Алгебра и теория чисел

Максим Бидва (Б08)
Иван Бочков (Б02, Б03)
Борис Золотов (Б10)
Иван Казменко (Б01, Б05)
Владислав Макаров (Б04, Б09)
Тимофей Москаленко (Б06, Б11)
Фёдор Ушаков (Б07)
Алексей Янин (Б11)

Санкт-Петербургский Государственный Университет
Факультет математики и компьютерных наук

Среда, 21 октября 2026 года

Содержание

1. Алгоритм Евклида
   - Наибольший общий делитель
   - Наименьшее общее кратное
   - Код
   - Линейное представление НОД
2. Вычисления по модулю
   - Работа с остатками по модулю
   - «Медленное» умножение
   - «Быстрое» возведение в степень
   - Код
3. Алгоритмы за sqrt n
   - Задачи
   - Код
4. Обратный элемент по модулю
   - Простой модуль
   - Любой модуль
5. Дискретный логарифм
   - Задача
   - Примеры
   - Как устроены степени
   - Baby-step giant-step
   - Код
6. Решето Эратосфена
   - Алгоритм
   - Код
7. Линейные рекуррентности
   - Числа Фибоначчи
   - Пример
   - Пример с многочленом

Содержание

1. Алгоритм Евклида
   - Наибольший общий делитель
   - Наименьшее общее кратное
   - Код
   - Линейное представление НОД
2. Вычисления по модулю
   - Работа с остатками по модулю
   - «Медленное» умножение
   - «Быстрое» возведение в степень
   - Код
3. Алгоритмы за sqrt n
   - Задачи
   - Код
4. Обратный элемент по модулю
   - Простой модуль
   - Любой модуль
5. Дискретный логарифм
   - Задача
   - Примеры
   - Как устроены степени
   - Baby-step giant-step
   - Код
6. Решето Эратосфена
   - Алгоритм
   - Код
7. Линейные рекуррентности
   - Числа Фибоначчи
   - Пример
   - Пример с многочленом

Наибольший общий делитель

Задача:

Даны целые числа a и b.

Найдите их наибольший общий делитель.

- Пример: $\gcd(4, 6) = 2$

Наибольший общий делитель

Задача:

Даны целые числа a и b.

Найдите их наибольший общий делитель.

- Пример: $\gcd(4, 6) = 2$
- $\gcd(a, b) = \gcd(a - b, b)$

Наибольший общий делитель

Задача:

Даны целые числа a и b.

Найдите их наибольший общий делитель.

- Пример: $\gcd(4, 6) = 2$
- $\gcd(a, b) = \gcd(a - b, b)$
- $\gcd(a, b) = \gcd(a \bmod b, b)$

Наибольший общий делитель

Задача:

Даны целые числа a и b.

Найдите их наибольший общий делитель.

- Пример: $\gcd(4, 6) = 2$
- $\gcd(a, b) = \gcd(a - b, b)$
- $\gcd(a, b) = \gcd(a \bmod b, b)$
- $\gcd(a, 0) = a$

Наименьшее общее кратное

Задача:

Даны целые числа a и b.

Найдите их наименьшее общее кратное.

- Пример: $\mathrm{lcm}(4, 6) = 12$

Наименьшее общее кратное

Задача:

Даны целые числа a и b.

Найдите их наименьшее общее кратное.

- Пример: $\operatorname{lcm}(4, 6) = 12$
- $\gcd(a, b) \cdot \operatorname{lcm}(a, b) = a \cdot b$

Наименьшее общее кратное

Задача:

Даны целые числа a и b.

Найдите их наименьшее общее кратное.

- Пример: $\operatorname{lcm}(4, 6) = 12$
- $\gcd(a, b) \cdot \operatorname{lcm}(a, b) = a \cdot b$
- Доказательство:

  Рассмотрим степень каждого простого числа.

  $a \vdots p^{\alpha}$

  $b \vdots p^{\beta}$

  $\gcd(a, b) \vdots p^{\min(\alpha,\beta)}$

  $\operatorname{lcm}(a, b) \vdots p^{\max(\alpha,\beta)}$

  $\alpha + \beta = \min(\alpha, \beta) + \max(\alpha, \beta)$

Код

```cpp
#include <iostream>

using namespace std;

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

int main () {
    int a, b;
    cin >> a >> b;
    cout << gcd (a, b) << endl;
    return 0;
}
```

Ввод:

```
500000
200000
```

Вывод:

```
100000
```

Код

```cpp
#include <iostream>

using namespace std;

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

int lcm (int a, int b) {
    return a / gcd (a, b) * b;
}

int main () {
    int a, b;
    cin >> a >> b;
    cout << lcm (a, b) << endl;
    return 0;
}
```

Ввод:

```
500000
200000
```

Вывод:

```
1000000
```

Код

```cpp
#include <cassert>
#include <iostream>
using namespace std;

int gcdExt (int a, int b, int & x, int & y) {
    if (a == 0) {x = 0; y = 1; return b;}
    auto res = gcdExt (b % a, a, y, x);
    x -= (b / a) * y;
    assert (x * a + y * b == res);

    return res;
}

int main () {
    int a, b, x, y;
    cin >> a >> b;
    cout << gcdExt (a, b, x, y) << endl;
    return 0;
}
```

Ввод:

```
12 5
```

Вывод:

```
1
```

Код

```cpp
#include <iostream>

using namespace std;

int gcdExt (int a, int b, int & x, int & y) {
    if (a == 0) {x = 0; y = 1; return b;}
    auto res = gcdExt (b % a, a, y, x);
    x -= (b / a) * y;
    cout << x << " * " << a << " + " <<
        y << " * " << b << " = " << res << endl;
    return res;
}

int main () {
    int a, b, x, y;
    cin >> a >> b;
    cout << gcdExt (a, b, x, y) << endl;
    return 0;
}
```

Ввод:

```
12 5
```

Вывод:

```
1 * 1 + 0 * 2 = 1
-2 * 2 + 1 * 5 = 1
5 * 5 + -2 * 12 = 1
-2 * 12 + 5 * 5 = 1
1
```

Линейное представление НОД

Задача:

Даны целые числа a и b. Пусть $d = \gcd(a, b)$.

Найдите такие целые числа x и y, что $xa + yb = d$.

- Пример: $a = 12,\ b = 5$: $(-2) \cdot 12 + 5 \cdot 5 = 1$

Линейное представление НОД

Задача:

Даны целые числа a и b. Пусть $d = \gcd(a, b)$.

Найдите такие целые числа x и y, что $xa + yb = d$.

- Пример: $a = 12,\ b = 5$: $(-2) \cdot 12 + 5 \cdot 5 = 1$
- $y' \cdot (b \bmod a) + x' \cdot a = d$

Линейное представление НОД

Задача:

Даны целые числа a и b. Пусть $d = \gcd(a, b)$.

Найдите такие целые числа x и y, что $xa + yb = d$.

- Пример: $a = 12,\ b = 5$: $(-2) \cdot 12 + 5 \cdot 5 = 1$
- $y' \cdot (b \bmod a) + x' \cdot a = d$
- $y \cdot b + x \cdot a = d$

Линейное представление НОД

Задача:

Даны целые числа a и b. Пусть $d = \gcd(a, b)$.

Найдите такие целые числа x и y, что $xa + yb = d$.

