

Лабораторная работа 4

Вычисление наибольшего общего делителя

Климин Никита Денисович

Российский университет дружбы народов

Содержание

- **1 Цель работы**
- **1.1 Изучение и реализация алгоритмов вычисления наибольшего общего делителя целых чисел: алгоритм евклида, бинарный, расширенный и расширенный бинарный.**
- **2 Задание**
- **3 Теоретическое введение**
- **4 Выполнение лабораторной работы**
- **5 Выводы**
- **Список литературы**

1 Цель работы

1.1 Изучение и реализация алгоритмов вычисления наибольшего общего делителя целых чисел: алгоритм евклида, бинарный, расширенный и расширенный бинарный.

2 Задание

Реализовать четыре алгоритма вычисления НОД, проверить их работу и вывести результаты.

3 Теоретическое введение

Наибольший общий делитель (НОД) целых чисел a и b — это число $d \neq 0$, которое делит оба числа, и любое другое число, делящее a и b , делится на d .

НОД можно представить как линейную комбинацию: $d = ax + by$, где $x, y \in \mathbb{Z}$

Числа называются **взаимно простыми**, если их НОД равен 1.

Алгоритмы вычисления НОД: - **Классический алгоритм Евклида** использует повторное деление с остатком. - **Бинарный алгоритм (Штейна)** применяет побитовые операции для ускорения вычислений. - **Расширенные алгоритмы** позволяют дополнительно находить коэффициенты x и y для линейной комбинации.

4 Выполнение лабораторной работы

Программа была написана на Julia.

```
1 gcd_euclid(a::Int, b::Int) = b == 0 ? abs(a) : gcd_euclid(b, a % b) # алгоритм Евклида
2
3 function gcd_binary(a::Int, b::Int) # бинарный алгоритм
4     a, b = abs(a), abs(b) # используем положительные числа
5     a == 0 && return b # если одно из чисел 0, то возвращаем другое
6     b == 0 && return a
7
8     k = 0
9     while iseven(a) && iseven(b) # пока оба числа чётные
10         a >>= 1
11         b >>= 1
12         k += 1
13     end
14
15     while a != b # пока числа не равны
16         if iseven(a)
17             a >>= 1 # делим a на 2 если оно чётное
18         elseif iseven(b)
19             b >>= 1 # делим b на 2 если оно чётное
20         elseif a > b
21             a = (a - b) >> 1 # вычитаем и делим на 2
22         else
23             b = (b - a) >> 1 # вычитаем и делим на 2
24         end
25     end
26
27     return a << k # возвращаем нод
```

Пример работы программы в терминале

Рисунок 1: Пример работы программы

5 Выводы

Все алгоритмы корректно вычисляют НОД. Практическая проверка показала идентичные результаты для всех методов

Список литературы

Speaker notes