

Лабораторная работа №5: отчет.

вероятностные алгоритмы проверки чисел на простоту

Евдокимов Максим Михайлович. Группа - НФИмд-01-24.

Содержание

Цели и задачи работы	4
Цель лабораторной работы	4
Задание	4
Теоретическое введение	5
Тест Ферма	5
Алгоритм действий:	5
Сравнение:	5
Символ Якоби	5
Свойства символа Якоби	6
Тест Соловья-Штрассена	7
Алгоритм действий:	7
Сравнение:	7
Тест Миллера-Рабина	7
Алгоритм действий:	7
Сравнение:	8
Вывод:	8
Ход работы	9
Подготовка	9
Алгоритм теста Ферма	10
Результат 1	11
Алгоритм вычисления символа Якоби	11
Результат 2	13
Алгоритм теста Соловья-Штрассена	14
Результат 3	15
Алгоритм теста Миллера-Рабина	16
Результат 4	18
Выводы по проделанной работе	19
Вывод	19
Список литературы	20

Список иллюстраций

1	Случайная тестовая группа	9
2	Код теста Ферма	10
3	Результат шифра Цезаря	11
4	Код вычисления символа Якоби	12
5	Результат вычисления символа Якоби	13
6	Код теста Соловья-Штрассена	14
7	Результат теста Соловья-Штрассена	15
8	Код теста Миллера-Рабина 1	16
9	Код теста Миллера-Рабина 2	17
10	Результат теста Миллера-Рабина	18

Цели и задачи работы

Цель лабораторной работы

Вероятностные алгоритмы проверки чисел на простоту

Задание

1. Реализовать алгоритм теста Ферма.
2. Реализовать алгоритм вычисления символа Якоби.
3. Реализовать алгоритм теста Соловья-Штрассена.
4. Реализовать алгоритм теста Миллера-Рабина.

Теоретическое введение

Тест Ферма

Тест Ферма основан на малой теореме Ферма, которая утверждает, что если n — простое число, то для любого целого a такого, что $1 \leq a < n$, выполняется:

$$a^{n-1} \equiv 1 \pmod{n}$$

Алгоритм действий:

1. Выбрать случайное число a такое, что $1 \leq a < n$.
2. Вычислить $a^{n-1} \pmod{n}$.
3. Если $a^{n-1} \not\equiv 1 \pmod{n}$, то n — составное.
4. Если $a^{n-1} \equiv 1 \pmod{n}$, то n вероятно простое.

Сравнение:

- **Плюсы:** Простой и быстрый.
- **Минусы:** Подвержен “числам Кармайкла” составным числам, которые проходят тест для всех a .

Символ Якоби

Символ Якоби — это обобщение символа Лежандра на случай, когда знаменатель является нечетным составным числом. Символ Якоби $\left(\frac{a}{n}\right)$ определяется для

целого числа a и нечетного натурального числа n .

Если n — простое число, то символ Якоби совпадает с символом Лежандра. Символ Лежандра $(\frac{a}{p})$ определяется для целого числа a и простого числа p и указывает, является ли a квадратичным вычетом по модулю p .

Свойства символа Якоби

1. Мультипликативность:

$$(\frac{ab}{n}) = (\frac{a}{n})(\frac{b}{n})$$

2. Симметрия:

$$(\frac{a}{n}) = (\frac{a \bmod n}{n})$$

3. Квадратичный закон взаимности:

Для нечетных натуральных чисел a и b :

$$(\frac{a}{b})(\frac{b}{a}) = (-1)^{\frac{(a-1)(b-1)}{4}}$$

4. Свойства для $a = -1$ и $a = 2$:

$$(\frac{-1}{n}) = (-1)^{\frac{n-1}{2}}$$

$$(\frac{2}{n}) = (-1)^{\frac{n^2-1}{8}}$$

Тест Соловея-Штрассена

Тест Соловея-Штрассена использует символ Якоби и малую теорему Ферма для определения вероятности простоты числа.