- Пример: $a = 12,\ b = 5$: $(-2) \cdot 12 + 5 \cdot 5 = 1$
- $y' \cdot (b \bmod a) + x' \cdot a = d$
- $y \cdot b + x \cdot a = d$
- $y \cdot (b \bmod a + b \operatorname{div} a \cdot a) + x \cdot a = d$

Линейное представление НОД

Задача:

Даны целые числа a и b. Пусть $d = \gcd(a, b)$.

Найдите такие целые числа x и y, что $xa + yb = d$.

- Пример: $a = 12,\ b = 5$: $(-2) \cdot 12 + 5 \cdot 5 = 1$
- $y' \cdot (b \bmod a) + x' \cdot a = d$
- $y \cdot b + x \cdot a = d$
- $y \cdot (b \bmod a + b \operatorname{div} a \cdot a) + x \cdot a = d$
- $y \cdot (b \bmod a) + (x + b \operatorname{div} a \cdot y) \cdot a = d$

Линейное представление НОД

Задача:

Даны целые числа a и b. Пусть $d = \gcd(a, b)$.
Найдите такие целые числа x и y, что $xa + yb = d$.

- Пример: $a = 12,\ b = 5$: $(-2) \cdot 12 + 5 \cdot 5 = 1$
- $y' \cdot (b \bmod a) + x' \cdot a = d$
- $y \cdot b + x \cdot a = d$
- $y \cdot (b \bmod a + b \operatorname{div} a \cdot a) + x \cdot a = d$
- $y \cdot (b \bmod a) + (x + b \operatorname{div} a \cdot y) \cdot a = d$
- $y = y',\ x = x' - b \operatorname{div} a \cdot y$

Содержание

1. Алгоритм Евклида
   - Наибольший общий делитель
   - Наименьшее общее кратное
   - Код
   - Линейное представление НОД
2. Вычисления по модулю
   - Работа с остатками по модулю
   - «Медленное» умножение
   - «Быстрое» возведение в степень
   - Код
3. Алгоритмы за sqrt n
   - Задачи
   - Код
4. Обратный элемент по модулю
   - Простой модуль
   - Любой модуль
5. Дискретный логарифм
   - Задача
   - Примеры
   - Как устроены степени
   - Baby-step giant-step
   - Код
6. Решето Эратосфена
   - Алгоритм
   - Код
7. Линейные рекуррентности
   - Числа Фибоначчи
   - Пример
   - Пример с многочленом

Работа с остатками по модулю

Постановка задачи: нужно производить арифметические действия над целыми числами по модулю m.

Работа с остатками по модулю

Постановка задачи: нужно производить арифметические действия над целыми числами по модулю m.

- **Лемма 1:** $(a \pm b) \bmod m = ((a \bmod m) \pm (b \bmod m)) \bmod m$.

Работа с остатками по модулю

Постановка задачи: нужно производить арифметические действия над целыми числами по модулю m.

- **Лемма 1:** $(a \pm b) \bmod m = ((a \bmod m) \pm (b \bmod m)) \bmod m$.
- **Лемма 2:** $(a \cdot b) \bmod m = ((a \bmod m) \cdot (b \bmod m)) \bmod m$.

Работа с остатками по модулю

Постановка задачи: нужно производить арифметические действия над целыми числами по модулю m.

- **Лемма 1:** $(a \pm b) \bmod m = ((a \bmod m) \pm (b \bmod m)) \bmod m$.
- **Лемма 2:** $(a \cdot b) \bmod m = ((a \bmod m) \cdot (b \bmod m)) \bmod m$.
- Для деления такое равенство неверно.

Работа с остатками по модулю

Постановка задачи: нужно производить арифметические действия над целыми числами по модулю m.

- **Лемма 1:** $(a \pm b) \bmod m = ((a \bmod m) \pm (b \bmod m)) \bmod m$.
- **Лемма 2:** $(a \cdot b) \bmod m = ((a \bmod m) \cdot (b \bmod m)) \bmod m$.
- Для деления такое равенство неверно.
- Python: $(-37) \bmod 10 = 3$
  Другие языки: $(-37) \bmod 10 = -7$

Работа с остатками по модулю

Постановка задачи: нужно производить арифметические действия над целыми числами по модулю m.

- **Лемма 1:** $(a \pm b) \bmod m = ((a \bmod m) \pm (b \bmod m)) \bmod m$.
- **Лемма 2:** $(a \cdot b) \bmod m = ((a \bmod m) \cdot (b \bmod m)) \bmod m$.
- Для деления такое равенство неверно.
- Python: $(-37) \bmod 10 = 3$
  Другие языки: $(-37) \bmod 10 = -7$
- Почему: div и mod вычисляются одновременно, и при этом div округляет к нулю.

«Медленное» умножение

Постановка задачи: вычисление $(a \cdot b) \bmod m$.

«Медленное» умножение

Постановка задачи: вычисление $(a \cdot b) \bmod m$.

Если числа порядка m^2 помещаются в тип данных, можно просто применить Лемму 2:

$(a \cdot b) \bmod m = ((a \bmod m) \cdot (b \bmod m)) \bmod m$.

«Медленное» умножение

Постановка задачи: вычисление $(a \cdot b) \bmod m$.
Если же число m^2 слишком велико, можно произвести умножение «в столбик» в двоичной записи:

- Вычислим $a \bmod m$, $2a \bmod m$, $4a \bmod m$, $8a \bmod m$, ...
- Рассмотрим двоичную запись b и просуммируем нужные слагаемые с предыдущего шага, после каждого сложения вычисляя остаток по модулю m.
- **Пример:** $a = 13$, $b = 10$, $m = 21$.
  $b = 10_{10} = 1010_2 = 2 + 8$
  $(1 \cdot a) \bmod m = 13$
  $(2 \cdot a) \bmod m = (13 + 13) \bmod 21 = 26 \bmod 21 = 5$
  $(4 \cdot a) \bmod m = (5 + 5) \bmod 21 = 10$
  $(8 \cdot a) \bmod m = (10 + 10) \bmod 21 = 20$
  Ответ: $(((2 \cdot a) \bmod m) + ((8 \cdot a) \bmod m)) \bmod m = (5 + 20) \bmod m = 25 \bmod 21 = 4$.

«Медленное» умножение

Постановка задачи: вычисление $(a \cdot b) \bmod m$.
Если же число m^2 слишком велико, можно произвести умножение «в столбик» в двоичной записи:

- Вычислим $a \bmod m$, $2a \bmod m$, $4a \bmod m$, $8a \bmod m$, ...
- Рассмотрим двоичную запись b и просуммируем нужные слагаемые с предыдущего шага, после каждого сложения вычисляя остаток по модулю m.
- **Пример:** $a = 13$, $b = 10$, $m = 21$.
  $b = 10_{10} = 1010_2 = 2 + 8$
  $(1 \cdot a) \bmod m = 13$
  $(2 \cdot a) \bmod m = (13 + 13) \bmod 21 = 26 \bmod 21 = 5$
  $(4 \cdot a) \bmod m = (5 + 5) \bmod 21 = 10$
  $(8 \cdot a) \bmod m = (10 + 10) \bmod 21 = 20$
  Ответ: $(((2 \cdot a) \bmod m) + ((8 \cdot a) \bmod m)) \bmod m = (5 + 20) \bmod m = 25 \bmod 21 = 4$.

