Лабораторна работа № 4

Вычисление наибольшего общего делителя

Покрас Илья Михайлович

Содержание

Список иллюстраций

1 Цель работы

Реализовать алгоритмы нахождения наибольшего общего делителя (НОД)

2 Задание

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

3 Теоретическое введение

Алгори́тм Евклида — эффективный алгоритм для нахождения наибольшего общего делителя двух целых чисел (или общей меры двух отрезков). Алгоритм назван в честь греческого математика Евклида (III век до н. э.). Это один из старейших численных алгоритмов, используемых в наше время. В самом простом случае алгоритм Евклида применяется к паре положительных целых чисел и формирует новую пару, которая состоит из меньшего числа и остатка от деления большего числа на меньшее. Процесс повторяется, пока числа не станут равными. Найденное число и есть наибольший общий делитель исходной пары.

Бинарный алгоритм Евклида — метод нахождения наибольшего общего делителя двух целых чисел. Данный алгоритм «быстрее» обычного алгоритма Евклида, так как вместо медленных операций деления и умножения используются сдвиги.

Расширенные алгоритмы Евклида — модификации алгоритмы Евклида, вычисляющая, кроме наибольшего общего делителя (НОД) целых чисел a и b, ещё и коэффициенты соотношения Безу, то есть такие целые x и y, что ax+by=НОД(a,b)

4 Выполнение лабораторной работы

4.1 Алгоритм Евклида

Я создал функцию алгоритма Евклида, принимающую числа a и b, находящую через рекурсию НОД и возвращающую его (рис. 4.1).

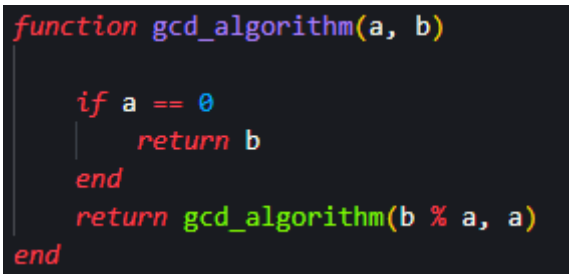


Рис. 4.1: Функция алгоритма Евклида

4.2 Расширенный алгоритм Евклида

Далее я создал расширенный алгоритм Евклида для нахождения и возврата НОД и коэффициентов соотношения Безу x и y (рис. 4.2).

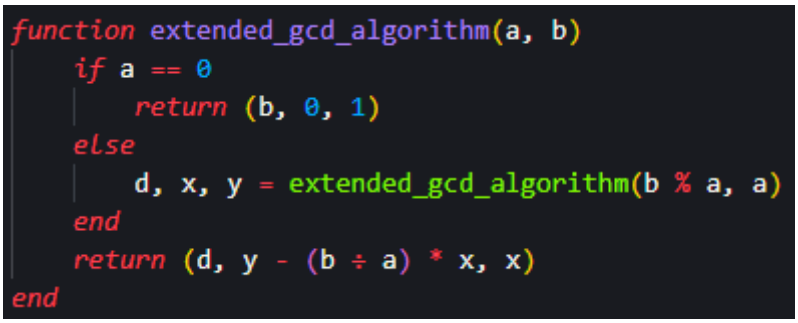


Рис. 4.2: Функция расширенного алгоритма Евклида

Я создал блок инициализации переменных a и b и вызов функций с входными переменными (рис. 4.3).

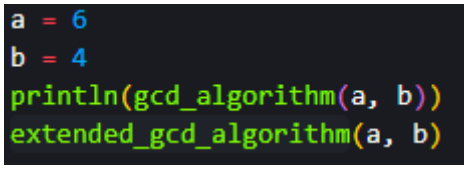


Рис. 4.3: Инициализация переменных и вызов функций(1)

И получил следующие значения (рис. 4.4).

