

Математические основы защиты информации и информационной безопасности

Отчет по лабораторной работе № 4

Меньшов Иван Сергеевич НПМмд-02-21

Содержание

1	Цель работы	4
2	Теоретические сведения	5
2.1	Алгоритм Евклида	5
2.2	Бинарный алгоритм Евклида	5
2.3	Расширенный алгоритм Евклида и Бинарный алгоритм Евклида .	6
3	Выполнение работы	7
3.1	Реализация алгоритмов на языке Python	7
3.2	Контрольный пример	11
4	Выводы	12
	Список литературы	13

List of Figures

3.1	Алгоритм Евклида	11
3.2	Бинарный алгоритм Евклида	11
3.3	Расширенный алгоритм Евклида	11
3.4	Расширенный бинарный алгоритм Евклида	11

1 Цель работы

Изучить алгоритмы для вычисления наибольшего общего делителя.

2 Теоретические сведения

2.1 Алгоритм Евклида

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

Алгоритм Евклида.

Вход. Целые числа a, b ; $0 < b < a$.

Выход. $d = \text{НОД}(a, b)$.

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

2.2 Бинарный алгоритм Евклида

Бинарный алгоритм Евклида является более быстрым при реализации на компьютере, поскольку использует двоичное представление чисел a и b . Бинарный алгоритм Евклида основан на следующих свойствах наибольшего общего делителя (считаем, что $0 < b < a$):

1. если оба числа a и b четные, то $\text{НОД}(a, b) = 2 \text{НОД}(a/2, b/2)$

2. если число a - нечетное, число b — четное, то $\text{НОД}(a, b) = \text{НОД}(a, b/2)$
3. если оба числа a и b нечетные, $a > b$, то $\text{НОД}(a, b) = \text{НОД}(a - b, b)$
4. если $a = b$, то $\text{НОД}(a, b) = a$

2.3 Расширенный алгоритм Евклида и Бинарный алгоритм Евклида

Данные алгоритмы Евклида находят, помимо $g = \text{НОД}(a, b)$ такие целые коэффициенты x и y , что:

$$ax + by = d$$

Заметим, что решений бесконечно много: имея решение (x, y) , можно x увеличить на b , а y уменьшить на a , и равенство при этом не изменится. С полным алгоритмом вы можете ознакомиться в инструкции к лабораторной работе №4

3 Выполнение работы

3.1 Реализация алгоритмов на языке Python

Алгоритм Евклида

```
print('a >= b > 0')
a = int(input('Введите a: '))
b = int(input('Введите b: '))
r = [a, b]
while True:
    if r[-2] % r[-1] == 0:
        d = r[-1]
        break
    else:
        r.append(r[-2] % r[-1])
print(f'НОД(a,b) = {d}')
```

Бинарный алгоритм Евклида

```
print('a >= b > 0')
a = int(input('Введите a: '))
b = int(input('Введите b: '))
g = 1
while True:
    if a % 2 == 0 and b % 2 == 0:
```

```

        a *= 0.5
        b *= 0.5
        g *= 2
    else:
        break
u = a
v = b
while u != 0:
    while u % 2 == 0:
        u *= 0.5
    while v % 2 == 0:
        v *= 0.5
    if u >= v:
        u = u-v
    else:
        v = v-u
d = g*v
print(f'НОД(a,b) = {d}')

```

Расширенный алгоритм Евклида

```

print('a >= b > 0')
a = int(input('Введите a: '))
b = int(input('Введите b: '))
r = [a, b]
x = [1, 0]
y = [0, 1]
while True:
    if r[-2] % r[-1] == 0:
        d = r[-1]

```



```

        X = x[-1]
        Y = y[-1]
        break
    else:
        q_i = r[-2] // r[-1]
        x.append(x[-2]-q_i*x[-1])
        y.append(y[-2]-q_i*y[-1])
        r.append(r[-2] % r[-1])
print(d, X, Y)