«Медленное» умножение

Постановка задачи: вычисление $(a \cdot b) \bmod m$.
Если же число m^2 слишком велико, можно произвести умножение «в столбик» в двоичной записи:

- Вычислим $a \bmod m$, $2a \bmod m$, $4a \bmod m$, $8a \bmod m$, ...
- Рассмотрим двоичную запись b и просуммируем нужные слагаемые с предыдущего шага, после каждого сложения вычисляя остаток по модулю m.
- **Пример:** $a = 13$, $b = 10$, $m = 21$.
  $b = 10_{10} = 1010_2 = 2 + 8$
  $(1 \cdot a) \bmod m = 13$
  $(2 \cdot a) \bmod m = (13 + 13) \bmod 21 = 26 \bmod 21 = 5$
  $(4 \cdot a) \bmod m = (5 + 5) \bmod 21 = 10$
  $(8 \cdot a) \bmod m = (10 + 10) \bmod 21 = 20$
  Ответ: $(((2 \cdot a) \bmod m) + ((8 \cdot a) \bmod m)) \bmod m = (5 + 20) \bmod m = 25 \bmod 21 = 4$.

«Медленное» умножение

Постановка задачи: вычисление $(a \cdot b) \bmod m$.
Если же число m^2 слишком велико, можно произвести умножение «в столбик» в двоичной записи:

- Вычислим $a \bmod m$, $2a \bmod m$, $4a \bmod m$, $8a \bmod m$, ...
- Рассмотрим двоичную запись b и просуммируем нужные слагаемые с предыдущего шага, после каждого сложения вычисляя остаток по модулю m.
- **Пример:** $a = 13$, $b = 10$, $m = 21$.
  $b = 10_{10} = 1010_2 = 2 + 8$
  $(1 \cdot a) \bmod m = 13$
  $(2 \cdot a) \bmod m = (13 + 13) \bmod 21 = 26 \bmod 21 = 5$
  $(4 \cdot a) \bmod m = (5 + 5) \bmod 21 = 10$
  $(8 \cdot a) \bmod m = (10 + 10) \bmod 21 = 20$
  Ответ: $(((2 \cdot a) \bmod m) + ((8 \cdot a) \bmod m)) \bmod m = (5 + 20) \bmod m = 25 \bmod 21 = 4$.

«Медленное» умножение

Постановка задачи: вычисление $(a \cdot b) \bmod m$.
Если же число m^2 слишком велико, можно произвести умножение «в столбик» в двоичной записи:

- **Пример:** $a = 13,\ b = 10,\ m = 21.$
  $b = 10_{10} = 1010_2 = 2 + 8$
  $(1 \cdot a) \bmod m = 13$
  $(2 \cdot a) \bmod m = (13 + 13) \bmod 21 = 26 \bmod 21 = 5$
  $(4 \cdot a) \bmod m = (5 + 5) \bmod 21 = 10$
  $(8 \cdot a) \bmod m = (10 + 10) \bmod 21 = 20$
  Ответ: $(((2 \cdot a) \bmod m) + ((8 \cdot a) \bmod m)) \bmod m =$
  $(5 + 20) \bmod m = 25 \bmod 21 = 4.$
- **Проверка:**
  $(13 \cdot 10) \bmod 21 = 130 \bmod 21 = (126 + 4) \bmod 21 = 4.$
- Заметим, что при вычислениях могут получиться только числа от 0 до $2 \cdot m - 2$.

«Медленное» умножение

Постановка задачи: вычисление $(a \cdot b) \bmod m$.
Если же число m^2 слишком велико, можно произвести умножение «в столбик» в двоичной записи:

- **Пример:** $a = 13$, $b = 10$, $m = 21$.
  $b = 10_{10} = 1010_2 = 2 + 8$
  $(1 \cdot a) \bmod m = 13$
  $(2 \cdot a) \bmod m = (13 + 13) \bmod 21 = 26 \bmod 21 = 5$
  $(4 \cdot a) \bmod m = (5 + 5) \bmod 21 = 10$
  $(8 \cdot a) \bmod m = (10 + 10) \bmod 21 = 20$
  Ответ: $(((2 \cdot a) \bmod m) + ((8 \cdot a) \bmod m)) \bmod m = (5 + 20) \bmod m = 25 \bmod 21 = 4$.
- **Проверка:**
  $(13 \cdot 10) \bmod 21 = 130 \bmod 21 = (126 + 4) \bmod 21 = 4$.
- Заметим, что при вычислениях могут получиться только числа от 0 до $2 \cdot m - 2$.

«Медленное» умножение

Постановка задачи: вычисление $(a \cdot b) \bmod m$.
Если же число m^2 слишком велико, можно произвести умножение «в столбик» в двоичной записи:

- Вычислим $a \bmod m$, $2a \bmod m$, $4a \bmod m$, $8a \bmod m$, ...
- Рассмотрим двоичную запись b и просуммируем нужные слагаемые с предыдущего шага, после каждого сложения вычисляя остаток по модулю m.
- **Время работы:** $\log_2 b$ сложений по модулю для вычисления $(2^k \cdot a) \bmod m$ и не более $\log_2 b$ сложений по модулю для суммирования нужных слагаемых.

«Быстрое» возведение в степень

Постановка задачи: вычисление $a^b \bmod m$.

«Быстрое» возведение в степень

Постановка задачи: вычисление $a^b \bmod m$.

Будем действовать аналогично: умножение можно представить как последовательность сложений, а возведение в степень — как последовательность умножений.

- Вычислим $a \bmod m$, $a^2 \bmod m$, $a^4 \bmod m$, $a^8 \bmod m$, ...
- Рассмотрим двоичную запись b и вычислим произведение нужных множителей с предыдущего шага, после каждого умножения вычисляя остаток по модулю m.

«Быстрое» возведение в степень

Постановка задачи: вычисление $a^b \bmod m$.

Будем действовать аналогично: умножение можно представить как последовательность сложений, а возведение в степень — как последовательность умножений.

- Вычислим $a \bmod m$, $a^2 \bmod m$, $a^4 \bmod m$, $a^8 \bmod m$, ...
- Рассмотрим двоичную запись b и вычислим произведение нужных множителей с предыдущего шага, после каждого умножения вычисляя остаток по модулю m.
- **Пример:** $a = 13$, $b = 5$, $m = 21$.
  $b = 5_{10} = 101_2 = 1 + 4$
  $a^1 \bmod m = 13$
  $a^2 \bmod m = (13 \cdot 13) \bmod 21 = 169 \bmod 21 = 1$
  $a^4 \bmod m = (1 \cdot 1) \bmod 21 = 1$
  Ответ: $((a^1 \bmod m) \cdot (a^4 \bmod m)) \bmod m = (13 \cdot 1) \bmod m = 13$.

«Быстрое» возведение в степень

Постановка задачи: вычисление $a^b \bmod m$.

Будем действовать аналогично: умножение можно представить как последовательность сложений, а возведение в степень — как последовательность умножений.

- Вычислим $a \bmod m$, $a^2 \bmod m$, $a^4 \bmod m$, $a^8 \bmod m$, ...
- Рассмотрим двоичную запись b и вычислим произведение нужных множителей с предыдущего шага, после каждого умножения вычисляя остаток по модулю m.
- **Пример:** $a = 13$, $b = 5$, $m = 21$.
  $b = 5_{10} = 101_2 = 1 + 4$
  $a^1 \bmod m = 13$
  $a^2 \bmod m = (13 \cdot 13) \bmod 21 = 169 \bmod 21 = 1$
  $a^4 \bmod m = (1 \cdot 1) \bmod 21 = 1$
  Ответ: $((a^1 \bmod m) \cdot (a^4 \bmod m)) \bmod m = (13 \cdot 1) \bmod m = 13$.

