Отчёт по лабораторной работе 5

Супонина Анастасия Павловна

Содержание

Список иллюстраций

Список таблиц

Элементы списка иллюстраций не найдены.

Цель работы

Реализовать три вероятностных алгоритма проверки чисел на простоту и алгоритм вычисления символа Якоби.

Задание

Программно реализовать на языке Julia следующие алгоритмы:

1. Алгоритм, реализующий тест Ферма
2. Алгоритм вычисления символа Якоби
3. Алгоритм, реализующий тест Соловэя-Штрассена
4. Алгоритм, реализующий тест Миллера-Рабина

Теоретическое введение

При написании данных алгоритмов часто приходиться реализовывать запись

$$a \equiv b \ (mod \ m)$$

Формула

Которая в языке программирования записывается как a - b % m == 0, что означается деление без остатка.

Выполнение лабораторной работы

1. Алгоритм, реализующий тест Ферма

Начинаю написание программы с ввода значений

```
1  n = 17
2  a = 3
```

Введение обозначений

Реализую код с проверкой данных значений, учитывая условия используемого алгоритма

```
# Проверка условий

if (n % 2 == 0) | (n < 5)
    println("Введите другое n")
else
    println("Всё отлично продолжаем работу")
end


if (a < 2) | (a > n - 2)
    println("Введите другое a")
else
    println("Всё отлично продолжаем работу")
end
```

проверка условий

По формуле ферма реализую программный код код, вычисляя значение r и проверяя его для получения результата

```
# Тест Ферма

r = a^(n-1) % n

if r == 1
    println("Число n, вероятно, простое")
else
    println("Число n составное")
end
```

Реализация

Выполняю программу и получаю результат

```
Всё отлично продолжаем работу
Всё отлично продолжаем работу
Число n, вероятно, простое
```

Результат

2. Алгоритм вычисления символа Якоби

Аналогично предыдущей программе начинаю с ввода значений и проверки правильности введенных значений учитывая условие задачи

```
n = 15
a = 9

# Проверка условий

if (n % 2 == 0) | (n < 3)
    println("Введите другое n")
else
    println("Всё отлично продолжаем работу")
end


if (a < 0) | (a >= n)
    println("Введите другое a")
else
    println("Всё отлично продолжаем работу")
end
```

Введение обозначений и проверка условий

Для того, чтобы реализовать алгоритм необходимо представить число a, как произведение простого числа на 2 в k степени. Для этого создаю отдельную функцию

```
# функция для приведения a к виду 2^k*a1

function devide(a)
    k = 0
    while a % 2 == 0
        k += 1
        a = Int(a / 2)
    end
    return a, k
end
```

