

Лабораторная работа №5

Вероятностные алгоритмы проверки чисел на простоту

Тазаева Анастасия Анатольевна

2025-10-04

Содержание I

1. Информация

2. Введение

3. Программный код

4. Заключение

Раздел 1

1. Информация

1.1 Докладчик

► Тазаева Анастасия Анатольевна



1.1 Докладчик

- ▶ Тазаева Анастасия Анатольевна
- ▶ студент группы НФИмд-02-25



1.1 Докладчик

- ▶ Тазаева Анастасия Анатольевна
- ▶ студент группы НФИмд-02-25
- ▶ Российский университет дружбы народов им. П. Лумумбы



1.1 Докладчик

- ▶ Тазаева Анастасия Анатольевна
- ▶ студент группы НФИмд-02-25
- ▶ Российский университет дружбы народов им. П. Лумумбы
- ▶ 1032259385@pfur.ru



Раздел 2

2. Введение

2.1 Цель работы

Ознакомиться с алгоритмами проверки чисел на простоту. Реализовать их.

2.2 Задачи

Реализовать на языке программирования Julia:

1. Алгоритм, реализующий тест Ферма.

2.2 Задачи

Реализовать на языке программирования Julia:

1. Алгоритм, реализующий тест Ферма.
2. Алгоритм вычисления числа Якоби.

2.2 Задачи

Реализовать на языке программирования Julia:

1. Алгоритм, реализующий тест Ферма.
2. Алгоритм вычисления числа Якоби.
3. Алгоритм, реализующий тест Соловея-Штрассена.

2.2 Задачи

Реализовать на языке программирования Julia:

1. Алгоритм, реализующий тест Ферма.
2. Алгоритм вычисления числа Якоби.
3. Алгоритм, реализующий тест Соловея-Штассена.
4. Алгоритм, реализующий тест Миллера-Рабина.

Раздел 3

3. Программный код

3.1 Алгоритм, реализующий тест Ферма

```
function test_Ferma(n)
    if n < 5
        return "incorrect n.please try again \n"
    end
    a = rand(2:n-2)
    r = powermod(a, n-1,n)
    if r == 1
        return "4islo " * string(n) * ", veroyatno, prostoe\n"
    else
        return "4islo " * string(n) * " sostavnoe \n"
    end
end
```

3.2 Алгоритм, реализующий тест Ферма

```
print(test_Ferma(43))
print(test_Ferma(44))
print(test_Ferma(4))
```

4islo 43, veroyatno, prostoe
4islo 44 sostavnoe
incorrect n.please try again

Рисунок 1: Алгоритм, реализующий тест Ферма. Пример отработки

3.3 Алгоритм вычисления числа Якоби

```
function simvol_Yakobi(n,a)
    if n < 3 || a >= n || a < 0
        return "\nincorrect n,a.please try again\n"
    end
    g = 1
    a1 = 0
    k = 0
    s = 0
    while a1 != 1
        if 1 == 0
            return 0
        elseif a == 1
            return 1
        end
    end
```

3.4 Алгоритм вычисления числа Якоби

```
print(simvol_Yakobi(44, 40))  
print(simvol_Yakobi(44, 50))
```

0

incorrect n,a. please try again

Рисунок 2: Алгоритм вычисления числа Якоби. Пример отработки

3.5 Алгоритм, реализующий тест Соловея-Штрассена

```
function test_Soloveya_Shtrassena(n)
    if n < 5
        return "incorrect n.please try again"
    end
    a = rand(2:n-2)
    r = powermod(a, round(Int64, (n-1)/2), n)
    if r != 1 && r != n-1
        return "4islo " * string(n) * " sostavnoe \n"
    else
        s = simvol_Yakobi(n,a)
        if r == s && r != NaN
            return "4islo " * string(n) * " sostavnoe \n"
        end
        return "4islo " * string(n) * ", veroyatno, prostoe \n"
```

3.6 Алгоритм, реализующий тест Соловея-Штрассена

```
print(test_Soloveya_Shtrassena(43))
print(test_Soloveya_Shtrassena(44))
print(test_Soloveya_Shtrassena(4))
```

4islo 43, veroyatno, prostoe
4islo 44 sostavnoe
incorrect n.please try again

Рисунок 3: Алгоритм, реализующий тест Соловея-Штрассена. Пример отработки

3.7 Алгоритм, реализующий тест Миллера-Рабина

```
function test_Millera_Robina(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
```

3.8 Алгоритм, реализующий тест Миллера-Рабина

[80]:

```
test_Millera_Robina(44)
```

[80]:

```
"4islo 44 sostavnoe \n"
```

[81]:

```
test_Millera_Robina(43)
```

[81]:

Раздел 4

4. Заключение

4.1 Вывод

В ходе лабораторной работы реализованы на языке программирования Julia:

1. Алгоритм, реализующий тест Ферма.

4.1 Вывод

В ходе лабораторной работы реализованы на языке программирования Julia:

1. Алгоритм, реализующий тест Ферма.
2. Алгоритм вычисления числа Якоби.

4.1 Вывод

В ходе лабораторной работы реализованы на языке программирования Julia:

1. Алгоритм, реализующий тест Ферма.
2. Алгоритм вычисления числа Якоби.
3. Алгоритм, реализующий тест Соловея-Штрассена.

4.1 Вывод

В ходе лабораторной работы реализованы на языке программирования Julia:

1. Алгоритм, реализующий тест Ферма.
2. Алгоритм вычисления числа Якоби.
3. Алгоритм, реализующий тест Соловея-Штассена.
4. Алгоритм, реализующий тест Миллера-Рабина.