# Математические основы защиты информации и информационной безопасности

Супонина Анастасия Павловна

5 Октября 2024

РУДН, Москва, Россия

# Лабораторная работа 5

# 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

# 1. Тест Ферма - Введение обозначений и проверка условий

```
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
```

```
19   # Тест Ферма
20
21   r = a^(n-1) % n
22
23   if r == 1
24       println("Число n, вероятно, простое")
25   else
26       println("Число n составное")
27   end
```

**Рис. 1:** Реализация

# 2. Символ Якоби - Введение обозначений и проверка условий

```
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
```

```
19  # функция для приведения a к виду 2^k*a1
20
21  function devide(a)
22      k = 0
23      while a % 2 == 0
24          k += 1
25          a = Int(a / 2)
26      end
27      return a, k
28  end
```

Рис. 3: Функция для выделения четной части

# 2. Символ Якоби - Реализация

```
33  function jacoby(a, n, g = 1)
34
35      while a >= 0
36          if a == 0
37              return 0
38          elseif a == 1
39              return g
40          end
41
42          a1, k = devide(a)
43
44          if (k % 2 == 0)
45              s = 1
46          else
47              if ((n-1) % 8 == 0) || ((n + 1) % 8 == 0)
48                  s = 1
49              elseif ((n - 3) % 8 == 0) || ((n + 3) % 8 == 0)
50                  s = -1
51              end
52          end
53
54          if a1 == 1
55              result = g*s
56              return result
57          end
58
59          if ((n - 3) % 4 == 0) && ((a1 - 3) % 4 == 0)
60              s = s * (-1)
61          end
62
63          a = n % a1
64          n = a1
65          g = g * s
66      end
67  end
```

# 3. Тест Соловэя-Штрассена - Введение обозначений и проверка условий

```
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
```

# 3. Тест Соловэя-Штрассена - Реализация

```
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)
```

# 4. Тест Миллера-Рабина - Введение обозначений и проверка условий

```
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
```

# 4. Тест Миллера-Рабина - Реализация

```
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
```

# Выводы

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

Спасибо за внимание!