

# **Отчет по лабораторной работе №4**

**Вычисление наибольшего общего делителя**

Бармина Ольга Константиновна

2024 September 7th

# Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выводы	12
5	Список литературы	13

# List of Figures

3.1	Алгоритм Евклида . . . . .	7
3.2	Бинарный алгоритм Евклида . . . . .	8
3.3	Расширенный алгоритм Евклида . . . . .	9
3.4	Расширенный бинарный алгоритм Евклида 1 . . . . .	10
3.5	Расширенный бинарный алгоритм Евклида 2 . . . . .	11

# 1 Цель работы

Целью данной работы является освоение алгоритмов вычисления наибольшего общего делителя.

## 2 Задание

1. Изучить методы вычисления наибольшего общего делителя.
2. Реализовать алгоритмы вычисления НОД.

### 3 Теоретическое введение

Пусть числа  $a$  и  $b$  целые и  $b \neq 0$ . Разделить  $a$  на  $b$  с остатком - значит представить  $a$  в виде  $a = qb + r$ , где  $q, r \in \mathbb{Z}$  и  $0 \leq r \leq |b|$ . Число  $q$  называется неполным частным, число  $r$  - неполным остатком от деления  $a$  на  $b$ .

Целое число  $d \neq 0$  называется *наибольшим общим делителем* целых чисел  $a_1, a_2, \dots, a_k$  (обозначается  $d = (a_1, a_2, \dots, a_k)$ ), 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  $d$  делится на  $d_1$ . [1] # Ход выполнения лабораторной работы

Работа выполняется на языке программирования Python с использованием среды Google Colab

1. Реализуем алгоритм Евклида:

```
def euclid(a,b):  
    while a!=0 and b!=0:  
        if a>=b:  
            a = a%b  
        else:  
            b = b%a  
    return a or b
```

```
a = 12345  
b = 54321  
euclid(a,b)
```

3

Figure 3.1: Алгоритм Евклида

2. Реализуем бинарный алгоритм Евклида:

```

def bin_euclid(a,b):
    g = 1
    while a%2 == 0 and b%2 == 0:
        a /= 2
        b /= 2
        g *= 2
    u = a
    v = b
    while u != 0:
        if u%2 == 0:
            u /= 2
        if v%2 == 0:
            v /= 2
        if u >= v:
            u -= v
        else:
            v -= u
    return g*v

```

```
bin_euclid(a,b)
```

3.0

Figure 3.2: Бинарный алгоритм Евклида

3. Реализуем расширенный алгоритм Евклида:



```
def ext_euclid(a, b):  
    if a == 0:  
        return(b, 0, 1)  
    else:  
        d, x, y = ext_euclid(b%a,a)  
        return d, x-(b//a)*y, y
```

```
ext_euclid(a,b)
```

```
(3, -177, 1)
```

Figure 3.3: Расширенный алгоритм Евклида

5. Реализуем расширенный бинарный алгоритм Евклида:

```

def ext_bin_euclid(a1,b1):
    g = 1
    while a1%2 == 0 and b1%2 == 0:
        a1 /= 2
        b1 /= 2
        g *= 2
    u, v = a1, b1
    a, b, c, d = 1, 0, 0, 1
    while u != 0:
        while u%2 == 0:
            u /= 2
            if a%2 == 0 and b%2 == 0:
                a /= 2
                b /= 2
            else:
                a = (a + b1)//2
                b = (b - a1)//2
        while v%2 == 0:
            v /= 2
            if c%2 == 0 and d%2 == 0:
                c /= 2
                d /= 2
            else:
                c = (c + b1)//2
                d = (d - a1)//2

```

Figure 3.4: Расширенный бинарный алгоритм Евклида 1

```

def ext_bin_euclid(a1,b1):
    g = 1
    while a1%2 == 0 and b1%2 == 0:
        a1 /= 2
        b1 /= 2
        g *= 2
    u, v = a1, b1
    a, b, c, d = 1, 0, 0, 1
    while u != 0:
        while u%2 == 0:
            u /= 2
            if a%2 == 0 and b%2 == 0:
                a /= 2
                b /= 2
            else:
                a = (a + b1)//2
                b = (b - a1)//2
        while v%2 == 0:
            v /= 2
            if c%2 == 0 and d%2 == 0:
                c /= 2
                d /= 2
            else:
                c = (c + b1)//2
                d = (d - a1)//2

```

Figure 3.5: Расширенный бинарный алгоритм Евклида 2

## 4 Выводы

В ходе работы мы изучили и реализовали алгоритмы вычисления наибольшего общего делителя.

## **5 Список литературы**

1. Методические материалы курса