Алгоритм действий:

1. Выбрать случайное число a такое, что $1 \leq a < n$.
2. Вычислить символ Якоби $\left(\frac{a}{n}\right)$.
3. Вычислить $a^{(n-1)/2} \bmod n$.
4. Если $\left(\frac{a}{n}\right) \not\equiv a^{(n-1)/2} \bmod n$, то n — составное.
5. Если $\left(\frac{a}{n}\right) \equiv a^{(n-1)/2} \bmod n$, то n вероятно простое.

Сравнение:

- **Плюсы:** Более надежный, чем тест Ферма, так как не подвержен “числам Кармайкла”.
- **Минусы:** Требуется вычисления символа Якоби, что может быть сложнее.

Тест Миллера-Рабина

Тест Миллера-Рабина — это вероятностный тест, основанный на расширении малой теоремы Ферма и использующий свойства квадратичных вычетов.

Алгоритм действий:

1. Представить $n - 1$ как $2^s \cdot d$, где d — нечетное.
2. Выбрать случайное число a такое, что $1 \leq a < n$.
3. Вычислить $a^d \bmod n$.
4. Если $a^d \equiv 1 \bmod n$ или $a^d \equiv -1 \bmod n$, то n вероятно простое.
5. Иначе, вычислить $a^{2^r \cdot d} \bmod n$ для $r = 1, 2, \dots, s - 1$.

6. Если для какого-то r выполняется $a^{2^r \cdot d} \equiv -1 \pmod n$, то n вероятно простое.
7. Если ни одно из условий не выполняется, то n — составное.

Сравнение:

- **Плюсы:** Один из самых надежных вероятностных тестов, не подвержен “числам Кармайкла”.
- **Минусы:** Требуется больше вычислений, чем тест Ферма.

Вывод:

- **Тест Ферма** — простой, но подвержен “числам Кармайкла”.
- **Тест Соловея-Штрассена** — более надежный, чем Ферма, но требует вычисления символа Якоби.
- **Тест Миллера-Рабина** — самый надежный из трех, но требует больше вычислений.

Каждый из этих тестов дает вероятностный результат, и для подтверждения простоты числа обычно используют несколько итераций теста.

Ход работы

Подготовка

Так для тестирования работы кода я создал простой шаблон генерирующий наше случайное число для проверки, а также необходимые для каждого алгоритма коэффициент или параметр который гарантированно меньше исходного, но не меньше 1.

```
1  # Создание списка для теста
2  tests = Matrix(undef, 0, 2)
3  for j in 1:10
4      p = rand(3:10000)
5      k = rand(1:p)
6      tests = vcat(tests, [p, k]')
7  end
8  tests
```



```
10×2 Matrix{Any}:
 3846  3655
 9603  4874
 2116  1347
   288     2
     5     3
    55     8
 3379  1591
 6637  2043
 8752  1625
   67    54
```

Рис. 1: Случайная тестовая группа

Алгоритм теста Ферма

```
1 # 1. Алгоритм теста Ферма
2 function fermat_test(n::Int, k::Int=5)
3     if n <= 1
4         return false
5     elseif n <= 3
6         return true
7     end
8
9     for _ in 1:k
10         a = rand(2:n-2)
11         if powermod(a, n-1, n) != 1
12             return false
13         end
14     end
15     return true
16 end
17
18 println("Тест Ферма:")
19 for i in 1:size(tests)[1]
20     t = fermat_test(tests[i, 1], tests[i, 2])
21     println("Число ", tests[i, 1], t ? "" : " НЕ", " простое.")
22 end
```

Рис. 2: Код теста Ферма

Результат 1

```
Тест Ферма:  
Число 3846 НЕ простое.  
Число 9603 НЕ простое.  
Число 2116 НЕ простое.  
Число 288 НЕ простое.  
Число 5 простое.  
Число 55 НЕ простое.  
Число 3379 НЕ простое.  
Число 6637 простое.  
Число 8752 НЕ простое.  
Число 67 простое.
```

Рис. 3: Результат шифра Цезаря

Алгоритм вычисления символа Якоби

Если посмотреть на полученные результаты может показаться что при данных значениях нет ни одного числа которое при использовании теста Соловья-Штрассена. Для проверки рассмотрим случай номер 5 при значениях $n = 5$ (число для проверки) и $a = 3$ (случайное число) оба при этом оказались простыми.

