### 数论基础

dcac



あ安電子科技大学 程序设计竞赛实训基地 Programming Contest Training Base

## 数论基础 ACM 中的数学小知识

陈德创

Codeforces: dcac

July 10, 2021

符号约定 整除与同余

整除 同余 快速幂

素数素数

筛法

欧几里得算 法

因数 最大公约数 最小公倍数 拓展欧几里得 算法

结束

整除与同余 整除 同金

快速幂

素数

素数 筛法 欧几里得算

因数 最大公约数

最小公倍数 拓展欧几里得 结束

概述

1 概述 ■简介

简介 ■ 符号约定 符号约定

■整除

■同余

■ 快速幂

■素数

筛法

■ 因数

■最大公约数

■最小公倍数

■ 拓展欧几里得算法

### 5 结束

概述

简介

符号约定

整除与同余 整除 同余

快速幂

素数

素数筛法

欧几里得算

法

因数 最大公约数 最小公倍数 拓展欧几里得 算法

结束

## 主要讲什么?

■ ACM 中的基础数学小知识

快速幂 素数 素数

筛法 欧几里得算

因数 最大公约数 最小公倍数

最小公倍数 拓展欧几里得 算法 结束

## 主要讲什么?

■ ACM 中的基础数学小知识 有啥用?

- 单独的题目考察
- 一些题目解答的必备前提
- 玩

麦数 泰数 筛法

欧几里得笪

因数 最大公约数 最小公倍数 拓展欧几里得

结束

1. x|y: x 整除 y, 即 x 是 y 的余数

2. x mod v: x 除以 v 的得到的余数

3. x ≡ y (mod M): x 同余 y (在模 M 的意义下)

4. gcd(x,y): x 和 y 的最大公约数, 简写作 (x,y)

5. lcm(x,y): x 和 y 的最小公倍数, 简写作 [x,y]

6. |x|: 下取整

7. [x]: 上取整

8. 
$$\Sigma$$
: 求和,如  $\sum\limits_{i=1}^n f(i)$ ,  $\sum\limits_{1 \leq i \leq n, i \in P} f(i)$ 

### 整除与同余

整除 同金 快速幂

素数

素数 筛法

## 欧几里得算

因数 最大公约数

最小公倍数 拓展欧几里得

结束

### Ⅱ 概述

- ■简介
- 符号约定

### 2 整除与同余

- ■整除
- ■同余
- 快速幂

- 素数
- 筛法

- 因数
- ■最大公约数
- ■最小公倍数
- 拓展欧几里得算法

### 5 结束

### 数论基础

dcac

概述 简介

符号约定

## 整除与同余

整除

快速幂

素数

筛法

欧几里得算 法

因数 最大公约数 最小公倍数 拓展欧几里得

结束

# 整除基本性质

整除的定义: 若整数 a 除以非零整数 b,商为整数且余数为零,即 a 能被 b 整除,记做 b|a,读作: b 整除 a 或 a 能被 b 整除。a 叫做 b 的倍数,b 叫做 a 的约数,或称为因数。

# 整除基本性质

整除的定义: 若整数 a 除以非零整数 b,商为整数且余数为零,即 a 能被 b 整除,记做 b|a,读作: b 整除 a 或 a 能被 b 整除。a 叫做 b 的倍数,b 叫做 a 的约数,或称为因数。

- 若 a|c,b|c,则 a|(b±c)
- 者 a|b, 则对任意的 c(c≠0), 有 a|bc
- 若 a|b, 且 b|c, 则 a|c
- 若 a|bc, 且 (a,c) = 1, 则 a|b
- 若 c|a, 且 c|b, 则对于任意整数 m,n, 有 c|(ma+nb)
- 若 a|c,b|c, 且 (a,b) = 1, 则 ab|c
- 带余除法定理:  $\forall a,b>0,a,b\in\mathbb{N}$ ,  $\exists$  唯一的数对 q,r, 使  $a=bq+r,(0\leq r< b)$ 。

结束

- 若  $a \equiv b \pmod{m}, c \equiv d \pmod{m}$ , 则  $a \pm b \equiv c \pm d \pmod{m}$
- 若  $a \equiv b \pmod{m}, c \equiv d \pmod{m}$ , 则  $a \times b \equiv c \times d \pmod{m}$

- 若  $a \equiv b \pmod{m}, c \equiv d \pmod{m},$ 则  $a \pm b \equiv c \pm d \pmod{m}$
- 若  $a \equiv b \pmod{m}, c \equiv d \pmod{m},$ 则  $a \times b \equiv c \times d \pmod{m}$
- 证明基本思路:  $a = k_1 m + b, c = k_2 m + b$

- 若  $a \equiv b \pmod{m}, c \equiv d \pmod{m},$ 则  $a \pm b \equiv c \pm d \pmod{m}$
- 若  $a \equiv b \pmod{m}, c \equiv d \pmod{m}$ , 则  $a \times b \equiv c \times d \pmod{m}$
- 证明基本思路:  $a = k_1 m + b, c = k_2 m + b$
- 基本用法: 求模运算下求解问题 (避免大数运算)、快速幂

结束

- 若  $a \equiv b \pmod{m}, c \equiv d \pmod{m},$ 则  $a \pm b \equiv c \pm d \pmod{m}$
- 若  $a \equiv b \pmod{m}, c \equiv d \pmod{m},$ 则  $a \times b \equiv c \times d \pmod{m}$
- 证明基本思路:  $a = k_1 m + b, c = k_2 m + b$
- 基本用法: 求模运算下求解问题 (避免大数运算)、快速幂
- 注意:除法不满足以上的性质

概述 简介

简介 符号约定

整除与同余 整除 同余

快速幂素数

素数筛法

欧几里得算

因数 最大公约数 最小公倍数 拓展欧几里得 算法

结束

求解:  $a^b \equiv x \pmod{m}, (a, b \le 1e9)$ 

整除与同余

欧几里得算

拓展欧几里得 结束

## 快速幂

```
求解: a^b \equiv x \pmod{m}, (a, b \le 1e9)
```

### 朴素求法:

```
inline 11 mpow(11 a, 11 b, 11 m) {
     11 \text{ res} = 1;
     for (int i = 1; i <= b; ++i)</pre>
       res = res * a % m;
4
5
     return res;
```

