

# Лабораторна работа № 4 - Алгоритмы нахождения НОД

---

Покрас Илья Михайлович НФИмд-01-24

25 октября, 2024, Москва, Россия

Российский Университет Дружбы Народов Имени Патриса Лумумбы

Реализовать алгоритмы нахождения наибольшего общего делителя (НОД)

- Реализовать алгоритм Евклида;
- Реализовать расширенный алгоритм Евклида;
- Реализовать бинарный алгоритм Евклида;
- Реализовать бинарный расширенный алгоритм Евклида;

```
function gcd_algorithm(a, b)
    if a == 0
        return b
    end
    return gcd_algorithm(b % a, a)
end
```

Рис. 1: Функция алгоритма Евклида

## Расширенный алгоритм Евклида

```
function extended_gcd_algorithm(a, b)
    if a == 0
        return (b, 0, 1)
    else
        d, x, y = extended_gcd_algorithm(b % a, a)
    end
    return (d, y - (b ÷ a) * x, x)
end
```

Рис. 2: Функция расширенного алгоритма Евклида

# Инициализация и вызов функций алгоритмов

```
a = 6  
b = 4  
println(gcd_algorithm(a, b))  
extended_gcd_algorithm(a, b)
```

**Рис. 3:** Инициализация переменных и вызов функций(1)

```
2  
  
(2, 1, -1)
```

**Рис. 4:** Результат выполнения(1)

# Бинарный алгоритм Евклида

```
function gcd_binary(a, b)
    g = 1

    while a % 2 == 0 && b % 2 == 0
        a = a ÷ 2
        b = b ÷ 2
        g = 2 * g
    end

    while a % 2 == 0
        a = a ÷ 2
    end

    while b % 2 == 0
        b = b ÷ 2
    end

    t = abs(a - b) ÷ 2

    if a >= b
        a = t
    else
        b = t
    end

    return g * b
end
```

Рис. 5: Функция бинарного алгоритма Евклида

# Расширенный бинарный алгоритм Евклида

```
function extended_gcd_binary(a, b)
    g = 1

    while a % 2 == 0 && b % 2 == 0
        a = a / 2
        b = b / 2
        g = 2 * g
    end

    u = a
    v = b
    A = 1
    B = 0
    C = 0
    D = 1

    while u % 2 == 0
        u = u / 2
        if A % 2 == 0 && B % 2 == 0
            A = A / 2
            B = B / 2
        else
            A = (A + b) / 2
            B = (B - a) / 2
        end
    end

    while u < v
        if v % 2 == 0
            v = v / 2
            if C % 2 == 0 && D % 2 == 0
                C = C / 2
                D = D / 2
            else
                C = (C + b) / 2
                D = (D - a) / 2
            end
        elseif u > v
            u, v, A, B, C, D = v, u, C, D, A, B
        else
            v = v - u
            C = C - A
            D = D - B
        end
    end

    return (g * v, C, D)
end
```

Рис. 6: Функция расширенного алгоритма Евклида



## Инициализация и вызов функций алгоритмов

```
a = 6  
b = 4  
  
println(gcd_binary(a, b))  
println(extended_gcd_binary(a, b))
```

Рис. 7: Инициализация переменных и вызов функций(2)

```
2  
  
(2, 1, -1)
```

Рис. 8: Результат выполнения(2)

Я реализовал алгоритмы нахождения НОД.

**Спасибо за внимание**

---