Лабораторная работа №4

Математические основы защиты информации и информационной безопасности

Полиенко Анастасия Николаевна

25 октября 2023

Российский университет дружбы народов, Москва, Россия

НПМмд-02-23

Вычисление наибольшего общего делителя

Цель лабораторной работы

Изучить алгоритмы вычисления наибольшего общего делителя.

Задачи лабораторной работы

Реализовать алгоритм Евклида в четырёх его вариациях:

1. Алгоритм Евклида
2. Бинарный алгоритм Евклида
3. Расширенный алгоритм Евклида
4. Расширенный бинарный алгоритм Евклида

Ход лабораторной работы

Теоретическое введение

Целое число $d \neq 0$ называется *наибольшим общим делителем* целых чисел $a_1, a_2, \dots, a_k$ (обозначается $d = \text{НОД}(a_1, a_2, \dots, a_k)$), если выполняются следующие условия:

1. Каждое из чисел $a_1, a_2, \dots, a_k$ делится на d;
2. Если $d_1 \neq 0$ - другой общий делитель чисел $a_1, a_2, \dots, a_k$, то d_1 делится на d.

Для любых целых чисел $a_1, a_2, \dots, a_k$ существует наибольший общий делитель d и его можно представить в виде *линейной комбинации* этих чисел:

$$d = c_1 a_1 + c_2 a_2 + \cdots + c_k a_k, c_i \in \mathbb{Z}$$

Алгоритм Евклида

```
a = int(input())
b = int(input())
gcd = -1

r_0 = a; r_1 = b
r_2 = -1
while r_2 != 0:
    gcd = r_2
    r_2 = r_0 % r_1
    r_0 = r_1
    r_1 = r_2
print("НОД(", a, ", ", b, ") = ", gcd, sep='')
```

```
48
36
НОД(48, 36) = 12
```

Бинарный алгоритм Евклида

```
a_0 = int(input())
b_0 = int(input())
gcd = -1

a = a_0
b = b_0
g = 1
while (a % 2 == 0) and (b % 2 == 0):
    a = a // 2
    b = b // 2
    g = 2 * g

u = a; v = b
while u != 0:
    while (u % 2 == 0):
        u = u // 2
    while (v % 2 == 0):
        v = v // 2
    if u >= v:
        u = u - v
    else:
        v = v - u

gcd = g*v
print("НОД(", a_0, ", ", b_0, ") = ", gcd, sep='')
```

```
48
36
НОД(48, 36) = 12
```

Расширенный алгоритм Евклида

```
a = int(input())
b = int(input())
gcd = -1
x = -1
y = -1

r_0 = a; r_1 = b;
x_0 = 1; x_1 = 0
y_0 = 0; y_1 = 1
r_2 = -1; q = -1; x_2 = -1; y_2 = -1

while r_2 != 0:
    print(r_2)
    gcd = r_2
    x = x_2
    y = y_2
    r_2 = r_0 % r_1
    q = r_0 // r_1
    x_2 = x_0 - q * x_1
    y_2 = y_0 - q * y_1
    r_0 = r_1
    r_1 = r_2

print("НОД(", a, ", ", b, ") = ", gcd, sep='')
print("x =", x)
print("y =", y)
print("ax + by =", a*x+b*y)
```

```
48
36
-1
12
НОД(48, 36) = 12
x = 1
y = -1
ax + by = 12
```

Расширенный бинарный алгоритм Евклида

```
a_0 = int(input())
b_0 = int(input())
gcd = -1
X = -1
y = -1

a = a_0
b = b_0
g = 1
while (a % 2 == 0) and (b % 2 == 0):
    a = a // 2
    b = b // 2
    g = 2 * g

u = a; v = b
A = 1; B = 0; C = 0; D = 1
while u != 0:
    while (u % 2 == 0):
        u = u // 2
        if (A % 2 == 0) and (B % 2 == 0):
            A = A // 2
            B = B // 2
        else:
            A = (A + b) // 2
            B = (B - a) // 2
    while (v % 2 == 0):
        v = v // 2
        if (C % 2 == 0) and (D % 2 == 0):
            C = C // 2
            D = D // 2
        else:
            C = (C + b) // 2
            D = (D - a) // 2
    if u >= v:
        u = u - v
        A = A - C
        B = B - D
    else:
        v = v - u
        C = C - A
        D = D - B

gcd = g*v
X = C
y = D
print("НОД(", a_0, ", ", b_0, ") = ", gcd, sep='')
print("x =", x)
print("y =", y)
print("ax + by =", a_0*x+b_0*y)
```

```
48
36
НОД(48, 36) = 12
x = 1
y = -1
ax + by = 12
```

Выводы

Изучила вычисление наибольшего общего делителя.