1. Начнём с символа Якоби: $3/5$; 5 - простое и совпадает с символом Лежандра который вычисляем как $5/3 \Rightarrow 2/3$ а, так как $2 = \text{mod}(3)$ и не является квадратом по модулю 3 то $2/3 = -1 \Rightarrow 3/5 \Rightarrow -1$
2. Теперь вычислим $a^{(n-1)/2} * \text{mod } n \Rightarrow 3^{(5-1)/2} * \text{mod } 5 \Rightarrow 3^2 * \text{mod } 5 = 4$
3. Сравним: -1 и 4 не равны значит чисто по тесту Соловья-Штрассена при $a = 3$ и $n = 5$, n не простое. При других значениях a возможно он даст правильный ответ, но не здесь.

```

1  # 2. Алгоритм вычисления символа Якоби
2  function jacobi_symbol(n::Int, a::Int=5)
3      if n < 0 || iseven(n)
4          return "Не подходит n четное."
5      end
6      a %= n
7      g = 1
8
9      while a != 0
10         while iseven(a)
11             a >>= 1
12             if (n % 8) in [3, 5]
13                 g = -g
14             end
15         end
16         a, n = n, a
17         if a % 4 == 3 && n % 4 == 3
18             g = -g
19         end
20         a %= n
21     end
22     return n == 1 ? g : 0
23 end
24
25 println("Символ Якоби:")
26 for i in 1:size(tests)[1]
27     t = jacobi_symbol(tests[i, 1])
28     println("При n = ", tests[i, 1], " и a = ", tests[i, 2], " символ Якоби = ", t)
29 end

```

Рис. 4: Код вычисления символа Якоби

Результат 2

Символ Якоби:

При $n = 3846$ и $a = 3655$ символ Якоби = Не подходит n четное.

При $n = 9603$ и $a = 4874$ символ Якоби = -1

При $n = 2116$ и $a = 1347$ символ Якоби = Не подходит n четное.

При $n = 288$ и $a = 2$ символ Якоби = Не подходит n четное.

При $n = 5$ и $a = 3$ символ Якоби = 0

При $n = 55$ и $a = 8$ символ Якоби = 0

При $n = 3379$ и $a = 1591$ символ Якоби = 1

При $n = 6637$ и $a = 2043$ символ Якоби = -1

При $n = 8752$ и $a = 1625$ символ Якоби = Не подходит n четное.

При $n = 67$ и $a = 54$ символ Якоби = -1

Рис. 5: Результат вычисления символа Якоби

Алгоритм теста Соловья-Штрассена

```
1  # 3. Алгоритм теста Соловья-Штрассена
2  function solovay_strassen__test(n::Int, k::Int=5)
3      if n <= 1
4          return false
5      elseif n <= 3
6          return true
7      end
8
9      for _ in 1:k
10         a = rand(2:n-2)
11         x = jacobi_symbol(n, a)
12         if typeof(x) == String
13             return x
14         end
15         if x == 0 || powermod(a, (n-1) ÷ 2, n) != x % n
16             return false
17         end
18     end
19     return true
20 end
21
22 println("Тест Соловья-Штрассена:")
23 for i in 1:size(tests)[1]
24     t = solovay_strassen__test(tests[i, 1], tests[i, 2])
25     println("При n = ", tests[i, 1], " и k = ", tests[i, 2], ": ", t)
26 end
```

Рис. 6: Код теста Соловья-Штрассена

Результат 3

```
Тест Соловья-Штрассена:  
При n = 3846 и k = 3655: Не подходит n четное.  
При n = 9603 и k = 4874: false  
При n = 2116 и k = 1347: Не подходит n четное.  
При n = 288 и k = 2: Не подходит n четное.  
При n = 5 и k = 3: false  
При n = 55 и k = 8: false  
При n = 3379 и k = 1591: false  
При n = 6637 и k = 2043: false  
При n = 8752 и k = 1625: Не подходит n четное.  
При n = 67 и k = 54: false
```

