Отчет по лабораторной работе №5

дисциплина: Математические основы защиты информации и информационной безопасности

Морозова Ульяна Константиновна

Содержание

1 Цель работы

Целью работы является изучение алгоритмов проверки чисел на простоту и реализация их на языке Julia.

2 Выполнение лабораторной работы

2.1 Тест Ферма

Тест Ферма — вероятностный тест для проверки простоты натурального числа n, основанный на малой теореме Ферма. Даёт ответ о составности числа либо его несоставности лишь с некоторой вероятностью.

Далее приведена реализация шифра на языке Julia.

```
function fermat_test(n, k=10)
    println("Тестируем число $n с k = $k")

    if n <= 1
        # println("$n не является простым (n <= 1)")
        return false
    elseif n == 2 || n == 3
        # println("$n является простым числом")
        return true
    elseif n % 2 == 0
        # println("$n составное (четное число)")
        return false
    end

    for i in 1:k
        a = rand(2:(n-2))
```

```
        result = powermod(a, n-1, n)
        # println("Тест $i: $a^($n-1) mod $n = $result")

        if result != 1
            # println("$a^($n-1) mod $n = $result ≠ 1")
            # println("$n составное (свидетель: $a)")
            return false
        end
    end

    # println("$n вероятно простое (все $k тестов пройдены)")
    return true
end

function test_fermat_examples()
    test_numbers = [
        7, 11, 13, 17, 19,  # Простые числа
        9, 10, 12, 14, 15,  # Составные числа
        35, 548, 827, 1983
    ]

    println("Тест Ферма для различных чисел:")
    println("="^50)

    for n in test_numbers
        result = fermat_test(n, 5)
        status = result ? "вероятно простое" : "составное"
        println("$n: $status")
    end
```

```
end
```

И результат его работы

```
Тест Ферма для различных чисел:
================================================
Тестируем число 7 с k = 5
7: вероятно простое
Тестируем число 11 с k = 5
11: вероятно простое
Тестируем число 13 с k = 5
13: вероятно простое
Тестируем число 17 с k = 5
17: вероятно простое
Тестируем число 19 с k = 5
19: вероятно простое
Тестируем число 9 с k = 5
9: составное
Тестируем число 10 с k = 5
10: составное
Тестируем число 12 с k = 5
12: составное
Тестируем число 14 с k = 5
14: составное
Тестируем число 15 с k = 5
15: составное
Тестируем число 35 с k = 5
35: составное
Тестируем число 548 с k = 5
548: составное
Тестируем число 827 с k = 5
```

```
827: вероятно простое
Тестируем число 1983 с k = 5
1983: составное
```

2.2 Тест Соловея-Штрассена и символ Якоби

Тест Соловея — Штрассена — вероятностный тест простоты, открытый в 1970-х годах Робертом Мартином Соловеем совместно с Фолькером Штрассеном. Тест опирается на малую теорему Ферма и свойства символа Якоби.

```
function jacobi_symbol(a, n)

    if n % 2 == 0 || n <= 0
        throw(ArgumentError("n должно быть нечетным положительным целым"))
    end

    # Приводим a по модулю n
    a = a % n
    result = 1

    while a != 0
        # Убираем множители 2
        while a % 2 == 0
            a ÷= 2
            # (2/n) = (-1)^((n^2-1)/8)
            if n % 8 == 3 || n % 8 == 5
                result = -result
            end
        end
```

```
        # Меняем местами по квадратичному закону взаимности
        a, n = n, a

        # (a/n) = (-1)^((a-1)(n-1)/4) * (n/a)
        if a % 4 == 3 && n % 4 == 3
            result = -result
        end

        a = a % n
    end

    return n == 1 ? result : 0
end

function solovay_strassen_test(n, k=10)

    # Обработка особых случаев
    if n <= 1
        return false
    elseif n == 2
        return true
    elseif n % 2 == 0
        return false
    end

    # Проверяем k случайных оснований
    for _ in 1:k
        # Выбираем случайное a в диапазоне [2, n-1]
        a = rand(2:(n-1))
```

```
        # Вычисляем символ Якоби
        jacobi = jacobi_symbol(a, n)
        if jacobi == 0
            return false  # gcd(a, n) > 1, число составное
        end

        # Вычисляем a^((n-1)/2) mod n
        exponent = (n - 1) ÷ 2
        mod_result = powermod(a, exponent, n)

        # Приводим символ Якоби к модулю n
        jacobi_mod = jacobi >= 0 ? jacobi : jacobi + n

        # Проверяем условие Эйлера
        if mod_result != jacobi_mod
            return false
        end
    end

    return true
end

function test_soloveya_examples()
    test_numbers = [
        7, 11, 13, 17, 19,  # Простые числа
        9, 10, 12, 14, 15,  # Составные числа
        35, 548, 827, 1983
    ]
```

```
    println("Тест Соловея-Штрассена для различных чисел:")
    println("="^50)

    for n in test_numbers
        result = solovay_strassen_test(n, 10)
        status = result ? "вероятно простое" : "составное"
        println("$n: $status")
    end
end
```

Результат его работы

```
Тест Соловея-Штрассена для различных чисел:
==================================================
7: вероятно простое
11: вероятно простое
13: вероятно простое
17: вероятно простое
19: вероятно простое
9: составное
10: составное
12: составное
14: составное
15: составное
35: составное
548: составное
827: вероятно простое
1983: составное
```

2.3 Тест Миллера-Рабина

Тест Миллера — Рабина на простоту — вероятностный полиномиальный тест, который позволяет эффективно определить, является ли данное число составным. Однако с его помощью строго доказать простоту числа нельзя.

```
function miller_rabin_test(n, k=10)

    # Обработка особых случаев
    if n <= 1
        return false
    elseif n == 2 || n == 3
        return true
    elseif n % 2 == 0
        return false
    end

    # Представляем n-1 в виде d * 2^s
    d = n - 1
    s = 0
    while d % 2 == 0
        d ÷= 2
        s += 1
    end

    # Проверяем k случайных оснований
    for _ in 1:k
        a = rand(2:(n-2))
        x = powermod(a, d, n)
```

```
        if x == 1 || x == n-1
            continue
        end

        # Проверяем последовательные возведения в квадрат
        composite = true
        for _ in 1:(s-1)
            x = powermod(x, 2, n)
            if x == n-1
                composite = false
                break
            elseif x == 1
                return false
            end
        end

        if composite
            return false
        end
    end

    return true
end

function test_miller_rabin()
    """
    Простая функция тестирования алгоритма Миллера-Рабина
    """
    println("Тестирование простого алгоритма Миллера-Рабина")
```

```
    println("="^50)

    # Тестовые числа
    test_numbers = [
        # Большие простые числа
        1009, 1013, 10007, 10009,

        # Большие составные числа
        1001, 1027, 10005, 10011,

    ]

    k = 5  # Количество тестов

    for n in test_numbers
        result = miller_rabin_test(n, k)
        status = result ? "простое" : "составное"
        println("$n: $status")
    end
end
```

Результаты:

```
Тестирование простого алгоритма Миллера-Рабина
==================================================
1009: простое
1013: простое
10007: простое
10009: простое
1001: составное
1027: составное
```

```
10005: составное
10011: составное
```

3 Выводы

Мы изучили работу алгоритмов, а также реализовали их на языке Julia.