«Быстрое» возведение в степень

Постановка задачи: вычисление $a^b \bmod m$.

Будем действовать аналогично: умножение можно представить как последовательность сложений, а возведение в степень — как последовательность умножений.

- Вычислим $a \bmod m$, $a^2 \bmod m$, $a^4 \bmod m$, $a^8 \bmod m$, ...
- Рассмотрим двоичную запись b и вычислим произведение нужных множителей с предыдущего шага, после каждого умножения вычисляя остаток по модулю m.
- **Пример:** $a = 13$, $b = 5$, $m = 21$.
  $b = 5_{10} = 101_2 = 1 + 4$
  $a^1 \bmod m = 13$
  $a^2 \bmod m = (13 \cdot 13) \bmod 21 = 169 \bmod 21 = 1$
  $a^4 \bmod m = (1 \cdot 1) \bmod 21 = 1$
  Ответ: $((a^1 \bmod m) \cdot (a^4 \bmod m)) \bmod m = (13 \cdot 1) \bmod m = 13$.

«Быстрое» возведение в степень

Постановка задачи: вычисление $a^b \bmod m$.

Будем действовать аналогично: умножение можно представить как последовательность сложений, а возведение в степень — как последовательность умножений.

- **Пример:** $a = 13,\ b = 5,\ m = 21$.
  $b = 5_{10} = 101_2 = 1 + 4$
  $a^1 \bmod m = 13$
  $a^2 \bmod m = (13 \cdot 13) \bmod 21 = 169 \bmod 21 = 1$
  $a^4 \bmod m = (1 \cdot 1) \bmod 21 = 1$
  Ответ: $((a^1 \bmod m) \cdot (a^4 \bmod m)) \bmod m = (13 \cdot 1) \bmod m = 13$.
- **Проверка:** $(13^5) \bmod 21 = 371\,293 \bmod 21 = 13$.
- Заметим, что при вычислениях могут получиться только числа от 0 до $(m-1)^2$.
- Если умножения по модулю реализовать при помощи «медленного» умножения, при вычислениях могут вновь получиться только числа от 0 до $2 \cdot m - 2$.

«Быстрое» возведение в степень

Постановка задачи: вычисление $a^b \bmod m$.

Будем действовать аналогично: умножение можно представить как последовательность сложений, а возведение в степень — как последовательность умножений.

- **Пример:** $a = 13$, $b = 5$, $m = 21$.
  $b = 5_{10} = 101_2 = 1 + 4$
  $a^1 \bmod m = 13$
  $a^2 \bmod m = (13 \cdot 13) \bmod 21 = 169 \bmod 21 = 1$
  $a^4 \bmod m = (1 \cdot 1) \bmod 21 = 1$
  Ответ: $((a^1 \bmod m) \cdot (a^4 \bmod m)) \bmod m = (13 \cdot 1) \bmod m = 13$.
- **Проверка:** $(13^5) \bmod 21 = 371\,293 \bmod 21 = 13$.
- Заметим, что при вычислениях могут получиться только числа от 0 до $(m-1)^2$.
- Если умножения по модулю реализовать при помощи «медленного» умножения, при вычислениях могут вновь получиться только числа от 0 до $2 \cdot m - 2$.

«Быстрое» возведение в степень

Постановка задачи: вычисление $a^b \bmod m$.

Будем действовать аналогично: умножение можно представить как последовательность сложений, а возведение в степень — как последовательность умножений.

- **Пример:** $a = 13,\ b = 5,\ m = 21$.
  $b = 5_{10} = 101_2 = 1 + 4$
  $a^1 \bmod m = 13$
  $a^2 \bmod m = (13 \cdot 13) \bmod 21 = 169 \bmod 21 = 1$
  $a^4 \bmod m = (1 \cdot 1) \bmod 21 = 1$
  Ответ: $((a^1 \bmod m) \cdot (a^4 \bmod m)) \bmod m = (13 \cdot 1) \bmod m = 13$.
- **Проверка:** $(13^5) \bmod 21 = 371\,293 \bmod 21 = 13$.
- Заметим, что при вычислениях могут получиться только числа от 0 до $(m-1)^2$.
- Если умножения по модулю реализовать при помощи «медленного» умножения, при вычислениях могут вновь получиться только числа от 0 до $2 \cdot m - 2$.

«Быстрое» возведение в степень

Постановка задачи: вычисление $a^b \bmod m$.

Будем действовать аналогично: умножение можно представить как последовательность сложений, а возведение в степень — как последовательность умножений.

- Вычислим $a \bmod m$, $a^2 \bmod m$, $a^4 \bmod m$, $a^8 \bmod m$, ...
- Рассмотрим двоичную запись b и вычислим произведение нужных множителей с предыдущего шага, после каждого умножения вычисляя остаток по модулю m.
- **Время работы:** $\log_2 b$ умножений по модулю для вычисления $a^{2^k} \bmod m$ и не более $\log_2 b$ умножений по модулю для перемножения нужных сомножителей.

Код

```cpp
#include <iostream>
using namespace std;

int mulMod (int a, int b, int m) {
    int res = 0;
    for ( ; b > 0; b >>= 1) {
        if (b & 1)
            res = (res + a) % m;
        a = (a + a) % m;
    }
    return res;
}

int main () {
    int a, b, m;
    cin >> a >> b >> m;
    cout << mulMod (a, b, m) << endl;
    return 0;
}
```

Ввод:

```
13 10 21
```

Вывод:

```
4
```

Код

```cpp
#include <iostream>
using namespace std;

int powMod (int a, int b, int m) {
    int res = 1 % m;
    for ( ; b > 0; b >>= 1) {
        if (b & 1)
            res = (res * 1LL * a) % m;
        a = (a * 1LL * a) % m;
    }
    return res;
}

int main () {
    int a, b, m;
    cin >> a >> b >> m;
    cout << powMod (a, b, m) << endl;
    return 0;
}
```

Ввод:

```
13 5 21
```

Вывод:

```
13
```

Содержание

1. Алгоритм Евклида
   - Наибольший общий делитель
   - Наименьшее общее кратное
   - Код
   - Линейное представление НОД
2. Вычисления по модулю
   - Работа с остатками по модулю
   - «Медленное» умножение
   - «Быстрое» возведение в степень
   - Код
3. Алгоритмы за sqrt n
   - Задачи
   - Код
4. Обратный элемент по модулю
   - Простой модуль
   - Любой модуль
5. Дискретный логарифм
   - Задача
   - Примеры
   - Как устроены степени
   - Baby-step giant-step
   - Код
6. Решето Эратосфена
   - Алгоритм
   - Код
7. Линейные рекуррентности
   - Числа Фибоначчи
   - Пример
   - Пример с многочленом

Задачи

Дано число n.

- **Задача 1:** верно ли, что n простое?
- Примеры: $n = 89$ простое, $n = 87$ составное.
- **Задача 2:** найдите разложения числа n на простые множители.
- **Задача 3:** найдите функцию Эйлера $\varphi(n)$: количество чисел от 0 до $n - 1$, взаимно простых с n.
- Как найти значение функции Эйлера?

