Лабораторная работа №4

Математические основы защиты информации и информационной безопасности

Леонтьева К. А., НПМмд-02-23

14 октября 2023

Российский университет дружбы народов

Москва, Россия

Цель лабораторной работы

1) Реализовать на языке программирования алгоритмы Евклида для вычисления наибольшего общего делителя

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

- каждое из чисел $a_1, a_2, ..., a_k$ делится на d,
- если $d_1 \neq 0$ - другой общий делитель чисел $a_1, a_2, ..., a_k$, то d делится на d_1.

Для вычисления наибольшего общего делителя двух целых чисел применяется способ повторного деления с остатком, называемый **алгоритмом Евклида**.

Бинарный алгоритм Евклида основан на следующих свойствах наибольшего общего делителя (считаем, что $0 < b \leq a$):

1. если оба числа a и b четные, то НОД$(a, b) = 2*$ НОД$(\frac{a}{2}, \frac{b}{2})$
2. если число a - нечетное, число b -четное, то НОД$(a, b) =$ НОД$(a, \frac{b}{2})$
3. если оба числа a и b нечетные, $a > b$, то НОД$(a, b) =$ НОД$(a - b, b)$
4. если $a = b$, то НОД$(a, b) = a$

Ход выполнения лабораторной работы

- Реализуем алгоритм Евклида

```
a = 54321
b = 12345
r = -1

if a >= b:
    r0 = a
    r1 = b
else:
    r0 = b
    r1 = a

while r != 0:
    r = r0 % r1
    r0 = r1
    r1 = r

print('НОД (',a,',',b, ') =', r0)
```

```
НОД ( 54321 , 12345 ) = 3
```

Figure 1: Рис.1: Алгоритм Евклида

Ход выполнения лабораторной работы

- Реализуем бинарный алгоритм Евклида

```
aa = 3400
bb = 1260
g = 1

if aa >= bb:
    a = aa
    b = bb
else:
    a = bb
    b = aa

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

print('НОД (',aa,',',bb, ') =', g*v)
```

```
НОД ( 3400 , 1260 ) = 20.0
```

Figure 2: Рис.2: Бинарный алгоритм Евклида

Ход выполнения лабораторной работы

- Реализуем расширенный алгоритм Евклида

```
a = 3400
b = 1260
r = -1

if a >= b:
    r0 = a
    r1 = b
else:
    r0 = b
    r1 = a

x0 = 1
x1 = 0
y0 = 0
y1 = 1

while r != 0:
    r = r0 % r1
    q = int((r0 - r) / r1)
    r0 = r1
    r1 = r
    x_new = x0 - q * x1
    x0 = x1
    x1 = x_new
    y_new = y0 - q * y1
    y0 = y1
    y1 = y_new

print('НОД (',a,',',b, ') =', r0)
print('x =', x0)
print('y =', y0)
```

```
НОД ( 3400 , 1260 ) = 20
x = -10
y = 27
```

Figure 3: Рис.2: Расширенный алгоритм Евклида

Ход выполнения лабораторной работы

- Реализуем расширенный бинарный алгоритм Евклида

```
aa = 3400
bb = 1260
g = 1

if aa >= bb:
    a = aa
    b = bb
else:
    a = bb
    b = aa

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

print('НОД (',aa,',',bb, ') =', g*v)
print('x =', C)
print('y =', D)
```

```
НОД ( 3400 , 1260 ) = 20.0
x = -10.0
y = 27.0
```

Figure 4: Рис.2: Расширенный бинарный алгоритм Евклида

Вывод

- В ходе выполнения данной лабораторной работы были реализованы алгоритмы Евклида для вычисления наибольшего общего делителя