РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

Отчёт по лабораторной работе №5
Вероятностные алгоритмы проверки чисел на простоту

Дисциплина: *Математические основы защиты информации и информационной безопасности*

Студент: Леонова Алина Дмитриевна, 1032212306
Группа: НФИмд-01-21
Преподаватель: Кулябов Дмитрий Сергеевич,
д-р.ф.-м.н., проф.

Москва 2021

Содержание

List of Figures

1 Цель работы

Целью данной работы является ознакомление и реализация на выбранном языке программирования трёх вероятностных алгоритмов проверки чисел на простоту, а также алгоритма вычисления символа Якоби.

2 Задание

Реализовать программно:

- алгоритм, реализующий тест Ферма
- алгоритм вычисления символа Якоби
- алгоритм, реализующий тест Соловэя-Штрассена
- алгоритм, реализующий тест Миллера-Рабина

3 Теоретическое введение

Тестом простоты (или проверкой простоты) называется алгоритм, который, приняв на входе число N, позволяет либо не подтвердить предположение о составности числа, либо точно утверждать его простоту. Во втором случае он называется истинным тестом простоты [1].

3.1 Тест Ферма

Если n — простое число, то оно удовлетворяет сравнению $a^{n-1} \equiv 1(\ mod\ n)$ для любого a, которое не делится на n.

Выполнение сравнения $a^{n-1} \equiv 1(\ mod\ n)$ является необходимым, но не достаточным признаком простоты числа. То есть, если найдётся хотя бы одно a, для которого $a^{n-1} \not\equiv 1(\ mod\ n)$, то число n — составное; в противном случае ничего сказать нельзя, хотя шансы на то, что число является простым, увеличиваются. Если для составного числа n выполняется сравнение $a^{n-1} \equiv 1(\ mod\ n)$, то число n называют псевдопростым по основанию a [2].

3.2 Символа Якоби

Символ Якоби — теоретико-числовая функция двух аргументов, введённая К. Якоби в 1837 году. Является квадратичным характером в кольце вычетов. Символ Якоби практически никогда не вычисляют по определению. Чаще всего для

вычисления используют свойства символа Якоби, главным образом — квадратичный закон взаимности.

Ключевое используемое при вычислении свойство символа Якоби — квадратичный закон взаимности. Благодаря ему алгоритм похож на алгоритм Евклида нахождения наибольшего общего делителя двух чисел, в котором тоже аргументы на каждом шаге меняются местами. Аналогично алгоритму Евклида, при перестановке аргументов больший заменяется на остаток от деления на меньший. Это возможно благодаря периодичности символа Якоби. Однако, поскольку символ Якоби определён только при условии нечётности второго аргумента, то до перестановки выделяется чётная часть первого аргумента [3].

3.3 Тест Соловэя-Штрассена

Тест Соловея-Штрассена — вероятностный тест простоты, открытый в 1970-х годах Робертом Мартином Соловеем совместно с Фолькером Штрассеном. Тест всегда корректно определяет, что простое число является простым, но для составных чисел с некоторой вероятностью он может дать неверный ответ [4].

Тест Соловея-Штрассена опирается на малую теорему Ферма и свойства символа Якоби $(\frac{a}{n})$:

- Если n — нечетное составное число, то количество целых чисел a, взаимнопростых с n и меньших n, удовлетворяющих сравнению $a^{(n-1)/2} \equiv (\frac{a}{n})(\ mod\ n)$, не превосходит $n/2$.

Составные числа n удовлетворяющие этому сравнению называются псевдопростыми Эйлера-Якоби по основанию a.

Алгоритм Соловея-Штрассена параметризуется количеством раундов k. В каждом раунде случайным образом выбирается число $a < n$. Если $(a, n) > 1$, то выносится решение, что n составное. Иначе проверяется справедливость сравнения $a^{(n-1)/2} \equiv (\frac{a}{n})(\ mod\ n)$. Если оно не выполняется, то выносится решение, что n — составное.