■ 时间复杂度: O(b)

最大公约数 最小公倍数 拓展欧几里得 算法 结束

快速幂

```
求解: a^b \equiv x \pmod{m}, (a, b \le 1e9)
```

### ■ 快速幂

```
inline 11 mpow(11 a, 11 b, 11 m) {
       11 \text{ res} = 1;
       while(b) {
         if (b & 1) res = res * a % m;
         a = a * a % m;
5
6
         b >>= 1;
       return res;
9
```

■ 时间复杂度: O(log b)

### 1 概述

- ■简介
- 符号约定

- ■整除
- ■同余
- 快速幂

### 3 素数

- 素数
- 筛法

- 因数
- ■最大公约数
- ■最小公倍数
- 拓展欧几里得算法

### 5 结束

表数

泰数 筛法

欧几里得笪

因数 最大公约数 最小公倍数 拓展欧几里得 算法

结束

 $\mathbf{c}$ 义:只能被 1 和自身整除的数称为素数,又称为质数

- 1 既不是素数,也不是合数
- 2 是最小的素数,也是唯一的偶素数

法田歌

■ 1 既不是素数,也不是合数■ 2 是最小的素数,也是唯一的偶素数

 $\mathbf{c}$ 义:只能被 1 和自身整除的数称为素数,又称为质数

### 基本性质:

■ 素数计数函数  $\pi(x)$ :  $\pi(x) \sim \frac{x}{\ln x}$ 

整除与同余整除

快速幂 素数

筛法 欧几里得算

因数最大公约数量小公倍数

最小公倍数 拓展欧几里得 算法

结束

定义: 只能被 1 和自身整除的数称为素数, 又称为质数

- 1 既不是素数,也不是合数
- 2 是最小的素数,也是唯一的偶素数

### 基本性质:

- 素数计数函数  $\pi(x)$ :  $\pi(x) \sim \frac{x}{\ln x}$
- 素数的分布: 1e9 范围内,任意两个相邻的素数差不超过 400

结束

定义: 只能被 1 和自身整除的数称为素数, 又称为质数

- 1 既不是素数,也不是合数
- 2 是最小的素数,也是唯一的偶素数

### 基本性质:

- 素数计数函数  $\pi(x)$ :  $\pi(x) \sim \frac{x}{\ln x}$
- 素数的分布: 1e9 范围内,任意两个相邻的素数差不超过 400
- 威尔逊定理:  $(p-1)! \equiv -1 \pmod{p}$

```
概述
简介
符号约定
```

整除与同余 整除 同金

快速幂 麦数

泰数 筛法

欧几里得笪 因数 最大公约数 最小公倍数

拓展欧几里得 结束

```
■ 暴力枚举所有可能的因子即可
```

- 事实: 如果 x|a, 那么 <sup>a</sup>/<sub>x</sub>|a。
- 判定算法:

```
inline bool isPrime(ll n) {
     if (n < 2) return false;
     for (int i = 2; i \leftarrow sqrt(n) + 1; ++i) {
       if (!(n % i)) return false;
6
     return true;
```

■ 时间复杂度: O(√n)

```
概述
简介
符号约定
```

整除与同余 整除

同余 快速幂 <del>素</del>数

素数

欧几里得算法

法 因数 最大公约数 最小公倍数 拓展欧几里得 算法

结束

■ 暴力枚举所有可能的因子即可

事实: 如果 x|a, 那么 a/x |a。

判定算法:

```
inline bool isPrime(11 n) {
    if (n < 2) return false;
    for (int i = 2; i <= sqrt(n) + 1; ++i) {
        if (!(n % i)) return false;
    }
    return true;
}</pre>
```

■ 时间复杂度: O(√n)

■ Miller-Rabin 素性测试: O(klog<sup>3</sup> n)

我不会

概述 简介 符号约定

符号约定整除与同余

整除 同余 快速幂

素数素数

筛法

欧几里得算 法

因数 最大公约数 最小公倍数 拓展欧几里得 算法

结束

筛法用于求出  $1 \sim n$  中所有的素数。

■ 事实: 一个数 x 的任意整数倍一定是素数

概述 简介 符号约定

整除与同余整除

快速幂 素数 素数

筛法 欧几里得算

因数 最大公约数 最小公倍数 拓展欧几里得

结束

筛法用于求出  $1 \sim n$  中所有的素数。

- 事实: 一个数 x 的任意整数倍一定是素数
- 对每个数,将其倍速标记为**非素数**,那么没被标记的数,就是素数

整除 同余 快速幂

表数 泰数 筛法

欧几里得复 因数 最大公约数 最小公倍数

拓展欧几里得 结束

備法用干求出  $1 \sim n$  中所有的素数。

- 事实: 一个数 x 的任意整数倍一定是素数
- 对每个数,将其倍速标记为非素数,那么没被标记的数,就是素数
- 算法实现:

```
const int MAXN = 1e6 + 5;
    bool inp[MAXN], prime[MAXN], cnt;
 3
    inline void getPrime(int n) {
        memset(inp, 0, sizeof(inp));
        for (int i = 2; i <= n; ++i) {
 6
           if (!inp[i]) prime[++cnt] = i;
           for (int j = 2; i * j <= n; ++j)
 8
           inp[i * i] = true;
 9
10
```

■ 时间复杂度: O(nlog n)

### 数论基础

dcac

概述 简介

简介 符号约定

整除与同余 整除 同余

快速幂 素数 素数

筛法 欧几.里得复

因数 最大公约数 最小公倍数 拓展欧几里得 算法

结束

# 埃拉托斯特尼筛法

■ 只需要对素数的倍数进行标记就好了

素数筛法

欧几里得算法

法 因数 最大公约数 最小公倍数 拓展欧几里得 算法

结束

## 埃拉托斯特尼筛法

- 只需要对素数的倍数进行标记就好了
- 算法实现:

```
inline void getPrime(int n) {
   memset(inp, 0, sizeof(inp));
   for (int i = 2; i <= n; ++i) {
      if (inp[i]) continue;
      prime[++cnt] = i;
      for (int j = 2; i * j <= n; ++j)
            inp[i * j] = true;
   }
}</pre>
```

