Лабораторная работа №4

Wang Yao

RUDN University, Moscow, Russian Federation

3 января 2026

Цель работы

Изучить основные алгоритмы вычисления наибольшего общего делителя (НОД), реализовать классический, бинарный и расширенный алгоритмы Евклида, провести сравнительный анализ их производительности и применимости.

Ход лабораторной работы

Введение в НОД

• **Определение НОД**: Наибольший общий делитель двух или более целых чисел

• **Применения**: - Криптография (RSA, вычисление обратного элемента по модулю) - Сокращение дробей - Решение диофантовых уравнений - Оптимизация алгоритмов в теории чисел

• **Необходимость**: Требуются эффективные алгоритмы для работы с большими числами

Классический алгоритм Евклида

- **Принцип работы**: Основан на делении с остатком

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

- **Пример реализации**:

```
def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return abs(a)
```

3.Бинарный алгоритм Евклида

• Преимущества: Использование побитовых операций вместо деления с остатком

• Основные правила:

Если оба числа четные: $\gcd(a, b) = 2 \cdot \gcd(a/2, b/2)$

Если одно число четное, другое нечетное: $\gcd(a, b) = \gcd(a/2, b)$ или $\gcd(a, b/2)$

Если оба числа нечетные: $\gcd(a, b) = \gcd(|a - b|, \min(a, b))$

• Особенность: Эффективен для чисел с большим количеством нулей в двоичном представлении

Бинарный алгоритм Евклида

• **Преимущества**: Использование побитовых операций вместо деления с остатком

• **Основные правила**: - Если оба числа четные: $\gcd(a, b) = 2 \cdot \gcd(a/2, b/2)$ - Если одно число четное, другое нечетное: $\gcd(a, b) = \gcd(a/2, b)$ или $\gcd(a, b/2)$ - Если оба числа нечетные: $\gcd(a, b) = \gcd(|a - b|, \min(a, b))$

• **Особенность**: Эффективен для чисел с большим количеством нулей в двоичном представлении

Расширенный алгоритм Евклида

- **Цель**: Найти коэффициенты Безу x и y, такие что:

$$\gcd(a, b) = a \cdot x + b \cdot y$$

- **Ключевая идея**: На каждом шаге поддерживать инвариант:

$$\gcd(a, b) = a \cdot x + b \cdot y$$

- **Пример реализации**:

```
def extended_gcd(a, b):
    if b == 0:
        return a, 1, 0
    g, x1, y1 = extended_gcd(b, a % b)
    return g, y1, x1 - (a // b) * y1
```

Вывод

Изучили три основных алгоритма вычисления НОД: классический алгоритм Евклида, бинарный алгоритм Евклида и расширенный алгоритм Евклида.

Основные выводы:

• Теоретическая ценность: Понимание фундаментальных алгоритмов теории чисел

• Практическая значимость: Критически важны для криптографических систем

• Криптографическое применение: Расширенный алгоритм Евклида является основой для вычисления обратных элементов в RSA и других асимметричных криптосистемах

• Итоговый вывод: Каждый алгоритм имеет свою область применения, и выбор конкретного метода зависит от решаемой задачи и характеристик входных данных.

Литература

Теория чисел и криптография. — М.: Физматлит, 2018.