РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

Презентауия по

ЛАБОРАТОРНОЙ РАБОТЕ №5

дисциплина: Математические основы защиты информации и информационной безопасности

Студент: Хиссен Али Уэддей
Группа: НПМмд-02-20
Ст. билет № 1032209306

Цель и задание работы

Изучие алгоритм для вычисления Символ Якоби и основные вероятные алгоритмы для проверки чисел на простоту..

1. Алгоритм, реализации тест Ферма

Вход. Нечетное целое цисло n>=5. Выход. “Число n,вероятно,простое” или “Число n составное”.

1. Выбрать случайное целое число a, 2<=a<=2.
2. Вычислить r = a^n-1^(mod n).
3. Если r = 0 результат : “Число n,вероятно,простое”.В противном случае ре- зультат: “Число n составное”.

2.2 Алгоритм, для вычичления Символ Якоби

Вход. Нечетное целое цисло n>=3, целое число a,0 <= a < n. Выход. Символ Якоби. 1. g=1 2. если a =0 результат: 0

3. если a =1 результат: g
4. прадствить a в виде a = 2ka~1~ , где a~1~ нечетное.
5. при четном k положить s=1, при нечетном положить s=1, если n=abs(1(mod8));по- ложить s=-1, если n=abs(3(mod8))
6. при a1 результат: gs
7. если n = 3(mod4) and a1 = 3(mod4) , то s = -s
8. положить a = n mod(a1) n = a1 g = gs и вернуться на шаг 2

2.3 Алгоритм , реализующий тест Соловея - Штрассена

Вход. Нечетное целое цисло n>=5. Выход. “Число n,вероятно,простое” или “Число n составное”.

1. Выбрать случайное целое число a, 2<=a<=2.
2. Вычислить r = a^(n+1)/2^(mod n)
3. Если r не равен 1 и n-1 реузультат: “Число n составное”.
4. Вычислить символ Якоби s = (a/n)
5. Если r = s(mod n) реузультат: “Число n составное”, иначе “Число n,вероят- но,простое”.

2.4 Алгоритм , реализующий тест Миллера - Рабина

Вход. Нечетное целое цисло n>=5. Выход. “Число n,вероятно,простое” или “Число n составное”.

1. представить n-1 в виде n-1 = 2s^r^ , где r нечетное
2. выбрать случайное целое число a, 2<=a<=2
3. вычислить y = a^r^(mod n)
4. при y не равном 1 и n-1 выполнить следующее 4.1. положить j = 1 4.2. если j <= s-1 и y не равен n-1 ,то 4.2.1. положить y = y^2^(mod n) 4.2.2. при y = 1 результат: "Число n составное" 4.2.3. положить j = j+1 4.3. при y не равном n-1 результат: "Число n составное"
5. Результат: “Число n,вероятно,простое”

Програмная чать

1. Алгоритм, реализации теста Ферма

вводим число n для дальнейших проверки на простоту

на данном скриншоте реализован алгоритм теста Ферма как функций в python

:
```
1 import math
2 import random
```

:
```
1 #Entrez deux nombres  n et a; telque a>= 3  et 0<= a < n
2 # definissons g
3 g=1
4 n=int(input("faite entez une valeur:"))
5
```

```
faite entez une valeur:7
```

realisation du premier algorithm

:
```
1  def Algorithm_1(n):
2      a=int(input("faite entrez un reel superieur 2 et inferieur a n "))
3      r =pow(a,n-1)%n
4
5
6      if r==1:
7          print("число ",n,"вероятно, простое")
8      else:
9          print("число",n ,"составное")
10
```

:
```
1 Algorithm_1(n)
```

```
faite entrez un reel superieur 2 et inferieur a n 5
число  7 вероятно, простое
```

2.2 Алгоритм, для вычичления Символ Якоби

на данном скриншоте реализован алгоритм для вычичления Символ Якоби как функций в python

```
1
2  def FonctionSymboleJacobi(n,a):
3
4      g=1
5
6      a_0=a
7
8      if (a_0==0):
9          return 0
10
11     elif(a_0==1):
12         return g
13
14     else:
15         k=0
16         ValeurInitialeK=0
17
18         if (a_0%2!=0):
19
20             a1 = a_0
21             k= ValeurInitialeK
22             #print("a est impair ",a,"la valeur de k est = ",K,"la valeur de a1 =", a1)
23
24             #return a1,k
25
26         else:
27             #print("a est pair =\n ",a,a_0)
28
29             while (a_0%2==0):
30
31                 a_0=a_0/2
32
33                 ValeurInitialeK=ValeurInitialeK + 1
34                 #print(ValeurInitialeK)
35
```

```
            #print(ValeurInitialeK)
        k=ValeurInitialeK
        a1=int(a_0)
       # k=ValeurInitialeK
       # print("a est pair",a,"la valeur de k est = ",k,"la valeur de a1 =", a1)

        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


```

```
FonctionSymboleJacobi(7,2)
```

```
-1
```

2.3 Алгоритм , реализующий тест Соловея - Штрассена

на данном скриншоте реализован алгоритм для вычичления Символ Якоби

|:
```
#a2=int(input("faite entrez un reel superieur a 0"))
#while(a2>=0):
#    a2=int(input("faite entrez un reel superieur a 0"))
```

|:
```
def AgorithmNightingaleStrassen():

    n = int(input('Введите нечетное целое число n>=5: '))
    a = random.randint(2, n -2)

    r1=int(pow(a,(n - 1) / 2))
    r = r1% n


    if r != 1 and r != n - 1:

         print(f'Число {n} - составное')

    else:
        #s = jacobian_symbol(a, n)
        s=FonctionSymboleJacobi(n,a)
        if r % n == s:
            print(f'Число {n} составное')
        else:
            print(f'Число {n} ,вероятно, простое')
```

|:
```
AgorithmNightingaleStrassen()
```

```
Введите нечетное целое число n>=5: 7
Число 7 ,вероятно, простое
```

2.4 Алгоритм , реализующий тест Миллера - Рабина

на данном скриншоте реализован алгоритм для вычичления Миллера - Рабина как функций в python

```
def Algorithm_4(n):

    s=0
    #print("faite entrez un reel a superieur a 5 )

    n_=n-1


    while(n_%2==0):

        n_=n_/2

        s+=1

    r=int(n_)

   # print("s est egal a\n",s)

    #print(" r est egal a\n", r )


    a=int(input("faite entrez un reel a superieur a 2 et inferieur a n-2\n"))

    a1=pow(a,r)

    y=a1%n

    print("a1 est =",a1,"y := ",y)

    if(y!= 1 and y!=n-1):

        j=1

        while( j<=s-1 and y!=n-1 ):
```

```
        j=1

        while( j<=s-1 and y!=n-1 ):

            y=y**2%n

            if y==1:

                print("Число n составное")
            j+=1

        print("y := \n",y)

        if(y!=n-1):

            print("Число n составное ")



     print("Число", n,"вероятно , простое ")

    # print("a est pair",a,"la valeur de k est = ",K ,"la valeur de a1 =", a1)
```

```
56
57
```

]:
```
1 Algorithm_4(n)
2
```

```
faite entrez un reel a superieur a 2 et inferieur a n-2
5
a1 est = 125 y :=  6
Число 7 вероятно , простое
```

ВЫДОД

Цель лабораторной работы была достигнута. Мы изучали алгоритм для вычисления Символов Якоби и основных вероятных алгоритмов для проверки чисел на простоту..