■ 时间复杂度: O(n log log n)

### 数论基础

dcac

概述 简介

符号约定

整除与同余 整除

同余 快速器

素数 泰数

筛法

欧几里得算

因数 最大公约数 最小公倍数 拓展欧几里得 算法

结束

## 线性筛 欧拉筛

■ 如果能让每个合数都只被标记一次就好了

结束

## 线性筛 欧拉筛

- 如果能让每个合数都只被标记一次就好了
- 算法实现:

```
inline void getPrime(int n) {
       for (int i = 2; i <= n; ++i) {
3
          if (!inp[i]) prime[++cnt] = i;
4
          for (int j = 1; j <= cnt && i * prime[j] <= n; ++j) {</pre>
5
              inp[i * prime[j]] = true;
              if (!(i % prime[j])) break;
6
7
8
9
```

麦数 泰数 筛法

欧几里得复

因数 最大公约数 最小公倍数 拓展欧几里得 質法

结束

## 线性筛 欧拉筛

- 如果能让每个合数都只被标记一次就好了
- 算法实现:

```
inline void getPrime(int n) {
       for (int i = 2; i <= n; ++i) {
3
          if (!inp[i]) prime[++cnt] = i;
4
          for (int j = 1; j <= cnt && i * prime[j] <= n; ++j) {</pre>
5
              inp[i * prime[j]] = true;
              if (!(i % prime[i])) break;
6
7
8
q
```

- 时间复杂度: O(n)
- 非常重要,可以用来线性求解某些积性函数

# 概述

简介

符号约定 整除与同余 整除 同金

快速幂

素数

泰数 筛法 欧几里得算 法

因数 最大公约数

结束

最小公倍数 拓展欧几里得

■简介

Ⅱ 概述

■ 符号约定

■整除

■同余

■ 快速幂

■ 素数

筛法

### 4 欧几里得算法

■因数

■ 最大公约数

■最小公倍数

■ 拓展欧几里得算法

5 结束

### 数论基础

dcac

因数

简介 符号约定

整除与同余

整除

概述

同余 快速幂

素数

系数素数

筛法

### 欧几里得算 法

### 因数

最大公约数

最小公倍数

拓展欧几里得 算法

结束

概述 简介 符号约定

符号约定整除与同余

整除同余快速器

素数素数

筛法 欧几里得算

欧几里得! 法

因数

最大公约数 最小公倍数 拓展欧几里得 管法

结束

## 定理 (算术基本定理)

 $\forall A \in \mathbb{N}, A > 1$   $\exists p_1 < p_2 < p_3 < \dots < p_n, a_i \in \mathbb{Z}^+ \text{ s.t. } \prod_{i=1}^n p_i^{a_i} = A$  其中  $p_i$  是一个质数。这种表示的方法存在,而且是唯一的。

概述 简介 符号约定

整除与同余

整除 同金 快速幂

表数 泰数 筛法

欧几里得复

最大公约数

最小公倍数 拓展欧几里得

结束

## 定理 (算术基本定理)

 $\forall A \in \mathbb{N}, A > 1 \quad \exists p_1 < p_2 < p_3 < \dots < p_n, a_i \in \mathbb{Z}^+ \text{ s.t. } \prod_{i=1}^n p_i^{a_i} = A$ 其中 pi 是一个质数。这种表示的方法存在,而且是唯一的。

■ 1 ~ 1e9 的数最多有 1344 个因子。(by 欢神)

最大公约数

## 定理 (算术基本定理)

 $\forall A \in \mathbb{N}, A > 1 \quad \exists p_1 < p_2 < p_3 < \dots < p_n, a_i \in \mathbb{Z}^+ \text{ s.t. } \prod_{i=1}^n p_i^{a_i} = A$ 其中 pi 是一个质数。这种表示的方法存在, 而且是唯一的。

- 1 ~ 1e9 的数最多有 1344 个因子。(by 欢神)
- 求解一个数的所有因数:

```
vector<int> breakdown(int n) {
      vector<int> res:
      for (int i = 2; i * i <= n; i++) {
       if (n % i == 0) {
         res.push back(i);
 6
         res.push back(n / i);
 9
      return res;
10
```

■ 时间复杂度: O(√n)

概述 简介 符号约定

整除与同余整除

快速幂 素数 素数 筛法

欧几里得算

因数

最大公约数 最小公倍数 拓展欧几里得 算法

结束

■ 求解一个数的所有质因数 (即质因数分解):

概述 简介

结束

```
拓展欧几里得
```

## ■ 求解一个数的所有质因数(即质因数分解):

```
vector<int> breakdown(int n) {
 1
      vector<int> res;
      for (int i = 2; i * i <= n; i++) {
        if (n % i == 0) {
         while (n % i == 0) n /= i;
 6
          res.push back(i);
 7
 8
 9
      if (n != 1) res.push back(n);
10
      return res;
11
```

■ 时间复杂度: O(√n/lnn)

### 数论基础

dcac

最大公约数

GCD

概述 简介

符号约定 整除与同余

整除同余

快速幂素数

素数筛法

欧几里得算

因数

最大公约数

最小公倍数 拓展欧几里得 質法

结束

最大公约数,是指一组数所有公约数里面最大的一个。

结束

## 最大公约数 GCD

最大公约数,是指一组数所有公约数里面最大的一个。

■ a 和 b 的最大公约数记为: gcd(a,b)