```

Расширенный бинарный алгоритм Евклида

```

print('a >= b > 0')
a = int(input('Введите a: '))
b = int(input('Введите b: '))
g = 1
while True:
    if a % 2 == 0 and b % 2 == 0:
        a *= 0.5
        b *= 0.5
        g *= 2
    else:
        break
u = a
v = b
A = 1
B = 0
C = 0
D = 1
while u != 0:

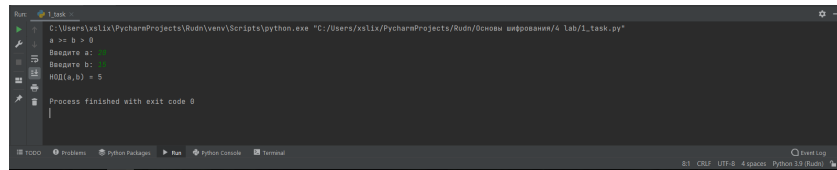
```

```

while u % 2 == 0:
    u = u * 0.5
    if A % 2 == 0 and B % 2 == 0:
        A *= 0.5
        B *= 0.5
    else:
        A = (A + b) * 0.5
        B = (B - a) * 0.5
while v % 2 == 0:
    v = v * 0.5
    if C % 2 == 0 and D % 2 == 0:
        C *= 0.5
        D *= 0.5
    else:
        C = (C + b) * 0.5
        D = (D - a) * 0.5
if u >= v:
    u = u - v
    A = A - C
    B = B - D
else:
    v = v - u
    C = C - A
    D = D - B
d = g * v
x = C
y = D
print(d, x, y)

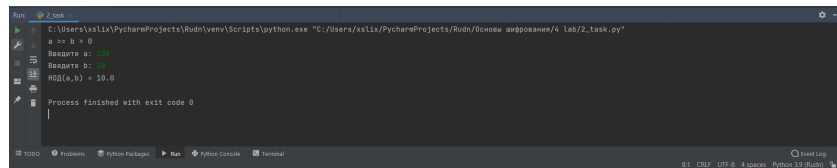
```

3.2 Контрольный пример



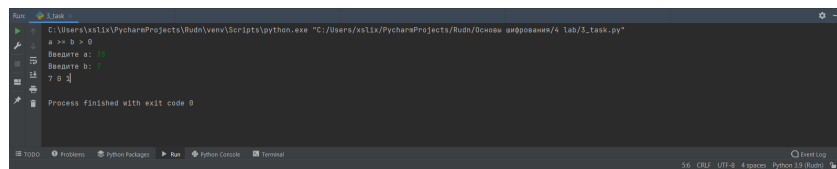
```
Run 1.task
C:\Users\xslix\PycharmProjects\Rudn\venv\Scripts\python.exe "C:/Users/xslix/PycharmProjects/Rudn/Основа_записки/a_lab/1_task.py"
a = 20
b = 5
HOD(a, b) = 5
Process finished with exit code 0
```

Figure 3.1: Алгоритм Евклида



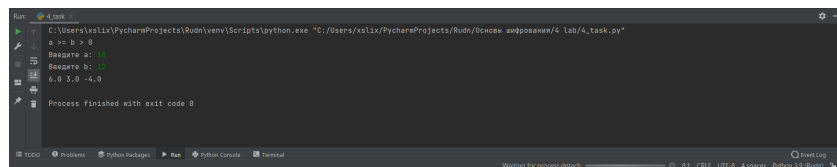
```
Run 2.task
C:\Users\xslix\PycharmProjects\Rudn\venv\Scripts\python.exe "C:/Users/xslix/PycharmProjects/Rudn/Основа_записки/a_lab/2_task.py"
a = 20
b = 10
HOD(a, b) = 10.0
Process finished with exit code 0
```

Figure 3.2: Бинарный алгоритм Евклида



```
Run 3.task
C:\Users\xslix\PycharmProjects\Rudn\venv\Scripts\python.exe "C:/Users/xslix/PycharmProjects/Rudn/Основа_записки/a_lab/3_task.py"
a = 20
b = 10
HOD(a, b) = 10.0
Process finished with exit code 0
```

Figure 3.3: Расширенный алгоритм Евклида



```
Run 4.task
C:\Users\xslix\PycharmProjects\Rudn\venv\Scripts\python.exe "C:/Users/xslix/PycharmProjects/Rudn/Основа_записки/a_lab/4_task.py"
a = 20
b = 10
HOD(a, b) = 10.0
Process finished with exit code 0
```

Figure 3.4: Расширенный бинарный алгоритм Евклида

4 Выводы

Мной были изучены алгоритмы для вычисления наибольшего общего делителя Евклида.

Список литературы

1. Алгоритмы Евклида
2. Инструкция к лабораторной работе №4