

Математические основы защиты информации и информационной безопасности. Лабораторная работа №5

Вероятностные алгоритмы проверки чисел на простоту

Студент: Лесков Данила Валерьевич НФИмд-02-21

Преподаватель: Кулябов Дмитрий Сергеевич

Содержание

1	Цель работы	5
2	Теоретические сведения	6
2.1	Тест Ферма	7
2.2	Тест Соловья-Штрассена	7
2.3	Тест Миллера-Рабина	8
3	Выполнение работы	9
3.1	Реализация алгоритмов	9
3.2	Пример работы алгоритма Ферма	13
3.3	Пример работы алгоритма Соловья-Штрассена	13
3.4	Пример работы алгоритма Миллера-Рабина	14
4	Выводы	15
	Список литературы	16

List of Figures

3.1	Пример работы алгоритма Ферма	13
3.2	Пример работы алгоритмов для Соловья-Штрассена	14
3.3	Пример работы алгоритмов для Миллера-Рабина	14

List of Tables

1 Цель работы

Изучение алгоритмов Ферма, Соловья-Штрассена, Миллера-Рабина.

2 Теоретические сведения

Для построения многих систем защиты информации требуются простые числа большой разрядности. В связи с этим актуальной является задача тестирования на простоту натуральных чисел.

Существует два типа критериев простоты: детерминированные и вероятностные. Детерминированные тесты позволяют доказать, что тестируемое число - простое. Практически применимые детерминированные тесты способны дать положительный ответ не для каждого простого числа, поскольку используют лишь достаточные условия простоты. Детерминированные тесты более полезны, когда необходимо построить большое простое число, а не проверить простоту, скажем, некоторого единственного числа. В отличие от детерминированных, вероятностные тесты можно эффективно использовать для тестирования отдельных чисел, однако их результаты, с некоторой вероятностью, могут быть неверными. К счастью, ценой количества повторений теста с модифицированными исходными данными вероятность ошибки можно сделать как угодно малой. На сегодня известно достаточно много алгоритмов проверки чисел на простоту. Несмотря на то, что большинство из таких алгоритмов имеет субэкспоненциальную оценку сложности, на практике они показывают вполне приемлемую скорость работы. На практике рассмотренные алгоритмы чаще всего по отдельности не применяются. Для проверки числа на простоту используют либо их комбинации, либо детерминированные тесты на простоту. Детерминированный алгоритм всегда действует по одной и той же схеме и гарантированно решает поставленную задачу. Вероятностный алгоритм использует генератор

случайных чисел и дает не гарантированно точный ответ. Вероятностные алгоритмы в общем случае не менее эффективны, чем детерминированные (если используемый генератор случайных чисел всегда дает набор одних и тех же чисел, возможно, зависящих от входных данных, то вероятностный алгоритм становится детерминированным).

2.1 Тест Ферма

- Вход. Нечетное целое число $n \geq 5$.
 - Выход. «Число n , вероятно, простое» или «Число n составное».
1. Выбрать случайное целое число a , $2 \leq a \leq n - 2$.
 2. Вычислить $r = a^{n-1} \pmod n$
 3. При $r = 1$ результат: «Число n , вероятно, простое». В противном случае результат: «Число n составное».

Подробнее об алгоритме: [1]

2.2 Тест Соловья-Штрассена

- Вход. Нечетное целое число $n \geq 5$.
 - Выход. «Число n , вероятно, простое» или «Число n составное».
1. Выбрать случайное целое число a , $2 \leq a \leq n - 2$.
 2. Вычислить $r = a^{\left(\frac{n-1}{2}\right)} \pmod n$
 3. При $r \neq 1$ и $r \neq n - 1$ результат: «Число n составное».
 4. Вычислить символ Якоби $s = \left(\frac{a}{n}\right)$
 5. При $r = s \pmod n$ результат: «Число n , вероятно, простое». В противном случае результат: «Число n составное».

Подробнее об алгоритме: [2]

2.3 Тест Миллера-Рабина

- Вход. Нечетное целое число $n \geq 5$.
 - Выход. «Число n , вероятно, простое» или «Число n составное».
1. Представить $n - 1$ в виде $n - 1 = 2^s r$, где r - нечетное число
 2. Выбрать случайное целое число a , $2 \leq a \leq n - 2$.
 3. Вычислить $y = a^r \pmod{n}$
 4. При $y \neq 1$ и $y \neq n - 1$ выполнить действия
 - Положить $j = 1$
 - Если $j \leq s - 1$ и $y \neq n - 1$ то
 - Положить $y = y^2 \pmod{n}$
 - При $y = 1$ результат: «Число n составное».
 - Положить $j = j + 1$
 - При $y \neq n - 1$ результат: «Число n составное».
 5. Результат: «Число n , вероятно, простое».