## 最大公约数 GCD

最大公约数,是指一组数所有公约数里面最大的一个。

- a 和 b 的最大公约数记为: gcd(a,b)
- 事实:

$$\mathsf{gcd}(a,b) = \mathsf{gcd}(b,a-b)$$

$$\mathsf{gcd}(a,b) = \mathsf{gcd}(a,a \ \mathsf{mod}\ b)$$

GCD

简介 符号约定

整除与同余整除

快速幂 素数 素数

筛法 欧几里得算

法因数

最大公约数

最小公倍数

拓展欧几里 算法

结束

最大公约数,是指一组数所有公约数里面最大的一个。

- a 和 b 的最大公约数记为: gcd(a,b)
- 事实:

$$gcd(a,b) = gcd(b,a-b)$$
  
 $gcd(a,b) = gcd(a,a \text{ mod } b)$ 

■ 集合  $S = \{ gcd(a_i, ..., a_n) | 1 \le i \le n \}$  至多有  $O(max(a_i))$  个元素。

泰数 筛法

欧几里得笪

因数

最大公约数

最小公倍数 拓展欧几里得 算法

结束

## 欧几里得算法

### 算法实现:

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

■ 时间复杂度: O(log n)

筛法 欧几里得算

因数

最大公约数最小公倍数

拓展欧几里得算法

结束

## 欧几里得算法

## 算法实现:

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

- 时间复杂度: O(log n)
- 求解**斐波那契数列**相邻两项时达到最坏时间复杂度

deac

概述 简介

符号约定 整除与同余

整除

快速幂 素数 素数

筛法 欧几.里得

欧几里得算 法

因数 最大公约数

最小公倍数

拓展欧几里? 算法

结束

## 欧几里得算法

## 算法实现:

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

- 时间复杂度: O(log n)
- 求解斐波那契数列相邻两项时达到最坏时间复杂度
- <algorithm>库内置了\_\_gcd()函数

## 欧几里得算法

## 算法实现:

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

- 时间复杂度: O(log n)
- 求解斐波那契数列相邻两项时达到最坏时间复杂度
- <algorithm>库内置了\_\_gcd()函数
- 本质上是将将大规模问题转化为小规模问题
- 将问题 (a,b) 转化为了 (b,a mod b)

## 多个数的最大公约数

### 求解多个数的最大公约数:

- 可以证明, gcd(a,gcd(b,c)) = gcd(a,b,c)
- 即每次取出两个数求出 gcd 后再放 gcd 回去,不会对所需要的答案造成 影响。

素数素数

欧几里得算法

因数 最大公约数

结束

最小公倍数 拓展欧几里得 算法

## 多个数的最大公约数

### 求解多个数的最大公约数:

- 可以证明, gcd(a,gcd(b,c)) = gcd(a,b,c)
- 即每次取出两个数求出 gcd 后再放 gcd 回去,不会对所需要的答案造成 影响。

```
int gcd(int a[], int n) {
   int res = a[1];
   for (int i = 2; i <= n; ++i)
    res = __gcd(res, a[i]);
   return res;
}</pre>
```

结束

# 例题

## 又是毕业季

题目链接: 又是毕业季 I

## 题目描述

给定 n 和 k, 问在  $1 \sim n$  中选 k 个数使得它们 gcd 最大, 求最大值是多少。 1 < k < n < 1e9

# 例题

## 又是毕业季

题目链接: 又是毕业季 I

## 题目描述

给定 n 和 k, 问在  $1\sim n$  中选 k 个数使得它们 gcd 最大, 求最大值是多少。 1 < k < n < 1e9

## 题解

■ 反向考虑,即考虑一个数 x, 1~n 中有多少数被它整除。

# 例题

## 又是毕业季

题目链接: 又是毕业季 I

## 题目描述

给定 n 和 k,问在  $1\sim n$  中选 k 个数使得它们 gcd 最大,求最大值是多少。  $1\leq k\leq n\leq 1e9$ 

- 反向考虑,即考虑一个数 x,1~n 中有多少数被它整除。
- 显然为 [<sup>n</sup>/<sub>x</sub>] 个
- 即求  $\max_{x}\{[\lfloor \frac{n}{x} \rfloor \geq k]\}$ ,显然答案为  $\lfloor \frac{n}{k} \rfloor$

概述

简介 符号约定

整除与同余整除

快速幂 素数 素数 筛法

欧几里得算 法

因数 最大公约数 最小公倍数 拓展欧几里得 算法

结束

题目链接: 又是毕业季 I

## 题目描述

给定 n 和 k,问在  $1\sim n$  中选 k 个数使得它们 gcd 最大,求最大值是多少。  $1\leq k\leq n\leq 1e9$ 

- 反向考虑,即考虑一个数 x, 1~n 中有多少数被它整除。
- 显然为 [n/x] 个
- 即求  $\max_x \{[\lfloor \frac{n}{x} \rfloor \ge k]\}$ , 显然答案为  $\lfloor \frac{n}{k} \rfloor$
- 时间复杂度: O(1)

概述

简介 符号约定

符号约定 整除与同余

整除 同余 快速幂

素数素数筛法

欧几里得算 法

因数 最大公约数 最小公倍数 拓展欧几里得

结束

题目链接: 又是毕业季 [

## 题目描述

给定 n 和 k,问在  $1\sim n$  中选 k 个数使得它们 gcd 最大,求最大值是多少。  $1\leq k\leq n\leq 1e9$ 

- 反向考虑,即考虑一个数 x,1~n 中有多少数被它整除。
- 显然为 [<sup>n</sup>/<sub>x</sub>] 个
- 即求  $\max\limits_{\mathbf{x}}\{[\lfloor \frac{\mathbf{n}}{\mathbf{x}} \rfloor \geq \mathbf{k}]\}$ ,显然答案为 $\lfloor \frac{\mathbf{n}}{\mathbf{k}} \rfloor$
- 时间复杂度: O(1)

```
n, k = map(int, input().split())
print(int(n / k))
```

概述 简介

符号约定 整除与同余

整除 同金 快速幂

麦数 泰数 筛法

欧几里得笪 因数

最大公约数 最小公倍数 拓展欧几里得

结束

题目链接: 又是毕业季 I

## 题目描述

给定 n 和 k, 问在  $1 \sim n$  中选 k 个数使得它们 gcd 最大, 求最大值是多少。 1 < k < n < 1e9

## 颞解

- 反向考虑、即考虑一个数 x, 1~n 中有多少数被它整除。
- 显然为 | ♣ | 个
- 即求  $\max_{\mathbf{x}}\{[\lfloor \frac{\mathbf{n}}{\mathbf{x}} \rfloor \geq \mathbf{k}]\}$ , 显然答案为  $\lfloor \frac{\mathbf{n}}{\mathbf{k}} \rfloor$
- 时间复杂度: O(1)

n, k = map(int, input().split()) print(int(n / k))

