

Отчет по лабораторной работе 5

Дисциплина: Математические основы защиты информации и информационной безопасности

Дяченко З. К.

12 ноября 2022

Российский университет дружбы народов, Москва, Россия

Данная лабораторная работа выполнялась мной для приобретения практических навыков проверки чисел на простоту.

Ознакомиться и реализовать вероятностные алгоритмы проверки чисел на простоту.

Задачи выполнения лабораторной работы

Реализовать проверку числа на простоту с использованием теста Ферма (рис. - fig. 1).

```
In [13]: import numpy as np
import random
import math

In [20]: def ferma(n):
    if (n>=5 and n%2!=0):
        a=random.randint(2, n-2)
        r=math.pow(a, n-1)%n
        if (r==1):
            return ("Число ", n, ", вероятно, простое")
        else:
            return ("Число ", n, "составное")
    else:
        return ("Введите нечетное число больше или равное 5")

In [21]: ferma(10)

Out[21]: 'Введите нечетное число больше или равное 5'

In [22]: ferma(7)

Out[22]: ('Число ', 7, ', вероятно, простое')

In [23]: ferma(9)

Out[23]: ('Число ', 9, 'составное')
```

Figure 1: Реализация алгоритма, реализующего тест Ферма

Реализовать алгоритм вычисления символа Якоби (рис. - fig. 2 и - fig. 3).

```
In [90]: def yakobi(n, a):
        if (n>=3 and n%2!=0 and 0<=a<n):
            g=1
            while True:
                if (a==0):
                    return 0
                if (a==1):
                    return g
                if (a%2!=0):
                    k=0
                    a1=a
                else:
                    k=0
                    a1=a
                    while (a1%2==0):
                        a1=a1/2
                        k=k+1
                if (k%2==0):
                    s=1
                else:
                    if (n%8==1 or n%8==--1):
                        s=1
                    if (n%8==3 or n%8==--3):
                        s=-1
                if (a1==1):
                    return g*s
                if (n%4==3 and a1%4==3):
                    s=-s
                a=n*a1
                n=a1
                g=g*s
            else:
                return ("Введите нечетное число больше или равно 3 и проверьте, что a больше или равно 0 и меньше введенного числа")
```

Figure 2: Реализация алгоритма вычисления символа Якоби

```
In [91]: yakobi(21, 11)
```

```
Out[91]: -1
```

```
In [92]: yakobi(21, 7)
```

```
Out[92]: 0
```

```
In [75]: yakobi(21, 4)
```

```
Out[75]: 1
```

Figure 3: Пример работы алгоритма вычисления символа Якоби

Задачи выполнения лабораторной работы

Реализовать проверку числа на простоту с использованием теста Соловья-Штрассена (рис. - fig. 4).

```
In [128]: def solshtr(n):  
            if (n>=5 and n%2!=0):  
                a=random.randint(2, n-3)  
                r=math.pow(a, (n-1)/2)%n  
                if (r!=1 and r!=n-1):  
                    return ("число ", n, "составное")  
                else:  
                    ss=yakobi(n, a)  
                    if (r%n==ss):  
                        return ("число ", n, "составное")  
                    else:  
                        return ("число ", n, ", вероятно, простое")  
            else:  
                return ("Введите нечетное число больше или равное 5")
```

```
In [131]: solshtr(13)
```

```
Out[131]: ('число ', 13, ', вероятно, простое')
```

Figure 4: Реализация теста Соловья-Штрассена

Задачи выполнения лабораторной работы

Реализовать проверку числа на простоту с использованием теста Миллера-Рабина (рис. - fig. 5).

```
In [125]: def milrab(n):
          if (n>=5 and n%2!=0):
              s=0
              r=n-1
              while (r%2==0):
                  r=r/2
                  s=s+1
              a=random.randint(2, n-3)
              y=math.pow(a, r)%n
              if (y!=1 and y!=n-1):
                  j=1
                  if (j<=s-1 and y!=n-1):
                      y=math.pow(y,2)%n
                      if (y==1):
                          return ("число ", n, "составное")
                      j=j+1
                  if (y!=n-1):
                      return ("число ", n, "составное")
              return ("число ", n, ", вероятно, простое")
          else:
              return ("введите нечетное число больше или равное 5")

In [126]: milrab(7)
Out[126]: ('число ', 7, ', вероятно, простое')

In [127]: milrab(9)
Out[127]: ('число ', 9, 'составное')
```

Figure 5: Реализация теста Миллера-Рабина

Результатом выполнения работы стала реализация алгоритмов проверки чисел на простоту, что отражает проделанную мной работу и полученные новые знания.