\sqrt{n} + n/k)$

线性筛与积性函数

如何 $O(n)$ 求 $1 \sim n$ 的素数?

线性筛与积性函数

如何 $O(n)$ 求 $1 \sim n$ 的素数?

把筛法做到 $O(n)$，每个合数必须只被筛去一次。

线性筛与积性函数

如何 $O(n)$ 求 $1 \sim n$ 的素数?

把筛法做到 $O(n)$，每个合数必须只被筛去一次。

考虑 dp 求 $1 \sim n$ 每个数的最小质因子 $f[]$
如果已经知道 $f[i]$，那么枚举 $1 \sim f[i]$ 的所有素数 p，就可以求得 $f[i \cdot p] = p$

如何 $O(n)$ 求 $1 \sim n$ 的素数?

把筛法做到 $O(n)$，每个合数必须只被筛去一次。

考虑 dp 求 $1 \sim n$ 每个数的最小质因子 $f[]$
如果已经知道 $f[i]$，那么枚举 $1 \sim f[i]$ 的所有素数 p，就可以求得 $f[i \cdot p] = p$
容易看出每个 $f[i]$ 只会被求一次
如果枚举至 i 时还未求出 $f[i]$，则 i 不存在小于 i 的质因子，即 i 为质数。

线性筛与积性函数

```
int sieve(int n, int f[], int prime[]) {
    int tot = 0;
    for (int i = 2; i <= n; ++i) {
        if (!f[i]) prime[++tot] = f[i] = i;
        for (int j = 1; j <= tot; ++j) {
            int t = i * prime[j];
            if (t > n) break;
            f[t] = prime[j];
            if (f[i] == prime[j]) break;
        }
    }
}
```

线性筛与积性函数

积性函数:
$f(1) = 1$;
若 $(a, b) = 1$, 则 $f(ab) = f(a)f(b)$

线性筛与积性函数

积性函数:
$f(1) = 1$;
若 $(a, b) = 1$, 则 $f(ab) = f(a)f(b)$

常见的积性函数:
$\varphi(n)$ 欧拉函数
$d(n)$ 约数个数
$d_k(n)$ 约数的 k 次幂和
$gcd(n, k)$，其中 k 给定的整数

线性筛与积性函数

给定积性函数 f，如何求 $f(1)...f(n)$ 的值?

线性筛与积性函数

给定积性函数 f，如何求 $f(1)...f(n)$ 的值?
利用积性：设 $n = p^k n'$，其中 p 是 n 最小的质因子，k 是 p 在 n 中的次数;
则有 $f(n) = f(p^k)f(n')$

只需特别考虑 $f(p^k)$；其它用线性筛递推即可。

线性筛与积性函数

给定积性函数 f，如何求 $f(1)...f(n)$ 的值?
利用积性：设 $n = p^k n'$，其中 p 是 n 最小的质因子，k 是 p 在 n 中的次数;
则有 $f(n) = f(p^k)f(n')$

只需特别考虑 $f(p^k)$；其它用线性筛递推即可。

以 φ 为例:
$\varphi(p) = p - 1$
$\varphi(p^k) = p \cdot \varphi(p^{k-1})(k > 1)$
因此
$\varphi(pq) = (p-1)\varphi(q)((p,q) = 1)$
$\varphi(p \cdot p^k q) = \varphi(p^{k+1})\varphi(q) = p \cdot \varphi(p^k q)$

线性筛与积性函数

```
int sieve(int n, int f[], int phi[], int prime[]) {
    int tot = 0; phi[1] = 1;
    for (int i = 2; i <= n; ++i) {
        if (!f[i]) {
            prime[++tot] = f[i] = i;
            phi[i] = i - 1;
        }
        for (int j = 1; j <= tot; ++j) {
            int t = i * prime[j];
            if (t > n) break;
            f[t] = prime[j];
            phi[t] = phi[i] * (prime[j] - (prime[j] < f[i]));
            if (prime[j] == f[i]) break;
        }
    }
}
```

