Отчёт по лабораторной работе №5

Вероятностные алгоритмы проверки чисел на простоту

Волкова Дарья Александровна НПМмд-02-21

Содержание

Цель работы

Изучение вероятностных алгоритмов проверки чисел на простоту, а также их программная реализация.

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).

Тест Ферма

Вход. Нечетное целое цисло n>=5.

Выход. “Число n, вероятно, простое” или “Число n составное”.

1. Выбрать случайное целое число a, 2<=a<=2.
2. Вычислить r = $a^{(n-1)}$(mod n).
3. Если r = 0 результат : “Число n, вероятно, простое”. В противном случае результат: “Число n составное”.

Символ Якоби

Вход. Нечетное целое цисло n>=3, целое число a, 0<=a<n.

Выход. Символ Якоби.

1. Положить g=1.
2. При a=0 результат: 0.
3. При a=1 результат: g.
4. Предствить a в виде a = 2^k*u , где число u нечетное.
5. При четном k положить s=1, при нечетном положить s=1, если n=abs(1(mod8)); положить s=-1, если n=abs(3(mod8)).
6. При u=1 результат: g*s.

7. Если n=3(mod4) и u = 3(mod4), то s=-s

8. Положить a=n mod(a1), n=a1, g = g*s и вернуться на шаг 2.

Тест Соловэя-Штрассена

Вход. Нечетное целое цисло n>=5.

Выход. “Число n, вероятно, простое” или “Число n составное”.

1. Выбрать случайное целое число a, 2<=a<=2.

2. Вычислить r=a^(n+1)/2(mod n).

3. Если r не равен 1 и не равен n-1 реузультат: “Число n составное”.

4. Вычислить символ Якоби s=(a/n).

5. Если r=s(mod n) реузультат: “Число n составное”. В противном случае результат: “Число n, вероятно, простое”.

Тест Миллера-Рабина

Вход. Нечетное целое цисло n>=5.

Выход. “Число n, вероятно, простое” или “Число n составное”.

1. Представить n-1 в виде n-1=2^s*r , где r нечетное.

2. Выбрать случайное целое число a, 2<=a<=2.

3. Вычислить y=a^r(mod n).

4. При y не равном 1 и не равном n-1 выполнить следующие действия:

4.1. Положить j = 1.

4.2. если j<=s-1 и y не равен n-1 ,то положить y=y^2(mod n). При y=1 результат: “Число n составное”. Положить j=j+1.

4.3. При y не равном n-1 результат: “Число n составное”.

5.Результат: “Число n, вероятно, простое”.

Выполнение работы

Реализация алгоритмов на языке Python

```
# 1. Тест Ферма

n=int(input('Введите нечетное число n больше или равно 5: '))

import random
```

```
a=random.randint(2, n-2)

r=(a**(n-1))%n

if r==1:
    print('Число n =', n, ', вероятно, простое')
else:
    print('Число n =', n, 'составное')

# 2. Символ Якоби

n=int(input('Введите нечетное число n больше или равно 3: '))
a=int(input('Введите число a больше или равно 0 и меньше n: '))

def jacobi(n,a):
    g=1
    while True:
        if a == 0:
            return 0
        if a == 1:
            return g
        else:
            k=0
            a1=a
            while a1%2 == 0:
                k+=1
                a1//=2
            if k%2==0:
                s=1
            else:
                if abs(n%8)==1:
                    s=1
                else:
                    s=-1
            if a1==1:
                return g*s
            if n%4==3 and a1%4 == 3:
                s*=-1
            a = n%a1
            n = a1
            g = g*s

print('Символ Якоби=', jacobi(n,a))

# 3. Тест Соловэя-Штрассена

n=int(input('Введите нечетное число n больше или равно 5: '))

a=random.randint(2, n-2)

r=(a**((n-1)/2))%n
```

```
if r!=1 and r!=n-1:
    print('Число n =', n, 'составное')
s=jacobi(n,a)
if s==r%n:
    print('Число n =', n, 'составное')
else:
    print('Число n =', n, ', вероятно, простое')

# 4. Тест Миллера-Рабина

n=int(input('Введите нечетное число n больше или равно 5: '))

s=0
r=n-1
while r%2 == 0:
    s+=1
    r//=2

a=random.randint(2, n-2)

y = (a**r)%n

if y!=1 and y != n-1:
    j=1
    if j<=s-1 and y!=n-1:
        y = (y**2)%n
        if y==1:
            print('Число n =', n, 'составное')
        j+=1
    if y!=n-1:
        print('Число n =', n, 'составное')
else:
    print('Число n =', n, ', вероятно, простое')
```

