

Отчёт по лабораторной работе 5

Андианова Марина Георгиевна

Содержание

Цель работы	2
Задание.....	2
Теоретическое введение	2
Выполнение лабораторной работы.....	2
1. Алгоритм, реализующий тест Ферма	2
2. Алгоритм вычисления символа Якоби	3
3. Алгоритм, реализующий тест Соловэя-Штассена.....	6
4. Алгоритм, реализующий тест Миллера-Рабина	7
Выводы.....	9

Список иллюстраций

Формула	2
Введение обозначений	2
Проверка условий	3
Реализация	3
Результат.....	3
Введение обозначений и проверка условий.....	4
Функция для выделения четной части	4
Реализация	5
Результат.....	6
Введение обозначений и проверка условий.....	6
Реализация	7
Результат.....	7
Введение обозначений и проверка условий.....	8
Реализация	9
Результат.....	9

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-Штассена
4. Алгоритм, реализующий тест Миллера-Рабина

Теоретическое введение

При написании данных алгоритмов часто приходится реализовывать запись

$$a \equiv b \pmod{m}$$

Формула

которая в языке программирования записывается как `a - b % m == 0`, что означает 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

```
4  # Проверка условий
5
6  if (n % 2 == 0) | (n < 5)
7  |   println("Введите другое n")
8  else
9  |   println("Всё отлично продолжаем работу")
10 end
11
12
13 if (a < 2) | (a > n - 2)
14 |   println("Введите другое a")
15 else
16 |   println("Всё отлично продолжаем работу")
17 end
```

Проверка условий

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

```
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
```

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

```
1 n = 15
2 a = 9
3
4 # Проверка условий
5
6 if (n % 2 == 0) | (n < 3)
7     println("Введите другое n")
8 else
9     println("Всё отлично продолжаем работу")
10 end
11
12
13 if (a < 0) | (a >= n)
14     println("Введите другое a")
15 else
16     println("Всё отлично продолжаем работу")
17 end
18
```

Введение обозначений и проверка условий

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

```
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
```

Функция для выделения четной части

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

```

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

```

Реализация

Выполняю программу и получаю результат

```
69     res = jacob(a, n)
70     println("Результат: $res")
71
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Всё отлично продолжаем работу
Всё отлично продолжаем работу
Результат: 0
```

Результат

3. Алгоритм, реализующий тест Соловэя-Штрассена

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

```
1  n = 17
2  a = 3
3
4  # Проверка условий
5
6  if (n % 2 == 0) || (n < 5)
7  |   println("Введите другое n")
8  else
9  |   println("Всё отлично продолжаем работу")
10 end
11
12
13 if (a < 2) || (a > n - 2)
14 |   println("Введите другое a")
15 else
16 |   println("Всё отлично продолжаем работу")
17 end
```

Введение обозначений и проверка условий

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

```
57 r = (a^(n-1)/2)) % n
58
59 ✓ function solovey_shtrassen(r, n, a)
60   ✓ if r != 1 && r != n - 1
61     | return("Число $n составное")
62   end
63
64   s = jacoby(a, n)
65
66   ✓ if s < 0
67     | s += n
68   end
69
70   ✓ if (r - s) % n == 0
71     | return("Число $n, вероятно, простое")
72   ✓ else
73     | return("Число $n составное")
74   end
75 end
76
77 res = solovey_shtrassen(r, n, a)
78 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

```
1 n = 15
2 a = 2
3
4 # Проверка условий
5
6 < if (n % 2 == 0) | (n < 5)
7 |   println("Введите другое n")
8 < else
9 |   println("Всё отлично продолжаем работу")
10 end
11
12 < if (a < 2) | (a > n - 2)
13 |   println("Введите другое a")
14 < else
15 |   println("Всё отлично продолжаем работу")
16 end
17
```

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

```
30  y = (a ^ r) % n
31
32  function miller_rabin(y, n, s)
33      if y == 1 || y == n - 1
34          return "Число $n, вероятно, простое"
35      end
36
37      for j in 1:(s - 1)
38          y = (y * y) % n
39
40          if y == n - 1
41              return "Число $n, вероятно, простое"
42          end
43
44          if y == 1
45              return "Число $n составное"
46          end
47      end
48
49 # Если ни одно из условий не выполнено, то п составное
50 return "Число $n составное"
51 end
52
```

Реализация

Выполняю программу и получаю результат

```
53  res = miller_rabin(y, n, s)
54  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.