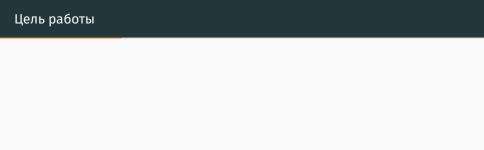
Лабораторная работа №5. Вероятностные алгоритмы проверки чисел на простоту.

Alexander S. Baklashov

10 Novvember, 2023

RUDN University, Moscow, Russian Federation

Цель работы



Рассмотреть и реализовать алгоритмы проверки чисел на простоту.

Задачи

Задачи

Реализовать следующие алгоритмы:

- Тест Ферма;
- Нахождение символа Якоби;
- Тест Соловэя-Штрассена;
- Тест Миллера-Рабина.

Тест Ферма

Реализуем тест Ферма

```
n=5
a= np.random.randint (2,n-1)
r=pow(a,n-1) % n
if (r==1):
    print ("Число, вероятно, простое")
else:
    print ("Число, вероятно, составное")
Число, вероятно, простое
```

Figure 1: Тест Ферма

Нахождение символа Якоби

Найдём символ Якоби

```
[3]: n=11
     a= np.random.randint (0,n)
     print ("a =",a)
     g=1
     d=1
     def representation(n):
         k = 0
         a1 = n
         while a1 % 2 == 0:
             k += 1
             a1 //= 2
         return k, al
     while (d==1):
         if (a--0):
             print ("Символ Якоби =".0)
         if (a==1):
             print ("Символ Якоби =",g)
             break
         k, a1 = representation(a)
         if (k%2==0):
             s=1
         if (k%2!=0):
             if (n%8==1 or n%8==-1):
                 s=1
             if (n%8==3 or n%8==-3):
                 s=-1
         if (a1--1):
             print ("Символ Якоби =", g*s)
             break
         if (n%4--3 and a1%4--3):
             5=-5
         a=n%a1
         n=a1
         g=g*s
      a = 4
     Символ Якоби = 1
```

Figure 2: Нахождение символа Якоби

Тест Соловэя-Штрассена

Реализуем тест Соловэя-Штрассена

```
n=11
a= np.random.randint (2,n-2)
r=pow(a,(n-1)/2) % n
if (r!=1) and (r!=n-1):
    print ("Число ", n," составное")
else:
    s = a/n
    if (r==s%n):
        print ("Число ", n," составное")
else:
        print ("Число ", n," составное")

число 11 вероятно, простое
```

Figure 3: Тест Соловэя-Штрассена

Тест Миллера-Рабина

Реализуем тест Миллера-Рабина

```
def representation(n):
    s = 0
   r = n - 1 # Начнем с максимально возможного нечётного r, который равен n - 1
    while r % 2 == 0:
        r //= 2
       s += 1
    return s, r
n = 13
s, r = representation(n)
a= np.random.randint (2,n-2)
v=pow(a,r) % n
while (y!=1 \text{ and } y!=n-1):
   j=1
    if (j<=n-1 and y!=n-1):
       y=(y*y) %n
        if (v==1):
           print ("Число ",n," составное")
            raise SystemExit("Stop right there!")
        j+=1
    if (v != n-1):
       print ("Число ",n," составное")
        raise SystemExit("Stop right there!")
print ("Число ",n," простое")
Число 13 простое
```

Figure 4: Тест Миллера-Рабина

Вывод

В ходе данной лабораторной работы я рассмотрел и реализовал следующие алгоритмы:

- Тест Ферма;
- Нахождение символа Якоби;
- Тест Соловэя-Штрассена;
- Тест Миллера-Рабина.