[Luogu P2568]gcd

给定整数 N，求 $1 \le x, y \le N$ 且 $gcd(x, y)$ 为素数的数对 (x, y) 有多少对.
$N \le 10^7$

[Luogu P2568]gcd

给定整数 N，求 $1 \le x, y \le N$ 且 $gcd(x, y)$ 为素数的数对 (x, y) 有多少对.
$N \le 10^7$

分析：线性筛出不大于 N 的所有素数，枚举 $gcd(x, y)$(设为 p)，
问题转化为求 $(x, y) = p$ 的个数。

[Luogu P2568]gcd

给定整数 N，求 $1 \le x, y \le N$ 且 $gcd(x, y)$ 为素数的数对 (x, y) 有多少对.
$N \le 10^7$

分析：线性筛出不大于 N 的所有素数，枚举 $gcd(x, y)$(设为 p)，
问题转化为求 $(x, y) = p$ 的个数。
设 $x = x'p, y = y'p$，那么有 $(x, y) = 1$ 且 $1 \le x, y \le N/p$。
转化为求 $(x, y) = 1$ 且 $1 \le x, y \le n$ 的个数。

[Luogu P2568]gcd

给定整数 N，求 $1 \le x, y \le N$ 且 $gcd(x, y)$ 为素数的数对 (x, y) 有多少对.
$N \le 10^7$

分析：线性筛出不大于 N 的所有素数，枚举 $gcd(x, y)$(设为 p)，
问题转化为求 $(x, y) = p$ 的个数。
设 $x = x'p, y = y'p$，那么有 $(x, y) = 1$ 且 $1 \le x, y \le N/p$。
转化为求 $(x, y) = 1$ 且 $1 \le x, y \le n$ 的个数。

答案为 $2(\varphi(1) + ... + \varphi(n)) - 1$
线性筛筛出欧拉函数、预处理前缀和即可

某道题

求 $\sum_{i=1}^{n}\sum_{j=1}^{n}(-1)^{d(i*j)}$，其中 d(n) 表示 n 的约数个数。
$1 \le n \le 10^7$

某道题

求 $\sum_{i=1}^{n}\sum_{j=1}^{n}(-1)^{d(i*j)}$，其中 d(n) 表示 n 的约数个数。
$1 \leq n \leq 10^7$

分析：注意到 $(-1)^2 = 1$，因此 $(-1)^k = (-1)^{k \ mod \ 2}$
$d(n)$ 为奇数当且仅当 n 为完全平方数。

某道题

求 $\sum_{i=1}^{n}\sum_{j=1}^{n}(-1)^{d(i*j)}$，其中 d(n) 表示 n 的约数个数。
$1 \le n \le 10^7$

分析：注意到 $(-1)^2=1$，因此 $(-1)^k=(-1)^{k \ mod \ 2}$
$d(n)$ 为奇数当且仅当 n 为完全平方数。
因此只需求出所有的 $i \cdot j$ 中有多少个完全平方数。
考虑满足条件的 i, j 有什么特殊的性质。

某道题

令 $i = ab^2$，其中 a 不能被大于 1 的平方数整除。
容易看出这样的 a, b 是唯一的。

某道题

令 $i = ab^2$，其中 a 不能被大于 1 的平方数整除。
容易看出这样的 a, b 是唯一的。

设 $f(i) = a$; 则 $i \cdot j$ 为平方数等价于 $f(i) = f(j)$
因此只需求出 $f(1)..f(n)$ 的值，统计每种取值的个数就可以了。

某道题

令 $i = ab^2$，其中 a 不能被大于 1 的平方数整除。
容易看出这样的 a, b 是唯一的。

设 $f(i) = a$; 则 $i \cdot j$ 为平方数等价于 $f(i) = f(j)$
因此只需求出 $f(1)..f(n)$ 的值，统计每种取值的个数就可以了。
可以证明 f 是积性函数;
p 为质数时 $f(p^k) = p^{k \bmod 2}$
线性筛即可

Luogu 2152][SDOI2009]SuperGCD

求 $\gcd(a, b)$; $0 < a, b \leq 10^{10000}$

分析：如果直接用辗转相除法，就需要实现高精度除法，复杂度会有问题。

Luogu 2152][SDOI2009]SuperGCD

求 $\gcd(a, b)$; $0 < a, b \leq 10^{10000}$

分析：如果直接用辗转相除法，就需要实现高精度除法，复杂度会有问题。

考虑对 $/2$ 分类。
若 a, b 同为偶数，则 $(a, b) = 2 * (a/2, b/2)$
若 a, b 同为奇数，不妨设 $a > b$，则 $(a, b) = ((a - b)/2, b)$
若 a, b 一奇一偶，不妨设 a 为偶数，则 $(a, b) = (a/2, b)$