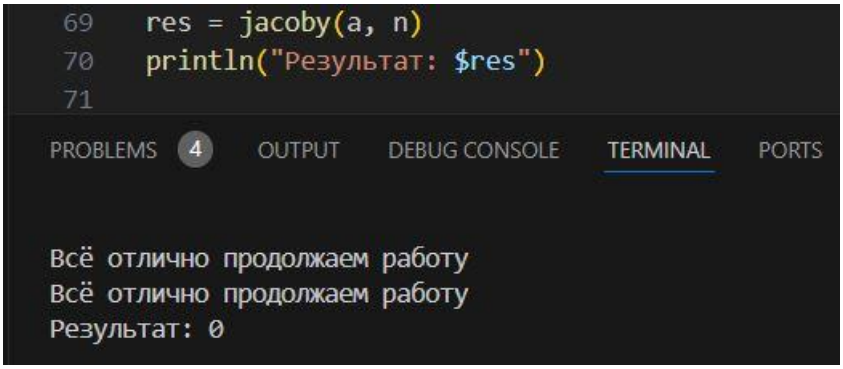
Функция для выделения четной части

Создаю алгорим вычисления символа Якоби использую функцию написанную ранее и правила создания указанные в документе с заданием

```
function jacoby(a, n, g = 1)

    while a >= 0
        if a == 0
            return 0
        elseif a == 1
            return g
        end

        a1, k = devide(a)

        if (k % 2 == 0)
            s = 1
        else
            if ((n-1) % 8 == 0) || ((n + 1) % 8 == 0)
                s = 1
            elseif ((n - 3) % 8 == 0) || ((n + 3) % 8 == 0)
                s = -1
            end
        end

        if a1 == 1
            result = g*s
            return result
        end

        if ((n - 3) % 4 == 0) && ((a1 - 3) % 4 == 0)
            s = s * (-1)
        end

        a = n % a1
        n = a1
        g = g * s
    end
end
```

Реализация

Выполняю программу и получаю результат



Результат

3. Алгоритм, реализующий тест Соловэя-Штрассена

Аналогично предыдущей программе начинаю с ввода значений и проверки правильности введенных значений учитывая условие задачи

```
n = 17
a = 3

# Проверка условий

if (n % 2 == 0) || (n < 5)
    println("Введите другое n")
else
    println("Всё отлично продолжаем работу")
end


if (a < 2) || (a > n - 2)
    println("Введите другое a")
else
    println("Всё отлично продолжаем работу")
end
```

Введение обозначений и проверка условий

В данной задаче необходимо найти символ Якоби, его я искала при помощи функции реализованной на ранее. По формуле Соловэя-Штрассена вычисляю r и при помощи функции для символа Якоби реализую данный алгоритм

```
r = (a^((n-1)/2)) % n

function solovey_shtrassen(r, n, a)
    if r != 1 && r != n - 1
        return("Число $n составное")
    end

    s = jacoby(a, n)

    if s < 0
        s += n
    end

    if (r - s) % n == 0
        return("Число $n, вероятно, простое")
    else
        return("Число $n составное")
    end
end

res = solovey_shtrassen(r, n, a)
println(res)
```

Реализация

Выполняю программу и получаю результат

```
Всё отлично продолжаем работу
Всё отлично продолжаем работу
Число 17, вероятно, простое
```

Результат

4. Алгоритм, реализующий тест Миллера-Рабина

Аналогично предыдущей программе начинаю с ввода значений и проверки правильности введенных значений учитывая условие задачи

```ruby
n = 15
a = 2

# Проверка условий

if (n % 2 == 0) | (n < 5)
    println("Введите другое n")
else
    println("Всё отлично продолжаем работу")
end

if (a < 2) | (a > n - 2)
    println("Введите другое a")
else
    println("Всё отлично продолжаем работу")
end
```

Введение обозначений и проверка условий

В данной задаче использую функцию написанную в алгоритме для символа Якоби, чтобы представить число в виде произведения некоторого нечетного числа на 2 в степени k. В остальном использую формулы для вычисления Миллера-Рабина пишу следующий код

```
y = (a ^ r) % n

function miller_rabin(y, n, s)
    if y == 1 || y == n - 1
        return "Число $n, вероятно, простое"
    end

    for j in 1:(s - 1)
        y = (y * y) % n

        if y == n - 1
            return "Число $n, вероятно, простое"
        end

        if y == 1
            return "Число $n составное"
        end
end

# Если ни одно из условий не выполнено, то n составное
return "Число $n составное"
end
```

Реализация

Выполняю программу и получаю результат

```
res = miller_rabin(y, n, s)
println(res)
```

```
PROBLEMS 4   OUTPUT   DEBUG CONSOLE   TERMINAL   P

Всё отлично продолжаем работу
Число 15 составное

Всё отлично продолжаем работу
Всё отлично продолжаем работу
Число 15 составное
```

Результат

Выводы

В процессе выполнения работы, я реализовала разные виды вероятностных алгоритмов проверки чисел на простоту на языке программирования Julia.

Список литературы

::: Пособие по лабораторной работе 5 {file:///C:/Users/bermu/Downloads/lab05.pdf}