

# Лабораторная работа №4

---

Морозова Ульяна

Российский университет дружбы народов, Москва, Россия

Целью работы является изучение алгоритмов вычисления НОД и реализация их на языке Julia.

Алгоритм Евклида — метод для нахождения наибольшего общего делителя (НОД) двух чисел. Алгоритм основан на принципе, что НОД двух чисел остаётся неизменным, когда большее число заменяется его остатком при делении на меньшее число.

```
function euclidean(a::T, b::T) where T<:Integer
    a=abs(a)
    b=abs(b)

    while b != 0
        a,b = b, a%b
    end
    return a
end
```

Бинарный алгоритм вычисления НОД (бинарный алгоритм Евклида) — метод нахождения наибольшего общего делителя (НОД) двух целых чисел, который использует операции сдвига и вычитания вместо деления. Разработан в 1967 году Джозефом Стайном.

Преимущества алгоритма: - скорость работы за счёт использования бинарных операций, которые выполняются быстрее, чем деление и умножение; - отсутствие дорогостоящих операций деления и остатка.

```
function binary_euc(a::T, b::T) where T<:Integer
    a=abs(a)
    b=abs(b)

    if a == 0
        return b
    end
    if b == 0
        return a
    end

    s = 0
    while ((a|b)&1) == 0
        a>>=1
```

Расширенный алгоритм Евклида — модификация алгоритма Евклида, которая, кроме вычисления наибольшего общего делителя (НОД) двух целых чисел, находит коэффициенты  $x$  и  $y$  для уравнения Безу:  $ax + by = \text{НОД}(a, b)$ . Это означает, что алгоритм не только находит НОД, но и выражает его как линейную комбинацию  $a$  и  $b$ .

```
function ext_euc(a::T, b::T) where T<:Integer
    if b == 0
        return a, one(T), zero(T)
    end

    gcd_val, x1, y1 = ext_euc(b, a % b)
    x = y1
    y = x1 - (a ÷ b) * y1

    return gcd_val, x, y
end
```



Бинарный расширенный алгоритм Евклида - это комбинация бинарного и расширенного алгоритмов, он использует битовые операции для повышения эффективности. Все функции работают с целыми числами любого размера и корректно обрабатывают отрицательные числа и нули.

Тестирование алгоритмов НОД:

=====

НОД(48, 18) = 6

Классический: 6

Бинарный: 6

Расширенный: 6 (коэффициенты: -1, 3)

Бинарный расширенный: 6 (коэффициенты: -4, 11)

Тождество Безу (расширенный): true

Тождество Безу (бинарный расширенный): true

-----

НОД(1071, 462) = 21

Классический: 21

Бинарный: 21

Расширенный: 21 (коэффициенты: -3, 7)

## Выводы

---

Мы изучили работу алгоритмов вычисления НОД, а также реализовали их на языке Julia.