Лабораторная работа №5: Вероятностные алгоритмы проверки чисел на простоту

Дисциплина: Математические основы защиты информации и информационной безопасности

Манаева Варвара Евгеньевна, НФИмд-01-24, 1132249514

09 ноября 2024

Российский университет дружбы народов, Москва, Россия

Общая информация о лабораторной работе

Цель работы

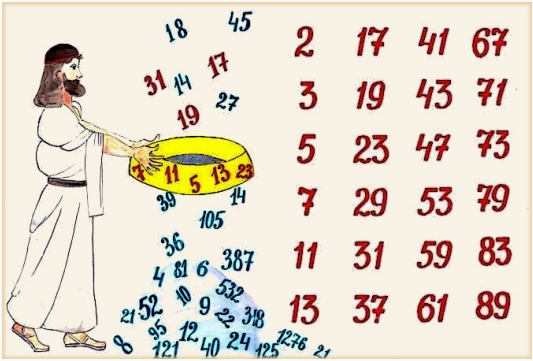
Ознакомиться с алгоритмами вероятностной проверки чисел на простоту.

Задание

1. Реализовать тест Ферма;
2. Реализовать алгоритм вычисления символа Якоби;
3. Реализовать тест Соловэя-Штрассена;
4. Реализовать тест Миллера-Робина.

Теоретическое введение

Проверки чисел на простоту



Существует два типа критериев простоты:

- детерминированные;
- вероятностные.

Выполнение лабораторной работы

1. Реализовать тест Ферма

1. Реализовать тест Ферма

[1]:
```
function testFerma(n)
    if n < 5
        return "Incorrect input."
    end
    a = rand(2:n-2)
    r = powermod(a, n-1, n)
    if r == 1
        return "Число " * string(n) * ", вероятно, простое."
    else
        return "Число " * string(n) * " составное."
    end
end
```

[1]:
```
testFerma (generic function with 1 method)
```

Проверка работы функции

```
display(testFerma(441))
display(testFerma(443))
```

Результат выполнения запуска функции шифрования

1. Тест Ферма

[1]:
```
function testFerma(n)
    if n < 5
        return "Incorrect input."
    end
    a = rand(2:n-2)
    r = powermod(a, n-1, n)
    if r == 1
        return "Число " * string(n) * ", вероятно, простое."
    else
        return "Число " * string(n) * " составное."
    end
end
```

[1]:
```
testFerma (generic function with 1 method)
```

[2]:
```
display(testFerma(441))
display(testFerma(443))
```

```
"Число 441 составное."
"Число 443, вероятно, простое."
```

2. Реализовать алгоритм вычисления символа Якоби

2. Реализовать алгоритм вычисления символа Якоби

2. Символ Якоби

[3]:
```
function YacobySymbol(n, a)
    if n < 3 || a >= n || a < 0
        return "Incorrect input."
    end
    g = 1
    a1 = 0
    k = 0
    s = 0
    while a1 != 1
        if a == 0
            return 0
        elseif a == 1
            return 1
        end
        a1 = a
        k = 0
        while a1 % 2 == 0
            k += 1
            a1 = round(Int64, a1 / 2)
        end

        if k % 2 == 0 || (k % 2 == 1 && (n % 8 == 1 || n % 8 == 7))
            s = 1
        elseif k % 2 == 1 && (n % 8 == 3 || n % 8 == 5)
            s = -1
        end
        if a1 == 1
            return g*s
        end

        if n % 4 == 3 && a % 4 == 3
            s = -s
        end

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

Проверка работы функции

```
YacobySymbol(443, 359)
```

Результат выполнения запуска функции шифрования

```
            a1 = round(Int64, a1 / 2)
        end

        if k % 2 == 0 || (k % 2 == 1 && (n % 8 == 1 || n % 8 == 7))
            s = 1
        elseif k % 2 == 1 && (n % 8 == 3 || n % 8 == 5)
            s = -1
        end
        if a1 == 1
            return g*s
        end

        if n % 4 == 3 && a % 4 == 3
            s = -s
        end

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

[3]: YacobySymbol (generic function with 1 method)

[4]:
```
YacobySymbol(443, 359)
```

[4]: 1

3. Реализовать тест Соловэя-Штрассена

3. Реализовать тест Соловэя-Штрассена

[5]:
```
function testSoloveyaShtrassena(n)
    if n < 5
        return "Incorrect input."
    end
    a = rand(2:n-2)
    r = powermod(a, round(Int64, (n-1)/2), n)
    if r != 1 && r != n-1
        return "Число " * string(n) * " составное."
    else
        s = YacobySymbol(n, a)
        if r == s && r != NaN
            return "Число " * string(n) * " составное."
        end
        return "Число " * string(n) * ", вероятно, простое."
    end
end
```