同样思想解决的题: 又是毕业季 II (双倍经验它不香吗?)。

整除与同余 整除

概述

简介 符号约定

同金 快速幂

麦数

泰数

## 例题

## 最大公约数

题目链接: 最大公约数

## 题目描述

有一个  $n \times m$  的矩阵 a。对此矩阵进行变换,定义为将这个矩阵内的所有元 素变为其上下左右四个元素(不存在则忽略)及自身的最大公约数。 询问  $a_{x,y}$  在进行最少多少次变换之后会变成 1。如果可以使  $a_{x,y}$  经过若干次 变换变成 1, 输出其中最小的次数; 否则输出 -1。

筛法 欧几里得笪

因数 最大公约数

最小公倍数 拓展欧几里得

结束

# 例颢

## 最大公约数

题目链接: 最大公约数

## 题目描述

有一个  $n \times m$  的矩阵 a。对此矩阵进行变换,定义为将这个矩阵内的所有元 素变为其上下左右四个元素(不存在则忽略)及自身的最大公约数。 询问  $a_{x,v}$  在进行最少多少次变换之后会变成 1。如果可以使  $a_{x,v}$  经过若干次 变换变成 1, 输出其中最小的次数; 否则输出 -1。

## 数据范围:

$$1 \leq n,m \leq 2 \times 10^3, 1 \leq a_{i,j} \leq 10^{18}, 1 \leq x \leq n, 1 \leq y \leq m$$

题目链接:最大公约数

## 题目描述

有一个  $n \times m$  的矩阵 a。对此矩阵进行变换,定义为将这个矩阵内的所有元 素变为其上下左右四个元素(不存在则忽略)及自身的最大公约数。 询问  $a_{x,v}$  在进行最少多少次变换之后会变成 1。如果可以使  $a_{x,v}$  经过若干次 变换变成 1, 输出其中最小的次数; 否则输出 -1。

数据范围:

$$1 \leq n,m \leq 2 \times 10^3, 1 \leq a_{i,j} \leq 10^{18}, 1 \leq x \leq n, 1 \leq y \leq m$$

### hint

■ 事实:

$$\begin{aligned} & \mathsf{gcd}(a,\mathsf{gcd}(b,c)) & &= \mathsf{gcd}(a,b,c) \\ & \mathsf{gcd}(\mathsf{gcd}(a,b),\mathsf{gcd}(b,c)) & &= \mathsf{gcd}(a,b,c) \end{aligned}$$

■ 分别考虑"判断可能"和"求解"。

### 数论基础

dcac

概述 简介

符号约定

整除与同余

整除 同余

快速器 素数

泰数 筛法

欧几里得算

因数

最大公约数

最小公倍数 拓展欧几里得

结束

## 例题 最大公约数

## 题解

显然, 当且仅当所有元素的 gcd 不为 1 时, 答案不存在。

### 数论基础

dcac

概述 篇介

符号约定

整除与同余 整除

同余 快速器

表数 泰数

筛法

欧几里得笪

因数

最大公约数

最小公倍数 拓展欧几里得

结束

## 例题 最大公约数

## 题解

## 例颢 最大公约数

### 题解

- 第一次, a<sub>x,v</sub> 由 a<sub>x,v</sub> 上下左右四个点来更新。
- 记 a<sub>x,v</sub> 周围的四个点为 b<sub>1</sub>,b<sub>2</sub>,b<sub>3</sub>,b<sub>4</sub>。 同时 b; 也由它们的上下左右四个点来更新

## 例颢 最大公约数

### 颞解

- 第一次, a<sub>x,v</sub> 由 a<sub>x,v</sub> 上下左右四个点来更新。
- 记 a<sub>x,v</sub> 周围的四个点为 b<sub>1</sub>,b<sub>2</sub>,b<sub>3</sub>,b<sub>4</sub>。 同时 b; 也由它们的上下左右四个点来更新
- 第二次,相当于 ax,y 由每个 bi 周围的四个点更新。 即相当于 ax.v 由距离它小于 2 的点更新。
- 同时 b; 也是由距离它小于 2 的点更新。

概述

### 简介 符号约定 整除与同余

整除 同余 快速幂

素数素数筛法

欧几里得算 法

最大公约数 最小公倍数 拓展欧几里得 算法

拓展欧几里? 算法 结束

## 题解

- 第一次、 $a_{x,y}$  由  $a_{x,y}$  上下左右四个点来更新。 即  $a_{x,y} = \text{gcd}(a_{x-1,y}, a_{x+1,y}, a_{x,y-1}, a_{x,y+2})$ 。
- 记 ax,y 周围的四个点为 b1,b2,b3,b4。
   同时 bi 也由它们的上下左右四个点来更新
- 第二次,相当于 ax,y 由每个 bi 周围的四个点更新。
   即相当于 ax,y 由距离它小于 2 的点更新。
- 同时 b<sub>i</sub> 也是由距离它小于 2 的点更新。
- 以此类推,假设第 k 次, $a_{x,y}$  由距离它小于 k 的点更新。 那么  $b_i$  也是由距离它小于 k 的点更新。

## 例颢 最大公约数

## 颞解

- 第一次, a<sub>x,v</sub> 由 a<sub>x,v</sub> 上下左右四个点来更新。
- 记 a<sub>x.v</sub> 周围的四个点为 b<sub>1</sub>,b<sub>2</sub>,b<sub>3</sub>,b<sub>4</sub>。 同时 b; 也由它们的上下左右四个点来更新
- 第二次,相当于 a<sub>x,v</sub> 由每个 b<sub>i</sub> 周围的四个点更新。 即相当于 ax.v 由距离它小于 2 的点更新。
- 同时 b; 也是由距离它小于 2 的点更新。
- 以此类推,假设第 k 次,a<sub>x,v</sub> 由距离它小于 k 的点更新。 那么 b: 也是由距离它小干 k 的点更新。
- 那么显然第 k+1 次, a<sub>x,v</sub> 由距离 b<sub>i</sub> 小于 k 的点更新。 即相当于距离它小于 k+1 的点更新。

麦数 泰数 筛法

欧几里得笪

因数 最大公约数 最小公倍数

拓展欧几里得

结束

## 例题 最大公约数

- 即我们归纳证明了,第 k 次,a<sub>x,y</sub> 由距离它小于 k 的点更新。
- 即,第 k 次变换后,a<sub>x,y</sub> 的值变成了所有距它小于等于 k 的点的值的 gcd.

