# Лабораторная работа №5 Вероятностные алгоритмы проверки чисел на простоту

ВОЛКОВА ДАРЬЯ АЛЕКСАНДРОВНА НПММД-02-21

# Цель работы

Изучение вероятностных алгоритмов проверки чисел на простоту:

1. Тест Ферма
2. Символ Якоби и тест Соловэя-Штрассена
3. Тест Миллера-Рабина

# Задачи

Программная реализация вероятностных алгоритмов проверки чисел на простоту:

1. Тест Ферма
2. Символ Якоби и тест Соловэя-Штрассена
3. Тест Миллера-Рабина

# 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.
- В отличие от детерминированных, вероятностные тесты можно эффективно использовать для тестирования отдельных чисел, однако их результаты, с некоторой вероятностью, могут быть неверными.

# Результаты

```
# 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

# Результаты

```
# 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

# Результаты

```
# 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

# Результаты

```
# 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 :

1. Тест Ферма
2. Символ Якоби и тест Соловэя-Штрассена
3. Тест Миллера-Рабина

А также реализовать данные алгоритмы программно на языке Python.