

Лабораторная работа №5

Сапёров Максим Александрович - студент группы НПМмд-02-22

08.01.2022

Вероятностные алгоритмы проверки чисел на простоту

Умение использовать и программировать вероятностные алгоритмы проверки чисел на простоту

Цель выполнения лабораторной работы

Освоить на практике вычисление проверки числа на простоту с помощью вероятности

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

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 1: Функция теста Ферма

Написал код для вычисления теста Соловья-Штрассена.

```
✓ [23] def sol(n):  
0s     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 2: Функция теста Соловья-Штрассена

Реализовал вычисление теста Миллера-Рабина

```
✓ [30] 0s 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: тест Миллера-Рабина

Реализовал вспомогательную функцию вычисления случайного числа

```
✓ [3] def jacobi(n,a):  
0s     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 4: Вспомогательная функция вычисления случайного числа

```
✓ [47]   ferma(71),sol(71),miller(71)
0s
      ('Число 71, вероятно, простое',
       'Число 71, вероятно, простое',
       'Число 71, вероятно, простое')
```

Figure 5: Результаты тестов

Освоил на практике вычисление наибольшего делителя разными способами