3.4 Тест Миллера-Рабина

Тест Миллера-Рабина — вероятностный полиномиальный тест простоты, позволяет эффективно определить, является ли данное число составным. Однако, с его помощью нельзя строго доказать простоту числа

Как и тесты Ферма и Соловея-Штрассена, тест Миллера-Рабина опирается на проверку ряда равенств, которые выполняются для простых чисел. Если хотя бы одно такое равенство не выполняется, это доказывает что число составное [5].

Для теста Миллера-Рабина используется следующее утверждение:

Пусть n — простое число и $n - 1 = 2^s d$, где d — нечётно. Тогда для любого a выполняется хотя бы одно из условий:

- $a^d \equiv 1(\ mod\ n)$
- Существует целое число $r < s$ такое что $a^{2^r d} \equiv -1(\ mod\ n)$

4 Выполнение лабораторной работы

4.1 Функция для проверки

Импорт используемых библиотек:

```
import random
```

4.2 Алгоритм, реализующий тест Ферма

Первым делом проверяю корректность входных чисел, а просле следую алгоритму из задания.

```
def test_Ferma(n):
    print('')
    if n % 2 == 0:
        return print('Ошибка: число ', n,' чётное')
    if n < 5:
        return print('Ошибка: число ', n,' < 5')

    a = random.randint(2, n - 2)
    r = ( a ** (n - 1) ) % n

    if r == 1:
        return print('Число ', n,' , вероятно, простое')
```

```
    else:
        return print('Число ', n,' составное')
```

4.3 Алгоритм вычисления символа Якоби

Сперва проверяю корректность входных чисел, а просле следую алгоритму из задания. Реализован с помощью использования бесконечного цикла.

```
def Yakobi(n, a):
    print('# символ Якоби (', a,'/',n,')')
    if n % 2 == 0:
        return print('Ошибка: число ', n,' чётное')
    if n < 3:
        return print('Ошибка: число n (', n,') < 3')
    if a < 0 or a >= n:
        return print('Ошибка: число a (', a,') некорректно')

    g = 1

    while True:
        if a == 0:
            return 0
        if a == 1:
            return g

        k = 0
        while a % (2**k) == 0:
            k += 1
        k -= 1
```

```
    a1 = a / (2**k)
    #print(a, ' = 2 ^', k, ' * ', a1)

    s = 0
    if k % 2 == 0:
        s = 1
    else:
        if (n - 1) % 8 == 0 or (n + 1) % 8 == 0:
            s = 1
        if (n - 3) % 8 == 0 or (n + 3) % 8 == 0:
            s = -1

    if a1 == 1:
        return g*s

    if (n - 3) % 4 == 0 and (a1 - 3) % 4 == 0:
        s = -s

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

4.4 Алгоритм, реализующий тест Соловэя-Штрассена

Первым делом проверяю корректность входных чисел, а просле следую алгоритму из задания. Симпола Якоби вычисляется вызовом реализованной ранее функции Yakobi. В алгоритме из задания исправлена ошибка на 5 шаге.

```
# 3. Тест Соловэя-Штрассена
def test_Sol_Shtr(n):
```

```
    print('')
    if n % 2 == 0:
        return print('Ошибка: число ', n,' чётное')
    if n < 5:
        return print('Ошибка: число ', n,' < 5')

    a = random.randint(2, n - 3)
    r = ( a ** ((n - 1)/2) ) % n

    if r != 1 and r != n-1:
        return print('Число ', n,' составное')

    s = Yakobi(n, a)

    if (r - s) % n != 0:
        return print('Число ', n,' составное')
    else:
        return print('Число ', n,' , вероятно, простое')
```

4.5 Алгоритм, реализующий тест Миллера-Рабина

Сперва проверяю корректность входных чисел, а просле следую алгоритму из задания.