Подробнее об алгоритме: [3]

3 Выполнение работы

3.1 Реализация алгоритмов

```
import random
```

```
def fermat(n):  
    a = random.randint(2, n - 2)  
    r = a ** (n - 1) % n  
    return r == 1
```

```
def jacoby(a, n):  
    if (a == 0):  
        return 0  
    result = 1  
    if (a < 0):  
        a = -a  
        if (n % 4 == 3):  
            result = -result  
    if (a == 1):  
        return result  
    while (a):
```

```

    if (a < 0):
        a = -a
        if (n % 4 == 3):
            result = -result
    while (a % 2 == 0):
        a = a // 2
        if (n % 8 == 3 or n % 8 == 5):
            result = -result
    a, n = n, a
    if (a % 4 == 3 and n % 4 == 3):
        result = -result
    a = a % n
    if (a > (n // 2)):
        a = a - n
    if (n == 1):
        return result
    return 0

```

```

def solovay_strassen(n):
    a = random.randint(2, n - 2)
    r = a ** ((n - 1) / 2)
    if (r != 1 and r != (n - 1)):
        return False
    s = jacoby(a, n)
    return not ((r - s) % n == 0)

```

```

def miller_rabin(n):

```

```

a = random.randint(2, n - 2)
d = n - 1
s = 0
while (d % 2 == 0):
    s = s + 1
    d = int(d / 2)
x = a ** d
x = x % n
if (x == 1 or x == (n - 1)):
    return True
r = 1
while (r < (s - 1)):
    x = x ** 2
    x = x % n
    if (x == 1):
        return False
    if (x == (n - 1)):
        return True
return False

if __name__ == '__main__':
    while True:
        try:
            result_code = int(input(
                """
Выберите алгоритм проверки числа на простоту
1 - Алгоритм Ферма;
2 - Алгоритм Соловья-Штрассена;

```

3 - Алгоритм Миллера-Рабина;

0 - Выход из программы

Введите номер операции: ""

```
    ))
    if result_code > 3:
        print("Ошибка ввода!")
        continue
    if result_code == 0:
        break
except:
    print("Ошибка ввода!")
    continue

try:
    n = int(input("Введите нечетное целое число > 5: "))
except:
    print("Ошибка ввода!")
    continue

result = False

if result_code == 1:
    result = fermat(n)

if result_code == 2:
    result = solovay_strassen(n)

if result_code == 3:
```

```

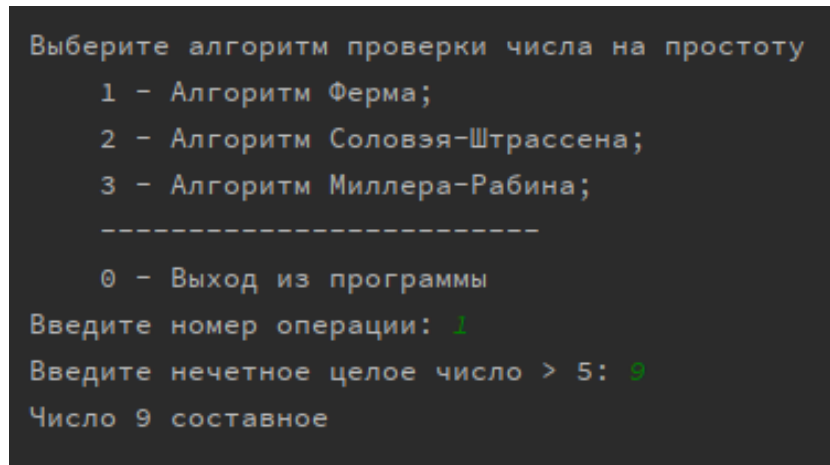
result = miller_rabin(n)

if result:
    print("Число {}, вероятно, простое".format(n))
else:
    print("Число {} составное".format(n))

```

3.2 Пример работы алгоритма Ферма

На рис. 3.1 представлены результаты работы алгоритма Ферма:



```

Выберите алгоритм проверки числа на простоту
  1 - Алгоритм Ферма;
  2 - Алгоритм Соловья-Штрассена;
  3 - Алгоритм Миллера-Рабина;
  -----
  0 - Выход из программы
Введите номер операции: 1
Введите нечетное целое число > 5: 9
Число 9 составное

```

Figure 3.1: Пример работы алгоритма Ферма

3.3 Пример работы алгоритма Соловья-Штрассена

На рис. 3.2 представлены результаты работы алгоритма Соловья-Штрассена:

```
Выберите алгоритм проверки числа на простоту
  1 - Алгоритм Ферма;
  2 - Алгоритм Соловья-Штрассена;
  3 - Алгоритм Миллера-Рабина;
  -----
  0 - Выход из программы
Введите номер операции: 2
Введите нечетное целое число > 5: 9
Число 9 составное
```

Figure 3.2: Пример работы алгоритмов для Соловья-Штрассена

3.4 Пример работы алгоритма Миллера-Рабина

На рис. 3.3 представлены результаты работы алгоритма Миллера-Рабина:

```
Выберите алгоритм проверки числа на простоту
  1 - Алгоритм Ферма;
  2 - Алгоритм Соловья-Штрассена;
  3 - Алгоритм Миллера-Рабина;
  -----
  0 - Выход из программы
Введите номер операции: 3
Введите нечетное целое число > 5: 7
Число 7, вероятно, простое
```

Figure 3.3: Пример работы алгоритмов для Миллера-Рабина

4 Выводы

В ходе выполнения работы были успешно изучены алгоритмы Ферма, Соловья-Штрассена, Миллера-Рабина, а также реализованы данные алгоритмы программно на языке Python.

Список литературы

1. Алгоритм Ферма [Электронный ресурс]. Википедия, 2021. URL: https://ru.wikipedia.org/wiki/Метод_факторизации_Ферма.
2. Алгоритм Соловья-Штрассена [Электронный ресурс]. Википедия, 2020. URL: https://ru.wikipedia.org/wiki/Тест_Соловья_—_Штрассена.
3. Расширенный Миллера-Рабина [Электронный ресурс]. Википедия, 2021. URL: https://ru.wikipedia.org/wiki/Тест_Миллера_—_Рабина.