筛法 欧几里得笪

因数

最大公约数 最小公倍数 拓展欧几里得

结束

## 例题 最大公约数

- 即我们归纳证明了,第 k 次,a<sub>x,v</sub> 由距离它小于 k 的点更新。
- 即,第 k 次变换后,a<sub>x,y</sub> 的值变成了所有距它小于等于 k 的点的值的 gcd.
- 题目转化为: 从 (x,y) 开始一层层取 gcd, 直到 gcd = 1。

素数素数筛法

欧几里得算 法

因数 最大公约数 最小公倍数 拓展欧几里得 管注

结束

## **例题** 最大公约数

- 即我们归纳证明了,第 k 次,a<sub>x,y</sub> 由距离它小于 k 的点更新。
- $\blacksquare$  即,第 k 次变换后, $a_{x,y}$  的值变成了所有距它小于等于 k 的点的值的 gcd  $_{\circ}$
- 题目转化为: 从 (x,y) 开始一层层取 gcd, 直到 gcd = 1。
- 从 (x,y) 开始 bfs,同时对每个遍历到的元素取 gcd,直到 gcd = 1。
- 遍历层数就是答案。

结束

# 最小公倍数

LCM

事实:

$$\forall a,b>0, a,b \in \mathbb{N}, a \times b = \text{gcd}(a,b) \times lcm(a,b)$$

■ 所以求出俩数的 GCD 后 LCM 可在 O(1) 的时间内求出

概述

简介 符号约定 整除与同余 整除

麦数

泰数 筛法

因数

## 最小公倍数

LCM

同金 快速幂

$$\forall a,b>0, a,b \in \mathbb{N}, a \times b = \text{gcd}(a,b) \times lcm(a,b)$$

- 所以求出俩数的 GCD 后 LCM 可在 O(1) 的时间内求出
- 算法实现:

```
1
   int lcm(int a, int b) {
     return a * b / __gcd(a, b);
```

事实:

```
最大公约数
最小公倍数
拓展欧几里得
算法
结束
```

欧几里得笪

# 最小公倍数

LCM

### 事实:

$$\forall a,b>0, a,b \in \mathbb{N}, a \times b = \text{gcd}\big(a,b\big) \times lcm\big(a,b\big)$$

- 所以求出俩数的 GCD 后 LCM 可在 O(1) 的时间内求出
- 算法实现:

```
int lcm(int a, int b) {
 return a * b / __gcd(a, b);
```

- 多个数的 LCM
  - 与多个数的 GCD 类似

## 最小公倍数

LCM

## ■ 事实:

```
\forall a,b>0, a,b \in \mathbb{N}, a \times b = \text{gcd}\big(a,b\big) \times lcm\big(a,b\big)
```

- 所以求出俩数的 GCD 后 LCM 可在 O(1) 的时间内求出
- 算法实现:

```
1  int lcm(int a, int b) {
2   return a * b / _gcd(a, b);
}
```

### ■ 多个数的 LCM

- 与多个数的 GCD 类似
- 每次取出两个数求 LCM 后再将 LCM 放回去,不会对所需要的答案造成影响

概述

简介 符号约定 整除与同余

整除 同金

麦数

泰数

题目链接: [NOIP2001 普及组] 最大公约数和最小公倍数问题

## 题目描述

输入两个正整数  $x_0, y_0, (2 \le x_0, y_0 \le 10^5)$ , 求出满足下列条件的 P, Q 的个 数:

- P, Q 是正整数。
- 要求 P, Q 以 x<sub>0</sub> 为最大公约数,以 y<sub>0</sub> 为最小公倍数。

试求: 满足条件的所有可能的 P, Q 的个数。

筛法 欧几里得笪

因数 最大公约数

最小公倍数 拓展欧几里得

结束

## 最大公约数和最小公倍数问题

题目链接: [NOIP2001 普及组] 最大公约数和最小公倍数问题

## 题目描述

输入两个正整数  $x_0, y_0, (2 < x_0, y_0 < 10^5)$ ,求出满足下列条件的 P, Q 的个 数:

- P, Q 是正整数。
- 要求 P, Q 以 x<sub>0</sub> 为最大公约数,以 y<sub>0</sub> 为最小公倍数。

试求: 满足条件的所有可能的 P, Q 的个数。

- 根据  $P \times Q = gcd(P,Q) \times lcm(P,Q) = x_0 y_0$ , 我们可以得到  $P \times Q$  的值
- 枚举  $i|(x_0y_0)$ ,考虑  $i, |\frac{x_0y_0}{i}|$  的最大公约数和最小公倍数是不是分别为 P,Q 即可

概述

简介 符号约定

整除 同金

快速幂 麦数

泰数

筛法 欧几里得复

因数

最大公约数

拓展欧几里得 结束

题目链接: [NOIP2001 普及组] 最大公约数和最小公倍数问题

## 题目描述

输入两个正整数  $x_0, y_0, (2 < x_0, y_0 < 10^5)$ ,求出满足下列条件的 P, Q 的个 数:

- P, Q 是正整数。
- 要求 P, Q 以 x<sub>0</sub> 为最大公约数,以 y<sub>0</sub> 为最小公倍数。

试求: 满足条件的所有可能的 P, Q 的个数。

- 根据  $P \times Q = gcd(P,Q) \times lcm(P,Q) = x_0 y_0$ , 我们可以得到  $P \times Q$  的值
- 枚举  $i|(x_0y_0)$ ,考虑  $i, |\frac{x_0y_0}{i}|$  的最大公约数和最小公倍数是不是分别为 P,Q 即可
- 时间复杂度: O(n log n)