```
# 4. Тест Миллера-Рабина
def test_Mil_Rab(n):
    print('')
    if n % 2 == 0:
        return print('Ошибка: число ', n,' чётное')
    if n < 5:
```

```
        return print('Ошибка: число ', n,' < 5')

    s = 0
    while n - 1 % (2**s) == 0:
        s += 1
    s -= 1
    r = (n - 1) / (2**s)
    #print(n - 1, ' = 2 ^', s, ' * ', r)

    a = random.randint(2, n - 3)
    y = ( a ** r ) % n

    if y != 1 and y != n-1:
        j = 1
        while j <= s - 1 and y != n - 1:
            y = (y*y) % n
            if y == 1:
                return print('Число ', n,' составное')
            j += 1
        if y != n - 1:
            return print('Число ', n,' составное')

    return print('Число ', n,' , вероятно, простое')
```

4.6 Функция для проверки

Функция для проверки функций тестов на 5 целых числах:

```
# Функция проверки функций тестов
def check(f):
```

```
    f(1)
    f(1122)
    f(11)
    f(27)
    f(31)
```

4.7 Проверка

Вызов проверок работы всех реализованных функций на пяти разных вариантах входных параметров, задаваемых в функции check:

```
print('Тест Ферма')
check(test_Ferma)

print('\n-----------------------------------')
print('Символ Якоби')
print('Результат:', Yakobi(27, 5))
print('Результат:', Yakobi(27, 12))
print('Результат:', Yakobi(51, 13))

print('\n-----------------------------------')
print('Тест Соловэя-Штрассена')
check(test_Sol_Shtr)

print('\n-----------------------------------')
print('Тест Миллера-Рабина')
check(test_Mil_Rab)
```

```
GitHub/1.2-IS/Lab_5')
Тест Ферма

Ошибка: число  1  < 5

Ошибка: число  1122  чётное

Число  11  , вероятно, простое

Число  27  составное

Число  31  , вероятно, простое

----------------------------------
Символ Якоби
# символ Якоби ( 5 / 27 )
Результат: -1
# символ Якоби ( 12 / 27 )
Результат: 0
# символ Якоби ( 13 / 51 )
Результат: 1

----------------------------------
Тест Соловэя-Штрассена

Ошибка: число  1  < 5

Ошибка: число  1122  чётное

# символ Якоби ( 3 / 11 )
Число  11  , вероятно, простое

Число  27  составное

# символ Якоби ( 28 / 31 )
Число  31  , вероятно, простое

----------------------------------
Тест Миллера-Рабина

Ошибка: число  1  < 5

Ошибка: число  1122  чётное

Число  11  , вероятно, простое

Число  27  составное

Число  31  , вероятно, простое

In [63]:
```

Figure 4.1: Результат выполнения L5_Leonova.py

Результат выполнения программы, проверка реализации 3-ех вероятностных алгоритмов проверки чисел на простоту, а также алгоритма вычисления символа Якоби (см. рис. 4.1). Поскольку алгоритмы вероятностные, в процессе генерируется случайное a, реультаты последних двух алгоритмов на одних числах не всегда совпадают.

5 Выводы

Цель лабораторной работы была достигнута, три вероятностных алгоритмов проверки чисел на простоту и алгоритм вычисления символа Якоби были реализованы на языке программирования Python.

Список литературы

1. Тест простоты [Электронный ресурс]. Википедия, 2020. URL: https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D0%BE%D1%82%D1%8B.
2. Тест Ферма [Электронный ресурс]. Википедия, 2020. URL: https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82_%D0%A4%D0%B5%D1%80%D0%BC%D0%B0.
3. Символ Якоби [Электронный ресурс]. Википедия, 2020. URL: https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB_%D0%AF%D0%BA%D0%BE%D0%B1%D0%B8.
4. Тест Соловея — Штрассена [Электронный ресурс]. Википедия, 2020. URL: https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82_%D0%A1%D0%BE%D0%BB%D0%BE%D0%B2%D0%B5%D1%8F_%E2%80%94_%D0%A8%D1%82%D1%80%D0%B0%D1%81%D1%81%D0%B5%D0%BD%D0%B0.
5. Тест Миллера — Рабина [Электронный ресурс]. Википедия, 2021. URL: https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82_%D0%9C%D0%B8%D0%BB%D0%BB%D0%B5%D1%80%D0%B0_%E2%80%94_%D0%A0%D0%B0%D0%B1%D0%B8%D0%BD%D0%B0.