

Защита лабораторной работы №4

Алгоритмы вычисления наибольшего общего делителя

Бармина Ольга

10 Октября 2024

Российский университет дружбы народов, Москва, Россия

- Освоение алгоритмов вычисления наибольшего общего делителя
- Программная реализация алгоритмов вычисления НОД

Для любых целых чисел a_1, a_2, \dots, a_k существует наибольший общий делитель d и его можно представить в виде *линейной комбинации* этих чисел:

$$d = c_1 a_1 + c_2 a_2 + \dots + c_k a_k$$

Реализовать алгоритмы вычисления наибольшего общего делителя, такие как: - Алгоритм Евклида - Бинарный алгоритм Евклида - Расширенный алгоритм Евклида - Расширенный бинарный алгоритм Евклида

Исходные данные: - $a = 12345$ - $b = 54321$

Результат выполнения лабораторной работы

Вычисление НОД при помощи алгоритма Евклида:

```
function euclid(a, b)
    while a != 0 && b != 0
        if a >= b
            a = a % b
        else
            b = b % a
        end
    end
    return a != 0 ? a : b
end
```

euclid (generic function with 1 method)

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

3

Рис. 1: Алгоритм Евклида

Вычисление НОД при помощи бинарного алгоритма Евклида:

```
function bin_euclid(a, b)
    g = 1
    while a % 2 == 0 && b % 2 == 0
        a /= 2
        b /= 2
        g *= 2
    end
    u = a
    v = b
    while u != 0
        if u % 2 == 0
            u /= 2
        end
        if v % 2 == 0
            v /= 2
        end
        if u >= v
            u -= v
        else
            v -= u
        end
    end
    return g * v
end

bin_euclid (generic function with 1 method)

bin_euclid(35,1359331)

1.0
```

Рис. 2: Бинарный алгоритм Евклида

Результат выполнения лабораторной работы

Вычисление НОД при помощи расширенного алгоритма Евклида:

```
function ext_euclid(a, b)
    r = Int[]
    x = Int[]
    y = Int[]
    d = 0
    push!(r, a)
    push!(r, b)
    push!(x, 1)
    push!(x, 0)
    push!(y, 0)
    push!(y, 1)
    i = 2
    while r[i] != 0
        i += 1
        push!(r, r[i-2] % r[i-1])
        if r[i] == 0
            d = r[i-1]
            x_val = x[i-1]
            y_val = y[i-1]
        else
            push!(x, x[i-2] - (div(r[i-2], r[i-1]) * x[i-1]))
            push!(y, y[i-2] - (div(r[i-2], r[i-1]) * y[i-1]))
        end
    end
    return d, x, y
end
```

ext_euclid (generic function with 1 method)

ext_euclid(a,b)

(3, [1, 0, 1, -4, 9, -22, 3617], [0, 1, 0, 1, -2, 5, -822])

Результат выполнения лабораторной работы

Вычисление НОД при помощи расширенного бинарного алгоритма Евклида:

```
function ext_bin_euclid(a1, b1)
    g = 1
    while a1 % 2 == 0 && b1 % 2 == 0
        a1 /= 2
        b1 /= 2
        g *= 2
    end
    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 && b % 2 == 0
                a /= 2
                b /= 2
            else
                a = (a + b1) + 2
                b = (b - a1) + 2
            end
        end
        while v % 2 == 0
            v /= 2
            if c % 2 == 0 && d % 2 == 0
                c /= 2
                d /= 2
            else
                c = (c + b1) + 2
                d = (d - a1) + 2
            end
        end
        if u > v
            u = u - v
            a = a - c
            b = b - d
        else
            v = v - u
            c = c - a
            d = d - b
        end
    end
    g
end
```

Рис. 4: Расширенный бинарный алгоритм Евклида

Вычисление НОД при помощи расширенного бинарного алгоритма Евклида:

```
if u >= v
    u -= v
    a -= c
    b -= d
else
    v -= u
    c -= a
    d -= b
end
end
d = g * v
x = c
y = d
return d, x, y
end

ext_bin_euclid (generic function with 1 method)

ext_bin_euclid(a,b)

(3.0, -14490.0, 3.0)
```

Рис. 5: Расширенный бинарный алгоритм Евклида 2

Выводы

1. Изучили алгоритмы вычисления наибольшего общего делителя
2. Реализовали алгоритмы вычисления НОД