

# **Отчет по лабораторной работе №4**

**Вычисление наибольшего общего делителя**

Асеинова Елизавета Валерьевна

28 октября 2023

# Содержание

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

## Список иллюстраций

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  $0 \leq r < |b|$ . 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:

```
✓ [1] a = 24690
0 сек. b = 12345

✓ [2] def alg_e(a, b):
0 сек.     while (a != 0) and (b != 0):
            if a >= b:
                a = a % b
            else:
                b = b % a
            return a or b

✓ [3] alg_e(a, b)
0 сек.
12345
```

Рис. 3.1.: Алгоритм Евклида

2. Реализуем бинарный алгоритм Евклида:

✓  
0  
сек.



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

✓  
0  
сек.

```
[5] alg_e_bin(a, b)
```

12345

Рис. 3.2.: Бинарный алгоритм Евклида

3. Реализуем расширенный алгоритм Евклида:



```
✓ [6] def alg_e_ext(a, b):  
0   if a == 0:  
сек.   return(b, 0, 1)  
       else:  
         d, y, x = alg_e_ext(b % a, a)  
       return (d, x-(b//a)*y, y)  
  
✓ [7] alg_e_ext(a, b)  
0  
сек.  
  
      (12345, 0, 1)
```

Рис. 3.3.: Расширенный алгоритм Евклида

5. Реализуем расширенный бинарный алгоритм Евклида:

```

def alg_e_bin_ext(a, b):
    g = 1
    while (a % 2 == 0) and (b % 2 == 0):
        a /= 2
        b /= 2
        g *= 2
    u, v = a, b
    A, B, C, D = 1, 0, 0, 1

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

```


Рис. 3.4.: Расширенный бинарный алгоритм Евклида 1

```
    D = (D - a) // 2

    if u >= v:
        u -= v
        A -= C
        B -= D
    else:
        v -= u
        C -= A
        D -= B

    d = g*v
    x = C
    y = D
    return(d, x, y)
```

✓  
0  
сек.

 alg\_e\_bin\_ext(a, b)

(12345, 0, 1)

Рис. 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