[5]:
```
testSoloveyaShtrassena (generic function with 1 method)
```

Проверка работы функции

```
display(testSoloveyaShtrassena(4463429))
display(testSoloveyaShtrassena(443))
```

Результат выполнения запуска функции шифрования

3. Тест Соловэя-Штрассена

[5]:
```
function testSoloveyaShtrassena(n)
    if n < 5
        return "Incorrect input."
    end
    a = rand(2:n-2)
    r = powermod(a, round(Int64, (n-1)/2), n)
    if r != 1 && r != n-1
        return "Число " * string(n) * " составное."
    else
        s = YacobySymbol(n, a)
        if r == s && r != NaN
            return "Число " * string(n) * " составное."
        end
        return "Число " * string(n) * ", вероятно, простое."
    end
end
```

[5]:
```
testSoloveyaShtrassena (generic function with 1 method)
```

[6]:
```
display(testSoloveyaShtrassena(4463429))
display(testSoloveyaShtrassena(443))
```

```
"Число 4463429 составное."
"Число 443, вероятно, простое."
```

4. Реализовать тест Миллера-Робина

4. Реализовать тест Миллера-Робина

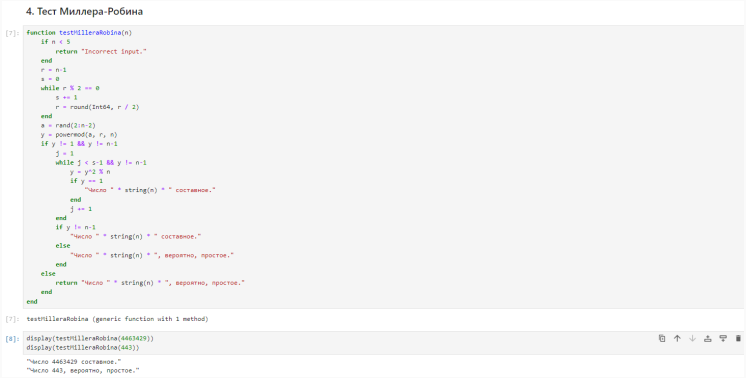
[7]:
```
function testMilleraRobina(n)
    if n < 5
        return "Incorrect input."
    end
    r = n-1
    s = 0
    while r % 2 == 0
        s += 1
        r = round(Int64, r / 2)
    end
    a = rand(2:n-2)
    y = powermod(a, r, n)
    if y != 1 && y != n-1
        j = 1
        while j < s-1 && y != n-1
            y = y^2 % n
            if y == 1
                "Число " * string(n) * " составное."
            end
            j += 1
        end
        if y != n-1
            "Число " * string(n) * " составное."
        else
            "Число " * string(n) * ", вероятно, простое."
        end
    else
        return "Число " * string(n) * ", вероятно, простое."
    end
end
```

[7]:
```
testMilleraRobina (generic function with 1 method)
```

Проверка работы функции

```
display(testMilleraRobina(4463429))
display(testMilleraRobina(443))
```

Результат выполнения запуска функции шифрования



4. Тест Миллера-Робина

```
[7]: function testMilleraRobina(n)
         if n < 5
             return "Incorrect input."
         end
         r = n-1
         s = 0
         while r % 2 == 0
             s += 1
             r = round(Int64, r / 2)
         end
         a = rand(2:n-2)
         y = powermod(a, r, n)
         if y != 1 && y != n-1
             j = 1
             while j < s-1 && y != n-1
                 y = y^2 % n
                 if y == 1
                     "число " * string(n) * " составное."
                 end
                 j += 1
             end
             if y != n-1
                 "число " * string(n) * " составное."
             else
                 "число " * string(n) * ", вероятно, простое."
             end
         else
             return "число " * string(n) * ", вероятно, простое."
         end
     end
```

```
[7]: testMilleraRobina (generic function with 1 method)
```

```
[8]: display(testMilleraRobina(4463429))
     display(testMilleraRobina(443))
```

```
"число 4463429 составное."
"число 443, вероятно, простое."
```

Выводы

Выводы

В результате работы мы ознакомились с вероятностными алгоритмами проверки чисел на простоту, а именно:

- Тестом Ферма;
- Алгоритмом вычисления символа Якоби;
- Тестом Соловэя-Штрассена;
- Тестом Миллера-Робина.

Были записаны скринкасты:

- выполнения лабораторной работы;
- создания отчёта по результатам выполения лабораторной работы;
- создания презентации по результатам выполнения лабораторной работы;
- защиты лабораторной работы.