Рис. 7: Результат теста Соловья-Штрассена

Алгоритм теста Миллера-Рабина

```
1  # 4. Алгоритм теста Миллера-Рабина
2  function miller_rabin_test(n::Int, k::Int=5)
3      if n <= 1
4          return false
5      elseif n <= 3
6          return true
7      elseif iseven(n)
8          return false
9      end
10
11     # Разложение n-1 на d * 2^r
12     r, d = 0, n - 1
13     while iseven(d)
14         d >>= 1
15         r += 1
16     end
17
18     for _ in 1:k
19         a = rand(2:n-2)
20         x = powermod(a, d, n)
21         if x == 1 || x == n - 1
22             continue
23         end
24
25         for _ in 1:(r-1)
26             x = powermod(x, 2, n)
27             if x == n - 1
28                 break
29             end
30         end
31     end
```

Рис. 8: Код теста Миллера-Рабина 1

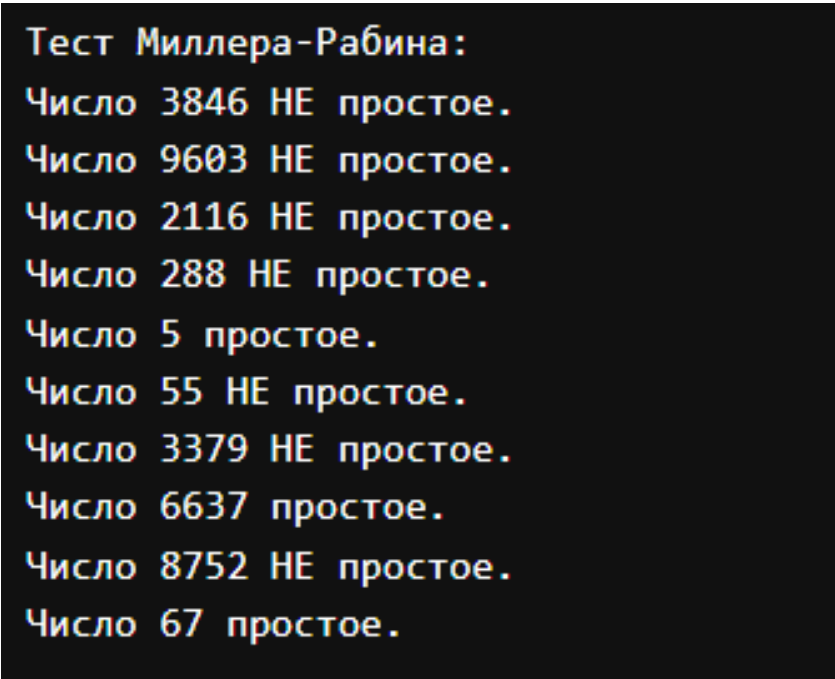

```

30     end
31
32     if x != n - 1
33         return false
34     end
35 end
36 return true
37 end
38
39 println("Тест Миллера-Рабина:")
40 for i in 1:size(tests)[1]
41     t = miller_rabin_test(tests[i, 1], tests[i, 2])
42     println("Число ", tests[i, 1], t ? "" : " НЕ", " простое.")
43 end

```

Рис. 9: Код теста Миллера-Рабина 2

Результат 4

A screenshot of a terminal window with a black background and yellow text. The text displays the results of the Miller-Rabin primality test for various numbers.

Тест Миллера-Рабина:
Число 3846 НЕ простое.
Число 9603 НЕ простое.
Число 2116 НЕ простое.
Число 288 НЕ простое.
Число 5 простое.
Число 55 НЕ простое.
Число 3379 НЕ простое.
Число 6637 простое.
Число 8752 НЕ простое.
Число 67 простое.

Рис. 10: Результат теста Миллера-Рабина

Выводы по проделанной работе

Вывод

В ходе выполнения лабораторной работе были изучены такие способы определение простоты числа как алгоритм теста Ферма, алгоритм теста Миллера-Рабина и алгоритм теста Соловья-Штрассена, и алгоритм вычисления символа Якоби.

Список литературы

1. Свойства символа Якоби и Тест на простоту Соловея-Штрассена
2. Метод Соловея -Штрассена для проверки на простоту
3. Primality Test | Set 3 (Miller–Rabin)
4. Тесты Ферма и Миллера-Рабина на простоту
5. Алгоритм Соловея-Штрассена