#### 数论基础

dcac

概述 简介

符号约定

整除与同余

同余

素数

素数筛法

VID 7ZS

欧几里得算 法

因数 最大公约数 最小公倍数

拓展欧几里得 算法

结束

# 拓展欧几里得算法 <sup>简介</sup>

■ 拓展欧几里得算法用于求解 ax + by = gcd(a,b) 的一组可行解

麦数 泰数

筛法

欧几里得笪

因数 最大公约数

最小公倍数

拓展欧几里得 算法

结束

# 拓展欧几里得算法 简介

- 拓展欧几里得算法用于求解 ax + by = gcd(a,b) 的一组可行解
- 常用于求解乘法逆元

欧几里得笪

法因数

最大公约数

最小公倍数 拓展**欧几里得** 

算法

结束

# 拓展欧几里得算法 <sup>简介</sup>

- 拓展欧几里得算法用于求解 ax + by = gcd(a,b) 的一组可行解
- 常用于求解**乘法逆元**
- 裴蜀定理:该方程组的解一定存在

#### 定理 (裴蜀定理)

设 a,b 是不全为零的整数,则存在整数 x,y,使得 ax + by = gcd(a,b)。

素数素数

筛法 欧几里得算

因数 最大公约数

最小公倍数

算法结束

结果

# 拓展欧几里得算法

- 拓展欧几里得算法用于求解 ax + by = gcd(a,b) 的一组可行解
- 常用于求解**乘法逆元**
- 裴蜀定理:该方程组的解一定存在

#### 定理 (裴蜀定理)

设 a,b 是不全为零的整数,则存在整数 x,y, 使得 ax + by = gcd(a,b)。

■ 算法思想: 将问题 (a,b) 转化为 (b,a mod b)

素数素数

筛法 欧几里得算

法

因数 最大公约数 最小公倍数

拓展欧几里得 算法

#/A

结束

# 拓展欧几里得算法

- 拓展欧几里得算法用于求解 ax + by = gcd(a,b) 的一组可行解
- 常用于求解**乘法逆元**
- 裴蜀定理:该方程组的解一定存在

# 定理 (裴蜀定理)

设 a,b 是不全为零的整数,则存在整数 x,y, 使得 ax + by = gcd(a,b)。

- 算法思想: 将问题 (a,b) 转化为 (b,a mod b)
- 要注意负值

#### 数论基础

dcac

概述 简介

符号约定

整除与同余

整除 同余 快速幂

素数 素数

筛法

欧几里得算

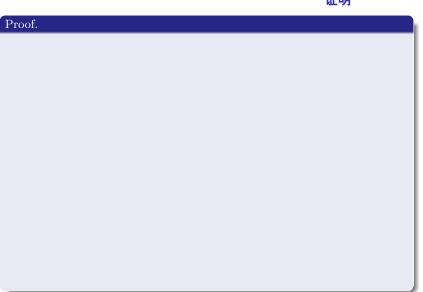
因数

最大公约数 最小公倍数

拓展欧几里得 算法

结束

# 拓展欧几里得算法 证明



$$ax_1 + by_1 = gcd(a,b)$$

$$bx_2 + (a \text{ mod } b)y_2 = \gcd(b, a \text{ mod } b)$$

$$ax_1 + by_1 = \gcd(a,b)$$

$$bx_2 + (a \text{ mod } b)y_2 = \text{gcd}(b, a \text{ mod } b)$$

$$\because \mathsf{gcd}(a,b) \quad = \! \mathsf{gcd}(b,a \; \mathsf{mod} \; b)$$

$$ax_1 + by_1 = gcd(a,b)$$
  
 $bx_2 + (a \text{ mod } b)y_2 = gcd(b, a \text{ mod } b)$ 

$$\because \mathsf{gcd}(a,b) = \mathsf{gcd}(b,a \mathsf{\ mod\ } b)$$

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

$$ax_1+by_1 \hspace{1cm} = \text{gcd}\big(a,b\big)$$

$$bx_2 + (a \text{ mod } b)y_2 = gcd(b, a \text{ mod } b)$$

$$\because \mathsf{gcd}(a,b) \quad = \! \mathsf{gcd}(b,a \; \mathsf{mod} \; b)$$

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

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

$$\begin{array}{ll} ax_1+by_1 & = \gcd(a,b) \\ bx_2+(a \text{ mod } b)y_2 & = \gcd(b,a \text{ mod } b) \\ \therefore \gcd(a,b) & = \gcd(b,a \text{ mod } b) \\ \therefore ax_1+by_1 & = bx_2+(a \text{ mod } b)y_2 \\ \therefore a \text{ mod } b & = a-(\lfloor\frac{a}{b}\rfloor\times b) \\ \therefore ax_1+by_1 & = bx_2+(a-(\lfloor\frac{a}{b}\rfloor\cdot b))y_2 \\ & = ay_2+bx_2-\lfloor\frac{a}{b}\rfloor\cdot by_2 \end{array}$$

$$\begin{array}{ll} ax_1+by_1 & = \gcd(a,b) \\ bx_2+(a \text{ mod } b)y_2 & = \gcd(b,a \text{ mod } b) \\ \therefore \gcd(a,b) & = \gcd(b,a \text{ mod } b) \\ \therefore ax_1+by_1 & = bx_2+(a \text{ mod } b)y_2 \\ \therefore a \text{ mod } b & = a-(\lfloor\frac{a}{b}\rfloor\times b) \\ \therefore ax_1+by_1 & = bx_2+(a-(\lfloor\frac{a}{b}\rfloor\cdot b))y_2 \\ & = ay_2+bx_2-\lfloor\frac{a}{b}\rfloor\cdot by_2 \\ \therefore ax_1+by_1 & = ay_2+b(x_2-\lfloor\frac{a}{b}\rfloor y_2) \end{array}$$

