Математические основы защиты информации и информационной безопасности.

Лабораторная работа №5.

Сапёров Максим Александрович.

Содержание

List of Figures

List of Tables

1 Цель работы

Освоить на практике вероятностные алгоримты проверки чисел на простоту

2 Задание

1. Реализовать алгоритм теста Ферма
2. Реализовать вычисления символа Якоби
3. Реализовать алгоритм теста Соловэя-Штрассена
4. Реализовать алгоритм теста Миллера-Рабина

3 Выполнение лабораторной работы

Написал код для вычисления теста Ферма. Все функции реализованы шаблонами:

```
[2]    def ferma(n):
         if n%2!=0 and n>=5:
           None
         else:
           return f'Число {n} составное'
         a = int(2+(n-4)*random())
         r = (a**(n-1))%n
         if r==1:
           return f'Число {n}, вероятно, простое'
         else:
           return f'Число {n} составное'
```

Figure 3.1: Функция теста Ферма

Написал код для вычисления теста Соловэя-Штрассена.

```
[23]   def sol(n):
         if n%2==1 and n>=5:
           None
         else:
           return f'Число {n} составное'
         a = randrange(n-1)+1
         r = modulo(a,(n-1)/2,n)
         s = n+jacobi(n,a)
         if s==0 or r!=s:
           return f'Число {n} составное'
         else:
           return f'Число {n}, вероятно, простое'
```

Figure 3.2: Функция теста Соловэя-Штрассена

Реализовал вычисление теста Миллера-Рабина

```
[30]  def miller(n):
        if n%2!=0 and n>=5:
          None
        else:
          return f'Число {n} составное'
        r, s = 0, n-1
        while s%2==0:
          r+=1
          s//=2
        a = randrange(2, n-1)
        x = pow(a, s ,n)
        if x == 1 or x == n-1:
          None
        for _ in range(r-1):
          x = pow(x, 2, n)
          if x==n-1:
            break
          else:
            return f'Число {n} составное2'
        return f'Число {n}, вероятно, простое'
```

Figure 3.3: тест Миллера-Рабина

Написал вспомогательную функцию вычисления символа Якоби

```
[3]  def jacobi(n,a):
       if n>=3 and n%2!=0 and a>=0 and a<=n:
         None
       else:
         return 'error'
       g = 1
       while a!=0:
         while a%2==0:
           a/=2
           r = n % 8
           if r==3 or r==5:
             g=-g
         a,n = n,a
         if a%4 == n%4 == 3:
           g = -g
         a %= n
       if n==1:
         return g
       else:
         return 0
```

Figure 3.4: Вспомогательная функция вычисления символа Якоби

Результаты тестов.

```
[47]    ferma(71),sol(71),miller(71)
```

```
('Число 71, вероятно, простое',
 'Число 71, вероятно, простое',
 'Число 71, вероятно, простое')
```

Figure 3.5: Результаты тестов

4 Выводы

Освоил на практике вероятностные алгоритмы проверки чисел на простоту