Задачи

Дано число n.

- **Задача 1:** верно ли, что n простое?
- Примеры: $n = 89$ простое, $n = 87$ составное.
- **Задача 2:** найдите разложения числа n на простые множители.
- **Задача 3:** найдите функцию Эйлера $\varphi(n)$: количество чисел от 0 до $n - 1$, взаимно простых с n.
- Как найти значение функции Эйлера?
- $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdots p_k^{\alpha_k}$
  $\varphi(n) = n \cdot \frac{p_1-1}{p_1} \cdot \frac{p_2-1}{p_2} \cdots \frac{p_k-1}{p_k}$

Код

```cpp
#include <iostream>
using namespace std;

bool prime (int n) {
    if (n < 2) return false;
    for (int d = 2; d * d <= n; d++) {
        if (n % d == 0) {
            return false;
        }
    }
    return true;
}

int main () {
    int k;
    while (cin >> k) {
        cout << prime (k) << endl;
    }
    return 0;
}
```

Ввод:

```
89
87
```

Вывод:

```
1
0
```

Код

```cpp
#include <iostream>
#include <utility>
#include <vector>
using namespace std;
vector <pair <int, int> > divisors (int n) {
    vector <pair <int, int> > res;
    for (int d = 2; d * d <= n; d++) {
        if (n % d == 0) {
            res.push_back ({d, 0});
            while (n % d == 0) {
                n /= d;
                res.back ().second += 1;
            }
        }
    }
    if (n > 1) res.push_back ({n, 1});
    return res;
}
int main () {
    int k;
    while (cin >> k) {
        auto d = divisors (k);
        cout << k << ":" << endl;
        for (auto p : d) cout << p.first << "^" << p.second << endl;
    }
    return 0;
}
```

Ввод:

```
12
58
81
1
7000
```

Вывод:

```
12:
2^2
3^1
58:
2^1
29^1
81:
3^4
1:
7000:
2^3
5^3
7^1
```

Код

```cpp
#include <iostream>
using namespace std;

int phi (int n) {
    int res = n;
    for (int d = 2; d * d <= n; d++) {
        if (n % d == 0) {
            res = res / d * (d - 1);
            while (n % d == 0)
                n /= d;
        }
    }
    if (n > 1) res = res / n * (n - 1);
    return res;
}

int main () {
    int k;
    while (cin >> k) {
        cout << phi (k) << endl;
    }
    return 0;
}
```

Ввод:

```
7
10
12
9
1
```

Вывод:

```
6
4
4
6
1
```

Содержание

1. Алгоритм Евклида
   - Наибольший общий делитель
   - Наименьшее общее кратное
   - Код
   - Линейное представление НОД
2. Вычисления по модулю
   - Работа с остатками по модулю
   - «Медленное» умножение
   - «Быстрое» возведение в степень
   - Код
3. Алгоритмы за sqrt n
   - Задачи
   - Код
4. Обратный элемент по модулю
   - Простой модуль
   - Любой модуль
5. Дискретный логарифм
   - Задача
   - Примеры
   - Как устроены степени
   - Baby-step giant-step
   - Код
6. Решето Эратосфена
   - Алгоритм
   - Код
7. Линейные рекуррентности
   - Числа Фибоначчи
   - Пример
   - Пример с многочленом

Простой модуль

Даны числа a и $p \in \mathbb{P}$ $(0 \leqslant a < p)$.
Найдите такое b, что $(a \cdot b) \bmod p = 1$.
Другая запись: $a \cdot b \equiv 1 \pmod{p}$.
Обозначение: $b = a^{-1}$.

Простой модуль

Даны числа a и $p \in \mathbb{P}$ $(0 \leqslant a < p)$.
Найдите такое b, что $(a \cdot b) \bmod p = 1$.
Другая запись: $a \cdot b \equiv 1 \pmod{p}$.
Обозначение: $b = a^{-1}$.

- **Малая теорема Ферма:** $a^{p-1} \equiv 1 \pmod{p}$.

Простой модуль

Даны числа a и $p \in \mathbb{P}$ $(0 \leqslant a < p)$.
Найдите такое b, что $(a \cdot b) \bmod p = 1$.
Другая запись: $a \cdot b \equiv 1 \pmod{p}$.
Обозначение: $b = a^{-1}$.

- **Малая теорема Ферма:** $a^{p-1} \equiv 1 \pmod{p}$.
- *А значит (?),* $a^{p-2} \equiv a^{-1} \pmod{p}$.

Простой модуль

Даны числа a и $p \in \mathbb{P}$ $(0 \leqslant a < p)$.
Найдите такое b, что $(a \cdot b) \bmod p = 1$.
Другая запись: $a \cdot b \equiv 1 \pmod{p}$.
Обозначение: $b = a^{-1}$.

- **Малая теорема Ферма:** $a^{p-1} \equiv 1 \pmod{p}$.
- *А значит (?),* $a^{p-2} \equiv a^{-1} \pmod{p}$.
- Пример: $p = 7$.

a^b	0	1	2	3	4	5	6	7	8
0	?	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1
2	1	2	4	1	2	4	1	2	4
3	1	3	2	6	4	5	1	3	2
4	1	4	2	1	4	2	1	4	2
5	1	5	4	6	2	3	1	5	4
6	1	6	1	6	1	6	1	6	1

Любой модуль

Даны числа a и n $(0 \leqslant a < n)$. Найдите $b = a^{-1}$.

Любой модуль

Даны числа a и n $(0 \leqslant a < n)$. Найдите $b = a^{-1}$.

- **Теорема Эйлера:** $\gcd(a, n) = 1 \Rightarrow a^{\varphi(n)} \equiv 1 \pmod{n}$.

Любой модуль

Даны числа a и n $(0 \leqslant a < n)$. Найдите $b = a^{-1}$.

- **Теорема Эйлера:** $\gcd(a, n) = 1 \Rightarrow a^{\varphi(n)} \equiv 1 \pmod{n}$.
- *А значит (?),* $a^{\varphi(n)-1} \equiv a^{-1} \pmod{n}$, когда ответ существует.

Любой модуль

Даны числа a и n $(0 \leqslant a < n)$. Найдите $b = a^{-1}$.

- **Теорема Эйлера:** $\gcd(a, n) = 1 \Rightarrow a^{\varphi(n)} \equiv 1 \pmod{n}$.
- *А значит (?),* $a^{\varphi(n)-1} \equiv a^{-1} \pmod{n}$, когда ответ существует.
- Пример: $n = 10$.

a^b	0	1	2	3	4	5	6	7	8
0	?	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1
2	1	2	4	8	6	2	4	8	6
3	1	3	9	7	1	3	9	7	1
4	1	4	6	4	6	4	6	4	6
5	1	5	5	5	5	5	5	5	5
6	1	6	6	6	6	6	6	6	6
7	1	7	9	3	1	7	9	3	1
8	1	4	2	6	8	4	2	6	8
9	1	9	1	9	1	9	1	9	1

Содержание

1. Алгоритм Евклида
   - Наибольший общий делитель
   - Наименьшее общее кратное
   - Код
   - Линейное представление НОД
2. Вычисления по модулю
   - Работа с остатками по модулю
   - «Медленное» умножение
   - «Быстрое» возведение в степень
   - Код
3. Алгоритмы за sqrt n
   - Задачи
   - Код
4. Обратный элемент по модулю
   - Простой модуль
   - Любой модуль
5. Дискретный логарифм
   - Задача
   - Примеры
   - Как устроены степени
   - Baby-step giant-step
   - Код
6. Решето Эратосфена
   - Алгоритм
   - Код
7. Линейные рекуррентности
   - Числа Фибоначчи
   - Пример
   - Пример с многочленом

Задача

- Даны основание a, число v и модуль m.
- Рассмотрим числа $a^0 \bmod m$, $a^1 \bmod m$, $a^2 \bmod m$, $\ldots$
- Найдём первое из них, которое равно v: $a^k \bmod m = v$.
- Показатель степени k — это и есть дискретный логарифм.

