Лабораторная работа №4

Математические основы защиты информации и информационной безопасности

Леонтьева Ксения Андреевна | НПМмд-02-23

Содержание

Список иллюстраций

1 Цель работы

Реализовать на языке программирования алгоритмы Евклида для вычисления наибольшего общего делителя.

2 Теоретическое введение

Целое число $d \neq 0$ называется **наибольшим общим делителем** целых чисел $a_1, a_2, ..., a_k$ (обозначается $d =$ 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.

Ненулевые целые числа a и b называются **ассоциированными**, если a делится на b и b делится на a.

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

$$d = c_1a_1 + c_2a_2 + ... + c_ka_k, c_i \in Z.$$

Целые числа $a_1, a_2, ..., a_k$ называются **взаимно простыми в совокупности**, если НОД$(a_1, a_2, ..., a_k) = 1$. Целые числа a и b называются **взаимно простыми**, если НОД$(a, b) = 1$.

Целые числа $a_1, a_2, ..., a_k$ называются **попарно взаимно простыми**, если НОД$(a_i, a_j) = 1$ для всех $1 \leq i \neq j \leq k$.

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$(a, b) = a$

Более подробно см. в [1].

3 Выполнение лабораторной работы

Алгоритм Евклида реализуем по следующей схеме:
На вход подаются целые числа a и b: $0 < b \leq a$.

1. Положить $r_0 \leftarrow a, r_1 \leftarrow b, i \leftarrow 1$
2. Найти остаток r_{i+1} от деления r_{i-1} на r_i
3. Если $r_{i+1} = 0$, то положить $d \leftarrow r_i$. В противном случае положить $i \leftarrow i+1$ и вернуться на шаг 2
4. Результат d

Код программы (рис. 3.1).

```
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
```

Рис. 3.1: Алгоритм Евклида

Бинарный алгоритм Евклида реализуем по следующей схеме:
На вход подаются целые числа a и b: $0 < b \leq a$.

1. Положить $g \leftarrow 1$
2. Пока оба числа a и b четные, выполнять $a \leftarrow \frac{a}{2}, b \leftarrow \frac{b}{2}, g \leftarrow 2g$ до получения хотя бы одного нечетного значения a или b
3. Положить $u \leftarrow a, v \leftarrow b$
4. Пока $u \neq 0$ выполнять следующие действия:

   4.1. Пока u четное, полагать $u \leftarrow \frac{u}{2}$

   4.2. Пока v четное, полагать $v \leftarrow \frac{v}{2}$

   4.3. При $u \geq v$ положить $u \leftarrow u - v$. В противном случае положить $v \leftarrow v - u$
5. Положить $d \leftarrow gv$
6. Результат d

Код программы (рис. 3.2).

```
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
```

Рис. 3.2: Бинарный алгоритм Евклида

Расширенный алгоритм Евклида реализуем по следующей схеме:
На вход подаются целые числа a и b: $0 < b \leq a$.

1. Положить $r_0 \leftarrow a, r_1 \leftarrow b, x_0 \leftarrow 1, x_1 \leftarrow 0, y_0 \leftarrow 0, y_1 \leftarrow 1, i \leftarrow 1$
2. Разделить с остатком r_{i-1} на r_i: $r_{i-1} = q_i r_i + r_{i+1}$
3. Если $r_{i+1} = 0$, то положить $d \leftarrow r_i, x \leftarrow x_i, y \leftarrow y_i$. В противном случае положить $x_{i+1} \leftarrow x_{i-1} - g_i x_i, y_{i+1} \leftarrow y_{i-1} - g_i y_i, i \leftarrow i + 1$ и вернуться на шаг 2
4. Результат d, x, y

Код программы (рис. 3.3).

```
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
```

Рис. 3.3: Расширенный алгоритм Евклида

Расширенный бинарный алгоритм Евклида реализуем по следующей схеме:
На вход подаются целые числа a и b: $0 < b \leq a$.

1. Положить $g \leftarrow 1$

2. Пока числа a и b четные, выполнять $a \leftarrow \frac{a}{2}, b \leftarrow \frac{b}{2}, g \leftarrow 2g$ до получения хотя бы одного нечетного значения a или b

3. Положить $u \leftarrow a, v \leftarrow b, A \leftarrow 1, B \leftarrow 0, C \leftarrow 0, D \leftarrow 1$

4. Пока $u \neq 0$ выполнять следующие действия:

   4.1. Пока u четное:

   4.1.1. Положить $u \leftarrow \frac{u}{2}$

   4.1.2. Если оба числа A и B четные, то положить $A \leftarrow \frac{A}{2}, B \leftarrow \frac{B}{2}$. В противном случае положить $A \leftarrow \frac{A+b}{2}, B \leftarrow \frac{B-a}{2}$

   4.2. Пока v четное:

   4.2.1. Положить $v \leftarrow \frac{v}{2}$

   4.2.2. Если оба числа C и D четные, то положить $C \leftarrow \frac{C}{2}, D \leftarrow \frac{D}{2}$. В противном случае положить $C \leftarrow \frac{C+b}{2}, D \leftarrow \frac{D-a}{2}$

   4.3. При $u \geq v$ положить $u \leftarrow u-v, A \leftarrow A-C, B \leftarrow B-D$. В противном случае положить $v \leftarrow v - u, C \leftarrow C - A, D \leftarrow D - B$

5. Положить $d \leftarrow gv, x \leftarrow C, y \leftarrow D$

6. Результат d, x, y

Код программы (рис. 3.4).

```
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
```

Рис. 3.4: Расширенный бинарный алгоритм Евклида

4 Выводы

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

Список литературы

1. Наибольший общий делитель [Электронный ресурс]. URL: http://www.cleverstudents.ru/divisibility/nod.html.