Отчёт по лабораторной работе 5

Вероятностные алгоритмы проверки чисел на простоту

Аристова Арина Олеговна

Содержание

Список иллюстраций

1 Цель работы

Изучить алгоритмы проверки чисел на простоту, реализовать их на языке Julia.

2 Задание

Реализовать несколько алгоритмов:

- алгоритм Ферма
- алгоритм Миллера-Рабина
- алгоритм Соловэя-Штрассена

3 Выполнение лабораторной работы

3.1 Код реализации алгоритмов

Выбор алгоритма, проверяемое число и число итераций программа получает из вводимых аргументов.

```
using Random

function modulo(base, exponent, mod)
    x = 1
    y = base % mod
    while exponent > 0
        if exponent % 2 == 1
            x = (x*y) % mod
        end
        y = (y*y) % mod
        exponent //= 2
    end
    return x % mod
end



function Ferma(n, count)
    for _ in 1:count
```

```
        a = rand(2:n-1)
        if powermod(a, n-1, n) != 1
            return false
        end
    end
    return true
end


function CalculateJacobian(a, n)
    if a == 0
        return 0
    end
    ans = 1
    if a < 0
        a = -a
        if n % 4 == 3
            ans = -ans
        end
    end
    if a == 1
        return ans
    end
    while a != 0
        if a < 0
            a = -a
            if n % 4 == 3
                ans = -ans
            end
```

```
        end
        while a % 2 == 0
            a //=2
            if n % 8 == 3 || n % 8 == 5
                ans = -ans
            end
        end
        a, n = n, a
        if a % 4 == 3 && n % 4 == 3
            ans = - ans
        end
        a = a%n
        if a > n //2
            a -= n
        end
    end
    return n == 1 ? ans : 0
end


function SoloveiStrassen(p, iterations)
    if p < 2 || (p != 2 && p % 2 == 0)
        return false
    end
    for _ in 1:iterations
        a = rand(1:p-1)
        jacobian = (p + CalculateJacobian(a, p)) % 2
        mod = modulo(a, (p-1)//2, p)
        if jacobian == 0 || mod != jacobian
```

```
            return false
        end
    end
    return true
end


function MillerRabin(n)
    if n in (0, 1, 4, 6, 8, 9)
        return false
    elseif n in (2, 3, 5, 7)
        return true
    end
    s = 0
    d = n-1
    while d % 2 == 0
        d >>= 1
        s += 1
    end
    function trial(a)
    if powermod(a, d, n) == n - 1
        return false
    end
    for i in 0:s-1
        if powermod(a, (1<<i)*d, n) == n-1
            return false
        end
    end
    return true
```

```
end
    for _ in 1:8
        a = rand(2:n-1)
        if trial(a)
            return false
        end
    end
    return true
end

method = lowercase(ARGS[1])
n = parse(Int, ARGS[2])
count = length(ARGS) > 2 ? parse(Int, ARGS[3]) : 25

println("Проверяется число $n методом $(uppercase(method)) (кол-
во итераций: $count)")

result = method == "f" ? Ferma(n, count) :
        method == "s" ? SoloveiStrassen(n, count) :
        method == "m" ? MillerRabin(n) :
        error("Неизвестный метод")

println(result ? "Simple" : "Complex")
```

3.2 Проверка работы кода

Проверяю работу кода. Алгоритм Ферма

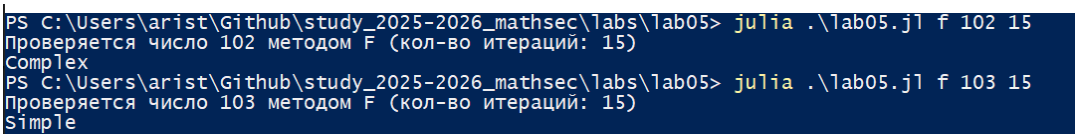

Рис. 3.1: Проверка работы кода для алгоритма Ферма.

Проверяю работу кода. Алгоритм Миллера-Рабина

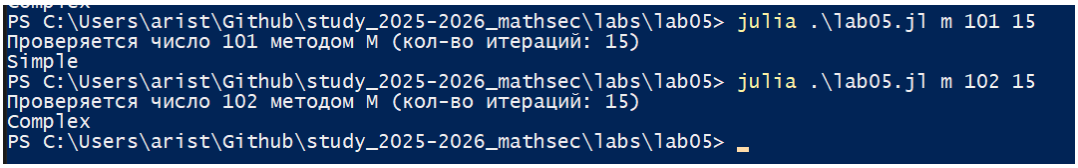

Рис. 3.2: Проверка работы кода для алгоритма Миллера-Рабина.

Проверяю работу кода. Алгоритм Соловэя-Штрассена

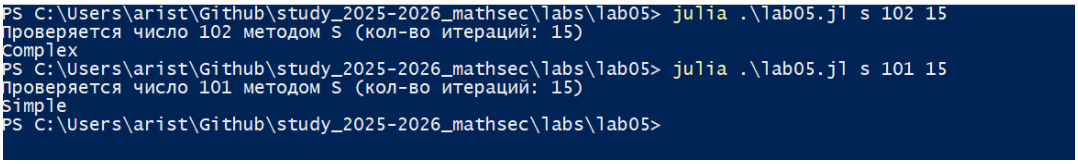

Рис. 3.3: Проверка работы кода для алгоритма Соловэя-Штрассена.

3.3 Вывод

В ходе выполнения лабораторной работы мною были реализованы программные решения алгоритмы проверки чисел на простоту: алгоритма Ферма, алгоритма Соловэя-Штрассена, алгоритма Миллера-Рабина.

Список литературы

- Описание лабораторной работы