Задача

- Даны основание a, число v и модуль m.
- Рассмотрим числа $a^0 \bmod m$, $a^1 \bmod m$, $a^2 \bmod m$, $\ldots$
- Найдём первое из них, которое равно v: $a^k \bmod m = v$.
- Показатель степени k — это и есть дискретный логарифм.
- Обычно достаточно найти не минимальное k, а любое.

Примеры

- Наивное решение: будем вычислять $a^0 \bmod m$, $a^1 \bmod m$, $a^2 \bmod m$, …, пока очередная степень не станет равна v.
- Каким может быть $k \geqslant 0$, для которого $a^k \bmod m = v$?

Примеры

- Наивное решение: будем вычислять $a^0 \bmod m$, $a^1 \bmod m$, $a^2 \bmod m$, …, пока очередная степень не станет равна v.
- Каким может быть $k \geqslant 0$, для которого $a^k \bmod m = v$?
- Пример: $m = 13$, $a = 2$.

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$a^k \bmod m$	1	2	4	8	3	6	12	11	9	5	10	7	1	2	4

- Максимальный ответ на 1 меньше, чем порядок элемента.

Примеры

- Наивное решение: будем вычислять $a^0 \bmod m$, $a^1 \bmod m$, $a^2 \bmod m$, ..., пока очередная степень не станет равна v.
- Каким может быть $k \geqslant 0$, для которого $a^k \bmod m = v$?
- Пример: $m = 13$, $a = 2$.

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$a^k \bmod m$	1	2	4	8	3	6	12	11	9	5	10	7	1	2	4

- Максимальный ответ на 1 меньше, чем порядок элемента.
- Пример: $m = 13$, $a = 3$ (иногда ответ не существует).

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$a^k \bmod m$	1	3	9	1	3	9	1	3	9	1	3	9	1	3	9

- Решим задачу в простом случае: m простое, $0 < a, v < m$.

Примеры

- Наивное решение: будем вычислять $a^0 \bmod m$, $a^1 \bmod m$, $a^2 \bmod m$, ..., пока очередная степень не станет равна v.
- Каким может быть $k \geqslant 0$, для которого $a^k \bmod m = v$?
- Пример: $m = 13$, $a = 2$.

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$a^k \bmod m$	1	2	4	8	3	6	12	11	9	5	10	7	1	2	4

- Максимальный ответ на 1 меньше, чем порядок элемента.
- Пример: $m = 13$, $a = 3$ (иногда ответ не существует).

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$a^k \bmod m$	1	3	9	1	3	9	1	3	9	1	3	9	1	3	9

- Решим задачу в простом случае: m простое, $0 < a, v < m$.
- Пример: $m = 12$, $a = 2$ (не взаимно простые).

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$a^k \bmod m$	1	2	4	8	4	8	4	8	4	8	4	8	4	8	4

Как устроены степени

- Пусть m простое и $0 < a, v < m$.
- Тогда $\varphi(m) = m - 1$.
- Если a — первообразный корень, то числа a^0, a^1, a^2, …, $a^{\varphi(m)-1}$ по модулю m принимают все значения от 1 до $m - 1$ в каком-то порядке.
- А дальше $a^{\varphi(m)} = a^0 = 1$, $a^{\varphi(m)+1} = a^1$, $a^{\varphi(m)+2} = a^2$, … по модулю m.

Как устроены степени

- Пусть m простое и $0 < a, v < m$.
- Тогда $\varphi(m) = m - 1$.
- Если a — первообразный корень, то числа a^0, a^1, a^2, …, $a^{\varphi(m)-1}$ по модулю m принимают все значения от 1 до $m - 1$ в каком-то порядке.
- А дальше $a^{\varphi(m)} = a^0 = 1$, $a^{\varphi(m)+1} = a^1$, $a^{\varphi(m)+2} = a^2$, … по модулю m.
- Даже если a — не первообразный корень, всё равно $a^{\varphi(m)} = a^0 = 1$.
- Итак, для любого $a \neq 0$ достаточно проверить для $\varphi(m)$ степеней k подряд, получается ли $a^k \bmod m = v$.
- Если $a^k \bmod m = v$, то $a^{k \pm \varphi(m)} \bmod m = v$.

Baby-step giant-step

- Как среди a^k найти число v быстрее, чем за $\varphi(m)$?
- Пусть $z = \left\lceil \sqrt{\varphi(m)} \right\rceil$.
- Рассмотрим $a^0, a^1, a^2, \ldots, a^{z-1}$ (baby-step).
- Рассмотрим $a^0, a^z, a^{2z} \ldots, a^{(z-1)z}$ (giant-step).

Baby-step giant-step

- Как среди a^k найти число v быстрее, чем за $\varphi(m)$?
- Пусть $z = \left\lceil \sqrt{\varphi(m)} \right\rceil$.
- Рассмотрим $a^0, a^1, a^2, \ldots, a^{z-1}$ (baby-step).
- Рассмотрим $a^0, a^z, a^{2z} \ldots, a^{(z-1)z}$ (giant-step).
- Любое k из $0 \leqslant k < z^2$ можно представить в виде $x \cdot z + y$, где $0 \leqslant x, y < z$.
- Например, при $z = 10$ и $k = 37$ получаем $x = 3$ и $y = 7$.

Baby-step giant-step

- Как среди a^k найти число v быстрее, чем за $\varphi(m)$?
- Пусть $z = \left\lceil \sqrt{\varphi(m)} \right\rceil$.
- Рассмотрим $a^0, a^1, a^2, \ldots, a^{z-1}$ (baby-step).
- Рассмотрим $a^0, a^z, a^{2z} \ldots, a^{(z-1)z}$ (giant-step).
- Любое k из $0 \leqslant k < z^2$ можно представить в виде $x \cdot z + y$, где $0 \leqslant x, y < z$.
- Например, при $z = 10$ и $k = 37$ получаем $x = 3$ и $y = 7$.
- Положим числа a^0, a^1, a^2, …, a^{z-1} в `set`.
- Будем искать числа $v/a^0, v/a^z, v/a^{2z}, \ldots, v/a^{(z-1)z}$ в `set`.
- Поиск выполняется за $\mathcal{O}(\log z)$, общее время $\mathcal{O}(z \log z)$.
- Но как узнать саму степень?