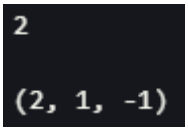
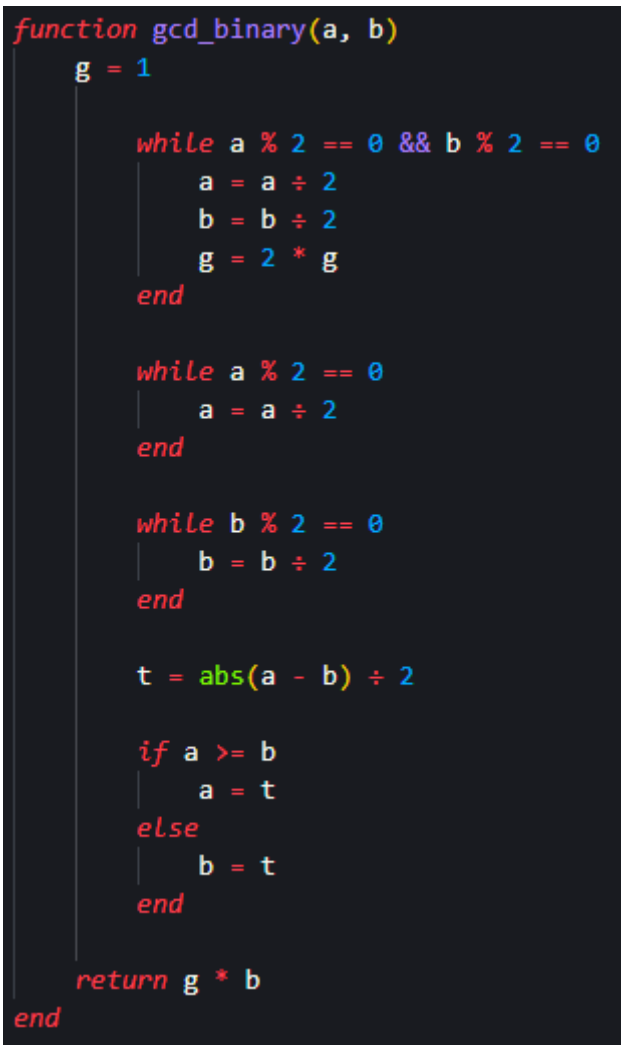


Рис. 4.4: Результат выполнения(1)

4.3 Бинарный алгоритм Евклида

Я создал функцию бинарного алгоритма Евклида, принимающую числа a и b, находящую с помощью сдвигов и возращающую НОД (рис. 4.5).

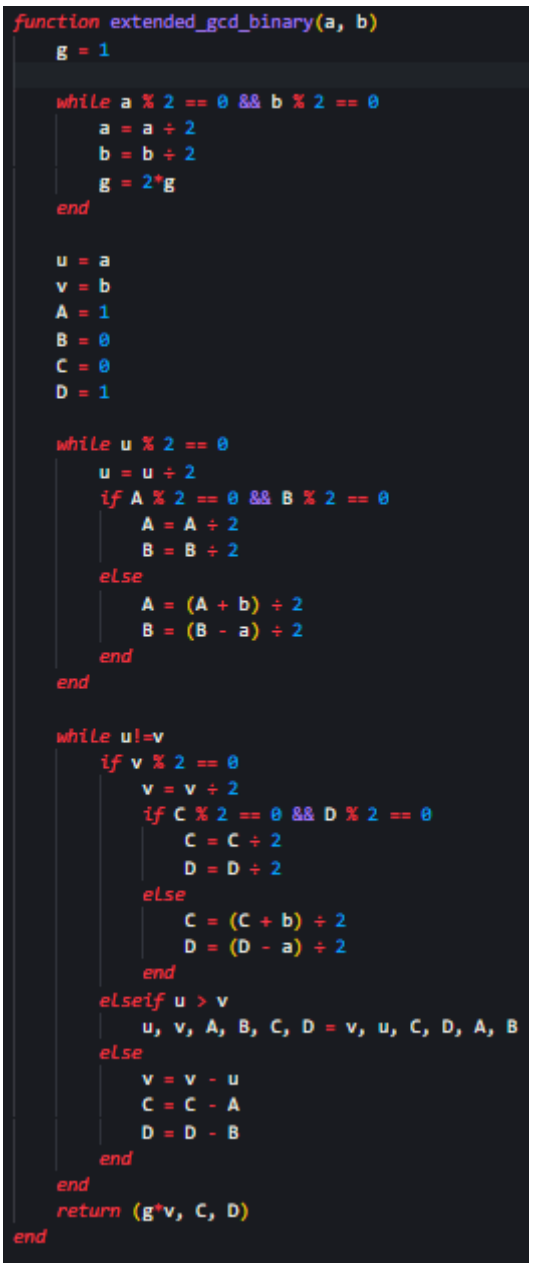


```
function gcd_binary(a, b)
    g = 1

        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. 4.5: Функция бинарного алгоритма Евклида

4.4 Расширенный бинарный алгоритм Евклида

Далее я создал расширенный бинарный алгоритм Евклида для нахождения коэффициентов и возврата НОД и соотношения Безу x и y (рис. 4.6).



```
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. 4.6: Функция расширенного алгоритма Евклида

Я создал блок инициализации переменных a и b и вызов функций с входными переменными (рис. 4.7).

```
a = 6
b = 4


println(gcd_binary(a, b))
println(extended_gcd_binary(a, b))
```

Рис. 4.7: Инициализация переменных и вызов функций(2)

И получил следующие значения (рис. 4.8).

```
2

(2, 1, -1)
```

Рис. 4.8: Результат выполнения(2)

5 Выводы

Я реализовал алгоритмы нахождения НОД.

Список Литературы

1. Julia - Control Flow
2. Julia - Mathematical Operations
3. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone - Handbook of Applied Cryptography