Контрольный пример

```
# 1. Тест Ферма
```

```
n=int(input('Введите нечетное число n больше или равно 5: '))
```

```
Введите нечетное число n больше или равно 5: 27
```

```
import random
```

```
a=random.randint(2, n-2)
```

```
r=(a**(n-1))%n
```

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

```
Число n = 27 составное
```

Пример работы алгоритма тест Ферма

```
# 2. Символ Якоби
```

```
n=int(input('Введите нечетное число n больше или равно 3: '))
a=int(input('Введите число a больше или равно 0 и меньше n: '))
```

```
Введите нечетное число n больше или равно 3: 15
Введите число a больше или равно 0 и меньше n: 7
```

```
def jacobi(n,a):
    g=1
    while True:
        if a == 0:
            return 0
        if a == 1:
            return g
        else:
            k=0
            a1=a
            while a1%2 == 0:
                k+=1
                a1//=2
            if k%2==0:
                s=1
            else:
                if abs(n%8)==1:
                    s=1
                else:
                    s=-1
            if a1==1:
                return g*s
            if n%4==3 and a1%4 == 3:
                s*=-1
            a = n%a1
            n = a1
            g = g*s
```

```
print('Символ Якоби=', jacobi(n,a))
```

```
Символ Якоби= -1
```

Пример работы алгоритма символ Якоби

```
# 3. Тест Соловэя-Штрассена
```

```
n=int(input('Введите нечетное число n больше или равно 5: '))
```

```
Введите нечетное число n больше или равно 5: 27
```

```
a=random.randint(2, n-2)
```

```
r=(a**((n-1)/2))%n
```

```
if r!=1 and r!=n-1:
    print('Число n =', n, 'составное')
s=jacobi(n,a)
if s==r%n:
    print('Число n =', n, 'составное')
else:
    print('Число n =', n, ', вероятно, простое')
```

```
Число n = 27 составное
```

Пример работы алгоритма тест Соловэя-Штрассена

```
# 4. Тест Миллера-Рабина
```

```
n=int(input('Введите нечетное число n больше или равно 5: '))
```

```
Введите нечетное число n больше или равно 5: 27
```

```
s=0
r=n-1
while r%2 == 0:
    s+=1
    r//=2
```

```
a=random.randint(2, n-2)
```

```
y = (a**r)%n
```

```
if y!=1 and y != n-1:
    j=1
    if j<=s-1 and y!=n-1:
        y = (y**2)%n
        if y==1:
            print('Число n =', n, 'составное')
        j+=1
    if y!=n-1:
        print('Число n =', n, 'составное')
else:
    print('Число n =', n, ', вероятно, простое')
```

```
Число n = 27 составное
```

Пример работы алгоритма тест Миллера-Рабина

Выводы

В ходе выполнения работы удалось изучить вероятностные алгоритмы проверки чисел на простоту, такие как тест Ферма, тест Соловэя-Штрассена, тест Миллера-Рабина, а также реализовать данные алгоритмы программно на языке Python.

Список литературы

1. Алгоритмы поиска простых чисел
2. Лекция 2: Алгоритмы тестирования на простоту и факторизации
3. Проверка чисел на простоту
4. Символ Якоби