Baby-step giant-step

- Как среди a^k найти число v быстрее, чем за $\varphi(m)$?
- Пусть $z = \left\lceil \sqrt{\varphi(m)} \right\rceil$.
- Рассмотрим $a^0, a^1, a^2, \ldots, a^{z-1}$ (baby-step).
- Рассмотрим $a^0, a^z, a^{2z} \ldots, a^{(z-1)z}$ (giant-step).
- Любое k из $0 \leqslant k < z^2$ можно представить в виде $x \cdot z + y$, где $0 \leqslant x, y < z$.
- Например, при $z = 10$ и $k = 37$ получаем $x = 3$ и $y = 7$.
- Положим соответствия $a^0 \to 0$, $a^1 \to 1$, $a^2 \to 2$, …, $a^{z-1} \to z - 1$ в `map`.
- Будем искать числа $v/a^0, v/a^z, v/a^{2z}, \ldots, v/a^{(z-1)z}$ в `map`.
- Поиск выполняется за $\mathcal{O}(\log z)$, общее время $\mathcal{O}(z \log z)$.
- Если мы нашли $v/a^{xz} \to y$, то $a^{xz+y} = v$, значит, подойдёт $k = xz + y$.

Baby-step giant-step

- Как среди a^k найти число v быстрее, чем за $\varphi(m)$?
- Пусть $z = \left\lceil \sqrt{\varphi(m)} \right\rceil$.
- Рассмотрим $a^0, a^1, a^2, \ldots, a^{z-1}$ (baby-step).
- Рассмотрим $a^0, a^z, a^{2z} \ldots, a^{(z-1)z}$ (giant-step).
- Любое k из $0 \leqslant k < z^2$ можно представить в виде $x \cdot z + y$, где $0 \leqslant x, y < z$.
- Например, при $z = 10$ и $k = 37$ получаем $x = 3$ и $y = 7$.
- Положим соответствия $a^0 \to 0$, $a^1 \to 1$, $a^2 \to 2$, …, $a^{z-1} \to z - 1$ в `map`.
- Будем искать числа $v \cdot a^0, v \cdot a^z, v \cdot a^{2z}, \ldots, v \cdot a^{(z-1)z}$ в `map`.
- Поиск выполняется за $\mathcal{O}(\log z)$, общее время $\mathcal{O}(z \log z)$.
- Если мы нашли $v \cdot a^{xz} \to y$, то $a^{y-xz} = v$, значит, подойдёт $k = y - xz \pmod{\varphi(m)}$.

Код

```cpp
#include <cmath>
#include <iostream>
#include <map>
using namespace std;
int discrete_log (int a, int v, int m) {
    int phi_m = m - 1;
    int lim = int (sqrt (phi_m)) + 1;
    map <int, int> power;
    int a_lim = 1;
    for (int i = 0; i < lim; i++) {
        power[a_lim] = i;
        a_lim = (a_lim * 1LL * a) % m;
    }
    for (int i = 0; i < lim; i++) {
        if (power.count (v))
            return (power[v] - i * lim + phi_m) % phi_m;
        v = (v * 1LL * a_lim) % m;
    }
    return -1;
}
int main () {
    int a, v, m;
    while (cin >> a >> v >> m)
        cout << discrete_log (a, v, m) << endl;
    return 0;
}
```

Ввод:

```
2 7 13
2 1 13
3 7 13
```

Вывод:

```
11
0
-1
```

Содержание

1. Алгоритм Евклида
   - Наибольший общий делитель
   - Наименьшее общее кратное
   - Код
   - Линейное представление НОД
2. Вычисления по модулю
   - Работа с остатками по модулю
   - «Медленное» умножение
   - «Быстрое» возведение в степень
   - Код
3. Алгоритмы за sqrt n
   - Задачи
   - Код
4. Обратный элемент по модулю
   - Простой модуль
   - Любой модуль
5. Дискретный логарифм
   - Задача
   - Примеры
   - Как устроены степени
   - Baby-step giant-step
   - Код
6. Решето Эратосфена
   - Алгоритм
   - Код
7. Линейные рекуррентности
   - Числа Фибоначчи
   - Пример
   - Пример с многочленом

Алгоритм

Задача: найдите все простые числа от 1 до n.

Алгоритм

Задача: найдите все простые числа от 1 до n.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

Алгоритм

Задача: найдите все простые числа от 1 до n.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

Пропустим числа 0 и 1.

Алгоритм

Задача: найдите все простые числа от 1 до n.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

Вычеркнем всё, что делится на 2.

Алгоритм

Задача: найдите все простые числа от 1 до n.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

Вычеркнем всё, что делится на 3.

Алгоритм

Задача: найдите все простые числа от 1 до n.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

Пропустим число 4.

Алгоритм

Задача: найдите все простые числа от 1 до n.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

Вычеркнем всё, что делится на 5.

Алгоритм

Задача: найдите все простые числа от 1 до n.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

Пропустим число 6.

Алгоритм

Задача: найдите все простые числа от 1 до n.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

Вычеркнем всё, что делится на 7.

Алгоритм

Задача: найдите все простые числа от 1 до n.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

Пропустим числа 8 и 9.

Алгоритм

Задача: найдите все простые числа от 1 до n.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

Числа 10 и далее можно не рассматривать.

Код

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main () {
6      int n;
7      cin >> n;
8      vector <int> s (n);
9      for (int i = 2; i * i < n; i++)
10         if (!s[i])
11             for (int j = i * i; j < n; j += i)
12                 s[j] = 1;
13     for (int i = 2; i < n; i++)
14         if (!s[i])
15             cout << i << endl;
16     return 0;
17 }
```

Ввод:

```
30
```

Вывод:

```
2
3
5
7
11
13
17
19
23
29
```

Код

```cpp
#include <iostream>
#include <vector>
using namespace std;

int main () {
    int n;
    cin >> n;
    vector <int> d (n);
    for (int i = 2; i * i < n; i++)
        if (!d[i])
            for (int j = i * i; j < n; j += i)
                if (!d[j])
                    d[j] = i;
    for (int i = 2; i < n; i++)
        cout << i << ": " << d[i] << endl;
    return 0;
}
```

Ввод:

```
16
```

Вывод:

