

Вероятностные алгоритмы проверки чисел на простоту

Гаджиев Нурсултан Тофик оглы ¹

2022 Moscow, Russia

¹RUDN University, Moscow, Russian Federation

Цель работы

Реализация алгоритмов Ферма, Соловья-Штрассена, Миллера-Рабина и вычисления Якоби.

Задачи

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

Реализация

Функция `ferma` для алгоритма ферма. (рис. 1)

```
-  
3 ▼ def ferma(n):  
4     print("Теста Ферма")  
5     a = random.randint(2, n - 2)  
6     r = a ** (n - 1) % n  
7 ▼   if r==1:  
8       print("Число n, вероятно, простое")  
9 ▼   else:  
10        print("Число n составное")  
11  
12 n = int(input("enter n(Odd number): "))  
13 ferma(n)  
14
```

Figure 1: Функция для алгоритма ферма

Функция `modul` для вычисления бинарного эксп. (рис. 2)

```
15 # функция для бинарного эксп
16 ▼ def modul(base, exponent, mod):
17     x = 1
18     y = base
19 ▼     while (exponent > 0):
20 ▼         if (exponent % 2 == 1):
21             x = (x * y) % mod
22
23             y = (y * y) % mod
24             exponent = exponent // 2
25
26     return x % mod
27
```

Figure 2: Функция для вычисления бинарного эксп

Реализация алгоритма вычисления Якоби.

Функция jacobian для вычисления Якоби. (рис. 3)

```
29 ▼ def jacobian(a, n):
30 ▼     if (a == 0):
31         return 0
32     ans = 1
33 ▼     if (a < 0):
34         a = -a
35 ▼         if (n % 4 == 3):
36             ans = -ans
37 ▼     if (a == 1):
38         return ans
39 ▼     while (a):
40 ▼         if (a < 0):
41             a = -a
42 ▼             if (n % 4 == 3):
43                 ans = -ans
44 ▼             while (a % 2 == 0):
45                 a = a // 2
46 ▼                 if (n % 8 == 3 or n % 8 == 5):
47                     ans = -ans
48             a, n = n, a
49 ▼             if (a % 4 == 3 and n % 4 == 3):
50                 ans = -ans
51             a = a % n
52 ▼             if (a > n // 2):
53                 a = a - n
54 ▼     if (n == 1):
55         return ans
56     return 0
```

Figure 3: Функция для вычисления Якоби

Функция solovoy для алгоритма Соловья-Штрассена. (рис. 4)

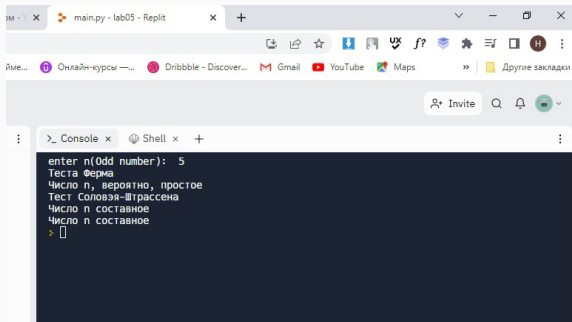
```
58 ▼ def solovoy(n):
59     print("Тест Соловья-Штрассена")
60     a = random.randrange(2,n-2)
61     r= (a**(n-1/2))%n
62 ▼   if (r != 1 and r!=n-1):
63         print("Число n составное")
64
65     s=jacobian(a,n)
66 ▼   if modul(r,s,n) == 1:
67         print( "Число n составное")
68 ▼   else:
69         print("Число n, вероятно, простое")
70
```

Figure 4: Функция для алгоритма Соловья-Штрассена

Функция MillerRabin для алгоритма Миллера-Рабина. (рис. 5)

```
71 def toBinary(n):
72     r = []
73     while (n > 0):
74         r.append(n % 2)
75         n = n / 2
76     return r
77
78 def MillerRabin(n, s = 10):
79
80     for j in range(1, s + 1):
81         a = random.randint(1, n - 1)
82         b = toBinary(n - 1)
83         d = 1
84         for i in range(len(b) - 1, -1, -1):
85             x = d
86             d = (d * d) % n
87             if d == 1 and x != 1 and x != n - 1:
88                 print("Число n составное") # Составное
89             if b[i] == 1:
90                 d = (d * a) % n
91                 if d != 1:
92                     print("Число n составное") # Составное
93                 print("Число n, вероятно, простое")
94
95     solovoy(n)
96     MillerRabin(n)
```

Figure 5: Функция для алгоритма Миллера-Рабина



The screenshot shows a web browser window with a Replit session titled 'main.py - lab05 - Replit'. The browser's address bar and tabs are visible at the top. Below the browser window, there is a terminal window with two tabs: '>_ Console' and 'Shell'. The terminal output shows the execution of a Python script. The prompt 'enter n(odd number):' is followed by the input '5'. The script then prints several lines of Russian text: 'Теста Ферма', 'Число n, вероятно, простое', 'Тест Соловья-Штрассена', 'Число n составное', and 'Число n составное'. The terminal ends with a prompt character '➤' and a cursor.

```
enter n(odd number): 5
Теста Ферма
Число n, вероятно, простое
Тест Соловья-Штрассена
Число n составное
Число n составное
➤
```

Figure 6: Результат алгоритмов

Реализовал алгоритмы Ферма, Соловья-Штрассена, Миллера-Рабина и вычисления Якоби.

Спасибо за внимание