Лабораторная работа №4

Математические основы защиты информации и информационной безопасности

Дэнэилэ Александр Дмитриевич

28 октября 2023

Российский университет дружбы народов, Москва, Россия

НПМмд-02-23

Алгоритмы нахождения наибольшего общего делителя

Цель лабораторной работы

Изучить алгоритмы вычисления наибольшего общего делителя.

Задачи лабораторной работы

1. Реализовать алгоритм Евклида.
2. Реализовать бинарный алгоритм Евклида
3. Реализовать расширенный алгоритм Евклида
4. Реализовать 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

Алгоритм Евклида

```
a = int(input())
b = int(input())
d = None

r0 = a; r1 = b
r2 = None

while r2 != 0:
  d = r2
  r2 = r0 % r1
  r0 = r1
  r1 = r2

print("НОД =", d)
```

```
48
36
НОД = 12
```

Бинарный алгоритм Евклида

```
a = int(input())
b = int(input())
d = None

g = 1

while (a % 2 == 0) & (b % 2 == 0):
  a /= 2; b /= 2; g *= 2

u = a; v = b

while u != 0:
  while u % 2 == 0:
    u /= 2
  while v % 2 == 0:
    v /= 2
  if u >= v:
    u = u - v
  else:
    v = v - u

d = g*v
print("НОД =", d)
```

```
48
36
НОД = 12.0
```

Расширенный алгоритм Евклида

```
a = int(input())
b = int(input())
d = None

r0 = a; r1 = b
r2 = None
x0 = 1; x1 = 0; y0 = 0; y1 = 1
x2 = None; y2 = None

while r2 != 0:
  d = r2
  x = x2
  y = y2
  r2 = r0 % r1
  q = r0 // r1
  r0 = r1
  r1 = r2
  x2 = x0 - q * x1
  y2 = y0 - q * y1

print("НОД =", d)
print(x, y, a*x + b*y)
```

```
48
36
НОД = 12
1 -1 12
```

Расширенный бинарный алгоритм Евклида

```
a = int(input())
b = int(input())
d = None

g = 1

while (a % 2 == 0) & (b % 2 == 0):
  a /= 2; b /= 2; g *= 2

u = a; v = b
A = 1; B = 0; C = 0; D = 1

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

d = g*v
x = C
y = D
print("НОД =", d)
print(x, y, d)
```

```
48
36
НОД = 12.0
1.0 -1.0 12.0
```

Выводы

Ознакомился с различными вариациями реализации алгоритмов нахождения наименьшего общего делителя.