```
2: 0
3: 0
4: 2
5: 0
6: 2
7: 0
8: 2
9: 3
10: 2
11: 0
12: 2
13: 0
14: 2
15: 3
```

Содержание

1. Алгоритм Евклида
   - Наибольший общий делитель
   - Наименьшее общее кратное
   - Код
   - Линейное представление НОД
2. Вычисления по модулю
   - Работа с остатками по модулю
   - «Медленное» умножение
   - «Быстрое» возведение в степень
   - Код
3. Алгоритмы за sqrt n
   - Задачи
   - Код
4. Обратный элемент по модулю
   - Простой модуль
   - Любой модуль
5. Дискретный логарифм
   - Задача
   - Примеры
   - Как устроены степени
   - Baby-step giant-step
   - Код
6. Решето Эратосфена
   - Алгоритм
   - Код
7. Линейные рекуррентности
   - Числа Фибоначчи
   - Пример
   - Пример с многочленом

Числа Фибоначчи

Числа Фибоначчи: $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$ при $n \geqslant 2$.

Первые несколько чисел:

$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233 \ldots$

Задача: найдите $F_n \bmod m$, где n и m до 10^9.

Числа Фибоначчи

Числа Фибоначчи: $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$ при $n \geqslant 2$.

Первые несколько чисел:

$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233 \ldots$

Задача: найдите $F_n \bmod m$, где n и m до 10^9.

$$F_n = F_{n-1} + F_{n-2}$$

Числа Фибоначчи

Числа Фибоначчи: $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$ при $n \geqslant 2$.

Первые несколько чисел:

$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233 \ldots$

Задача: найдите $F_n \bmod m$, где n и m до 10^9.

$$\begin{cases} F_n &= F_{n-1} + F_{n-2} \\ F_{n-1} &= F_{n-1} \end{cases}$$

Числа Фибоначчи

Числа Фибоначчи: $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$ при $n \geqslant 2$.

Первые несколько чисел:

$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233 \ldots$

Задача: найдите $F_n \bmod m$, где n и m до 10^9.

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} F_{n-1} + F_{n-2} \\ F_{n-1} \end{pmatrix}$$

Числа Фибоначчи

Числа Фибоначчи: $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$ при $n \geqslant 2$.

Первые несколько чисел:

$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233 \ldots$

Задача: найдите $F_n \bmod m$, где n и m до 10^9.

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_{n-1} \\ F_{n-2} \end{pmatrix}$$

Числа Фибоначчи

Числа Фибоначчи: $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$ при $n \geqslant 2$.

Первые несколько чисел:

$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233 \ldots$

Задача: найдите $F_n \bmod m$, где n и m до 10^9.

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_{n-2} \\ F_{n-3} \end{pmatrix}$$

Числа Фибоначчи

Числа Фибоначчи: $F_0 = 0$, $F_1 = 1$, $F_n = F_{n-1} + F_{n-2}$ при $n \geqslant 2$.

Первые несколько чисел:

$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233 \ldots$

Задача: найдите $F_n \bmod m$, где n и m до 10^9.

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1} \begin{pmatrix} F_1 \\ F_0 \end{pmatrix}$$

Пример

Рассмотрим такую последовательность: $G_0 = 0$, $G_1 = 1$, $G_2 = 3$, $G_n = 5G_{n-1} - G_{n-3}$ при $n \geqslant 3$.

Задача: найдите $G_n \bmod m$, где n и m до 10^9.

Пример

Рассмотрим такую последовательность: $G_0 = 0$, $G_1 = 1$, $G_2 = 3$, $G_n = 5G_{n-1} - G_{n-3}$ при $n \geqslant 3$.
Задача: найдите $G_n \bmod m$, где n и m до 10^9.

$$G_n = 5G_{n-1} - G_{n-3}$$

Пример

Рассмотрим такую последовательность: $G_0 = 0$, $G_1 = 1$, $G_2 = 3$, $G_n = 5G_{n-1} - G_{n-3}$ при $n \geqslant 3$.
Задача: найдите $G_n \bmod m$, где n и m до 10^9.

$$\begin{cases} G_n &= 5G_{n-1} - G_{n-3} \\ G_{n-1} &= G_{n-1} \\ G_{n-2} &= G_{n-2} \end{cases}$$

Пример

Рассмотрим такую последовательность: $G_0 = 0$, $G_1 = 1$, $G_2 = 3$, $G_n = 5G_{n-1} - G_{n-3}$ при $n \geqslant 3$.
Задача: найдите $G_n \bmod m$, где n и m до 10^9.

$$\begin{pmatrix} G_n \\ G_{n-1} \\ G_{n-2} \end{pmatrix} = \begin{pmatrix} 5 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} G_{n-1} \\ G_{n-2} \\ G_{n-3} \end{pmatrix}$$

Пример

Рассмотрим такую последовательность: $G_0 = 0$, $G_1 = 1$, $G_2 = 3$, $G_n = 5G_{n-1} - G_{n-3}$ при $n \geqslant 3$.
Задача: найдите $G_n \bmod m$, где n и m до 10^9.

$$\begin{pmatrix} G_n \\ G_{n-1} \\ G_{n-2} \end{pmatrix} = \begin{pmatrix} 5 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^{n-2} \begin{pmatrix} G_2 \\ G_1 \\ G_0 \end{pmatrix}$$

Пример с многочленом

Рассмотрим такую последовательность: $H_0 = 4$, $H_1 = -3$, $H_n = 3H_{n-1} - 4H_{n-2} + n^2 + 1$ при $n \geqslant 2$.

Задача: найдите $H_n \bmod m$, где n и m до 10^9.

Пример с многочленом

Рассмотрим такую последовательность: $H_0 = 4$, $H_1 = -3$, $H_n = 3H_{n-1} - 4H_{n-2} + n^2 + 1$ при $n \geqslant 2$.
Задача: найдите $H_n \bmod m$, где n и m до 10^9.

$$H_n = 3H_{n-1} - 4H_{n-2} + n^2 + 1$$

Пример с многочленом

Рассмотрим такую последовательность: $H_0 = 4$, $H_1 = -3$, $H_n = 3H_{n-1} - 4H_{n-2} + n^2 + 1$ при $n \geqslant 2$.
Задача: найдите $H_n \bmod m$, где n и m до 10^9.

$$\begin{cases} H_n &= 3H_{n-1} - 4H_{n-2} + n^2 + 1 \\ H_{n-1} &= H_{n-1} \end{cases}$$

Пример с многочленом

Рассмотрим такую последовательность: $H_0 = 4$, $H_1 = -3$, $H_n = 3H_{n-1} - 4H_{n-2} + n^2 + 1$ при $n \geqslant 2$.
Задача: найдите $H_n \bmod m$, где n и m до 10^9.

$$\begin{cases} H_n & = 3H_{n-1} - 4H_{n-2} + n^2 + 1 \\ H_{n-1} & = H_{n-1} \\ 1 & = 1 \end{cases}$$

Пример с многочленом

Рассмотрим такую последовательность: $H_0 = 4$, $H_1 = -3$, $H_n = 3H_{n-1} - 4H_{n-2} + n^2 + 1$ при $n \geqslant 2$.
Задача: найдите $H_n \bmod m$, где n и m до 10^9.

$$\begin{cases} H_n & = 3H_{n-1} - 4H_{n-2} + n^2 + 1 \\ H_{n-1} & = H_{n-1} \\ n^2 & = (n-1)^2 + 2(n-1) + 1 \\ n & = (n-1) + 1 \\ 1 & = 1 \end{cases}$$

Пример с многочленом

Рассмотрим такую последовательность: $H_0 = 4$, $H_1 = -3$, $H_n = 3H_{n-1} - 4H_{n-2} + n^2 + 1$ при $n \geqslant 2$.
Задача: найдите $H_n \bmod m$, где n и m до 10^9.

$$\begin{pmatrix} H_n \\ H_{n-1} \\ n^2 \\ n \\ 1 \end{pmatrix} = \begin{pmatrix} 3 & -4 & 1 & 2 & 2 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} H_{n-1} \\ H_{n-2} \\ (n-1)^2 \\ n-1 \\ 1 \end{pmatrix}$$

Пример с многочленом

Рассмотрим такую последовательность: $H_0 = 4$, $H_1 = -3$, $H_n = 3H_{n-1} - 4H_{n-2} + n^2 + 1$ при $n \geqslant 2$.
Задача: найдите $H_n \bmod m$, где n и m до 10^9.

$$\begin{pmatrix} H_n \\ H_{n-1} \\ n^2 \\ n \\ 1 \end{pmatrix} = \begin{pmatrix} 3 & -4 & 1 & 2 & 2 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}^{n-1} \begin{pmatrix} H_1 \\ H_0 \\ 1^2 \\ 1 \\ 1 \end{pmatrix}$$

Вопросы?

Вопросы?