$$\begin{array}{ll} ax_1+by_1 & = \gcd(a,b) \\ bx_2+(a \ \mathsf{mod} \ b)y_2 & = \gcd(b,a \ \mathsf{mod} \ b) \\ & \because \gcd(a,b) & = \gcd(b,a \ \mathsf{mod} \ b) \\ & \therefore ax_1+by_1 & = bx_2+(a \ \mathsf{mod} \ b)y_2 \\ & \because a \ \mathsf{mod} \ b & = a-(\lfloor\frac{a}{b}\rfloor\times b) \\ & \therefore ax_1+by_1 & = bx_2+(a-(\lfloor\frac{a}{b}\rfloor\cdot b))y_2 \\ & = ay_2+bx_2-\lfloor\frac{a}{b}\rfloor\cdot by_2 \\ & \therefore ax_1+by_1 & = ay_2+b(x_2-\lfloor\frac{a}{b}\rfloor y_2) \\ & \therefore ax_1+by_1 & = ay_2+b(x_2-\lfloor\frac{a}{b}\rfloor y_2) \\ & \therefore a=a, & b=b \\ & \therefore x_1=y_2, & y_1=x_2-\lfloor\frac{a}{b}\rfloor y_2 \end{array}$$

筛法

欧几里得算

因数 最大公约数 最小公倍数

拓展欧几里得 算法

结束

# 拓展欧几里得算法 实现

- $\mathbf{x}_1 = \mathbf{y}_2, \mathbf{y}_1 = \mathbf{x}_2 \lfloor \frac{\mathbf{a}}{\mathbf{b}} \rfloor \mathbf{y}_2$
- 其中 x<sub>1</sub>,y<sub>1</sub> 是问题 (a,b) 的答案 x<sub>2</sub>,y<sub>2</sub> 是问题 (b,a mod b) 的答案

# 拓展欧几里得算法 实现

- $\mathbf{x}_1 = \mathbf{y}_2, \mathbf{y}_1 = \mathbf{x}_2 \left| \frac{\mathbf{a}}{\mathbf{b}} \right| \mathbf{y}_2$
- 其中 x<sub>1</sub>,y<sub>1</sub> 是问题 (a,b) 的答案 x<sub>2</sub>,y<sub>2</sub> 是问题 (b,a mod b) 的答案
- 算法实现:

```
ll exgcd(ll a, ll b, ll& x, ll& y){
        if (!b){
3
            x = 1, y = 0;
            return a;
        11 gcd = exgcd(b, a % b, y, x);
6
        y = a / b * x;
        return gcd;
```

# 例题 同余方程

题目链接: [NOIP2012 提高组] 同余方程

### 题目描述

求关于 x 的同余方程  $ax \equiv 1 \pmod{b}$  的最小正整数解。 数据保证一定有解。

2 < a, b < 2e9

# 例题 同余方程

题目链接: [NOIP2012 提高组] 同余方程

### 题目描述

求关于 x 的同余方程  $ax \equiv 1 \pmod{b}$  的最小正整数解。 数据保证一定有解。 2 < a, b < 2e9

#### 题解

 $\mathbf{a} \mathbf{x} \equiv 1 \pmod{\mathbf{b}}$  可以转化为  $\mathbf{a} \mathbf{x} = (-\mathbf{b})\mathbf{y} + 1$ ,即  $\mathbf{a} \mathbf{x} + \mathbf{b} \mathbf{y} = 1$ 。

题目链接: [NOIP2012 提高组] 同余方程

### 题目描述

求关于 x 的同余方程  $ax \equiv 1 \pmod{b}$  的最小正整数解。数据保证一定有解。 2 < a,b < 2e9

# 

- $\mathbf{a} \mathbf{x} \equiv 1 \pmod{\mathbf{b}}$  可以转化为  $\mathbf{a} \mathbf{x} = (-\mathbf{b})\mathbf{y} + 1$ ,即  $\mathbf{a} \mathbf{x} + \mathbf{b} \mathbf{y} = 1$ 。
- 拓展欧几里得裸题。

# 例题 同余方程

题目链接: [NOIP2012 提高组] 同余方程

## 题目描述

求关于 x 的同余方程  $ax \equiv 1 \pmod{b}$  的最小正整数解。 数据保证一定有解。

2 < a, b < 2e9

#### 题解

- $\mathbf{a} \mathbf{x} \equiv 1 \pmod{\mathbf{b}}$  可以转化为  $\mathbf{a} \mathbf{x} = (-\mathbf{b})\mathbf{y} + 1$ ,即  $\mathbf{a} \mathbf{x} + \mathbf{b} \mathbf{y} = 1$ 。
- 拓展欧几里得裸题。
- 时间复杂度: O(log a)

简介 符号约定 整除与同余

整除同余快速器

素数素数筛法

欧几里得算 法 因数

最大公约数最小公倍数

拓展欧几里得 算法

结束

题目链接: [NOIP2012 提高组] 同余方程

## 题目描述

求关于 x 的同余方程  $ax \equiv 1 \pmod{b}$  的最小正整数解。 数据保证一定有解。 2 < a,b < 2e9

#### 题解

- $ax \equiv 1 \pmod{b}$  可以转化为 ax = (-b)y + 1, 即 ax + by = 1.
- 拓展欧几里得裸题。
- 时间复杂度: O(log a)

#### 拓展欧几里得习题推荐:

■ 青蛙的约会

概述 简介

符号约定

同余

素数

素数 筛法 欧几里得算 因数

最大公约数

最小公倍数 拓展欧几里得 算法 结束

快速幂

#### 整除与同余 整除

1 概述 ■简介

■整除

■同余

■ 快速幂

■ 符号约定

■素数

筛法

■ 因数

■最大公约数

■最小公倍数

■ 拓展欧几里得算法

# 5 结束

#### 数论基础

dcac

概述

简介 符号约定

符号列定 整除与同余

整除

同余 快速幂

素数

素数筛法

欧几里得算

政儿里得 法

因数 最大公约数

最小公倍数

拓展欧几里得 算法

结束

# Questions?