Luogu 2152][SDOI2009]SuperGCD

求 $\gcd(a, b)$; $0 < a, b \le 10^{10000}$

分析：如果直接用辗转相除法，就需要实现高精度除法，复杂度会有问题。

考虑对 $/2$ 分类。
若 a, b 同为偶数，则 $(a, b) = 2 * (a/2, b/2)$
若 a, b 同为奇数，不妨设 $a > b$，则 $(a, b) = ((a - b)/2, b)$
若 a, b 一奇一偶，不妨设 a 为偶数，则 $(a, b) = (a/2, b)$
同样只经过 log 次迭代，每次操作只有 $/2, *2, -$，都可以在 $O(\log a)$ 内完成

给出 n 个数，要求选出尽可能多的数，满足它们的最小公倍数不大于 m。
$1 \leq n, m \leq 10^6, 1 \leq a_i \leq 10^9$

[CF 632D] Longest Subsequence

给出 n 个数，要求选出尽可能多的数，满足它们的最小公倍数不大于 m。
$1 \leq n, m \leq 10^6, 1 \leq a_i \leq 10^9$

分析：设取的所有数都是 k 的约数，则这些数的 lcm 必然不大于 k。
对于 $[1, m]$ 中的每个数，统计 a 中有多少个数是它的约数即可。

[CF 582A] GCD Table

对一个长度为 n 的数列 a，
定义它的 GCD Table G 是一张 $n \times n$ 的二维表，其中 $G_{i,j} = gcd(a_i, a_j)$
现在乱序给出 G 中所有 n^2 个数，求原数列 a。
数据保证有解；输出任意一种方案即可。
$1 \leq n \leq 500, G_{i,j} \leq 10^9$

[CF 582A] GCD Table

对一个长度为 n 的数列 a，
定义它的 GCD Table G 是一张 $n \times n$ 的二维表，其中 $G_{i,j} = gcd(a_i, a_j)$
现在乱序给出 G 中所有 n^2 个数，求原数列 a。
数据保证有解；输出任意一种方案即可。
$1 \leq n \leq 500, G_{i,j} \leq 10^9$

分析：G 中最大的数一定也是 a 中最大的数。
G 中次大的数一定也是 a 次大的数。
第三、第四可能是由最大和次大的 gcd 产生的。

[CF 582A] GCD Table

那么就不难想到下面的算法：

1. 令 p 为 G 中最大的数。在 G 中删除 p，a 中加入 p。
2. 对于 a 中的所有其他数 (设为 q)，在 G 中删除 2 个 $gcd(p,q)$。
3. 若 G 为空则结束；否则回到 (1)。

时间复杂度 $O(n^2 \log n)$ (使用 map)

[CF 687B] Remainders Game

Alice 和 Bob 协定一个数 k，以及 n 个数 $\{c_1, c_2, ..., c_n\}$
然后 Alice 想一个数 x，并把 $x \ mod \ c_i (i = 1..n)$ 告诉 Bob
问是否对任意的 x，Bob 都能猜出 $x \ mod \ k$
$1 \leq n, k, c_i \leq 10^6$

[CF 687B] Remainders Game

Alice 和 Bob 协定一个数 k，以及 n 个数 $\{c_1, c_2, ..., c_n\}$
然后 Alice 想一个数 x，并把 x mod $c_i(i = 1..n)$ 告诉 Bob
问是否对任意的 x，Bob 都能猜出 x mod k
$1 \leq n, k, c_i \leq 10^6$

分析：容易猜出，答案为“是”当且仅当 $k|LCM\{c_1, c_2, ..., c_n\}$
将 k 质因数分解，对每个 p^a 判定是否存在 $p^a|c_i$

威尔逊定理

$$(p-1)! \equiv -1 (mod\ p)$$

威尔逊定理

$(p-1)! \equiv -1(mod\ p)$

证明：$2 \sim p-2$ 逆元两两对应
又 $(p-1) \equiv -1(mod\ p)$

约数个数

$$\lim_{n\to\infty} sup\frac{\log d(n)}{\log n/\log\log n} = \log 2$$

$$sup\ d(n) = O(n^{\frac{\log 2}{\log\log n}})$$