

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

Презентация 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

Изучение алгоритм для вычисления Символ Якоби и основные вероятностные алгоритмы для проверки чисел на простоту..

1. Алгоритм, реализации тест Ферма Вход. Нечетное целое число $n \geq 5$. Выход. "Число n , вероятно, простое" или "Число n составное".

1. Выбрать случайное целое число a , $2 \leq a \leq n-2$.
2. Вычислить $r = a^{n-1} \pmod n$.
3. Если $r = 1$ результат : "Число n , вероятно, простое". В противном случае результат: "Число n составное".

2.2 Алгоритм, для вычисления Символ Якоби Вход. Нечетное целое число $n \geq 3$, 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. представить a в виде $a = 2^k a_1$, где a_1 нечетное.
5. при четном k положить $s=1$, при нечетном положить $s=1$, если $n \pmod 8 = 1$; положить $s=-1$, если $n \pmod 8 = 3$
6. при a_1 результат: gs
7. если $n \pmod 4 = 3$ and $a_1 \pmod 4 = 3$, то $s = -s$
8. положить $a = n \pmod{a_1}$ $n = a_1$ $g = gs$ и вернуться на шаг 2

2.3 Алгоритм , реализующий тест Соловея - Штрассена Вход. Нечетное целое число $n \geq 5$. Выход. "Число n , вероятно, простое" или "Число n составное".

1. Выбрать случайное целое число a , $2 \leq a \leq n-2$.
2. Вычислить $r = a^{(n+1)/2} \pmod n$

3. Если r не равен 1 и $n-1$ результат: "Число n составное".
4. Вычислить символ Якоби $s = (a/n)$
5. Если $r = s \pmod{n}$ результат: "Число n составное", иначе "Число n , вероятно, простое".

2.4 Алгоритм , реализующий тест Миллера - Рабина Вход. Нечетное целое число $n \geq 5$. Выход. "Число n , вероятно, простое" или "Число n составное".

1. представить $n-1$ в виде $n-1 = 2^s r$, где r нечетное
2. выбрать случайное целое число a , $2 \leq a \leq n-2$
3. вычислить $y = a^r \pmod{n}$
4. при y не равном 1 и $n-1$ выполнить следующее
 - 4.1. положить $j = 1$
 - 4.2. если $j \leq s-1$ и y не равен $n-1$, то
 - 4.2.1. положить $y = y^2 \pmod{n}$
 - 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 часть

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 entez 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 entez un reel superieur 2 et inferieur a n 5
число 7 вероятно, простое

2.2 Алгоритм, для вычисления Символ Якоби на данном скриншоте реализован алгоритм для вычисления 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
```

```

34     #print(ValeurInitialeK)|
35     k=ValeurInitialeK
36     a1=int(a_0)
37     # k=ValeurInitialeK
38     # print("a est pair",a,"la valeur de k est = ",k,"la valeur de a1 =", a1)
39
40     if (k% 2) == 0:
41
42         s = 1
43
44     else:
45
46         if abs(n % 8) == 1:
47             s = 1
48         else:
49             s = -1
50     if a1 == 1:
51
52         return g * s
53
54     if (n % 4 == 3 and a1 % 4 == 3):
55
56         s *= -1
57
58     a = n % a1
59
60     n = a1
61
62     g = g * s
63
64

```

1 FonctionSymboleJacobi(7,2)

-1

2.3 Алгоритм , реализующий тест Соловея - Штрассена на данном скриншоте реализован алгоритм для вычисления Символ Якоби

```
|: 1 #a2=int(input("faite entrez un reel superieur a 0"))
2 #while(a2>=0):
3 #    a2=int(input("faite entrez un reel superieur a 0"))
```

```
|: 1 def AgorithmNightingaleStrassen():
2
3     n = int(input('Введите нечетное целое число n>=5: '))
4     a = random.randint(2, n - 2)
5
6     r1=int(pow(a,(n - 1) / 2))
7     r = r1% n
8
9
10    if r != 1 and r != n - 1:
11
12        print(f'Число {n} - составное')
13
14    else:
15        #s = jacobian_symbol(a, n)
16        s=FonctionSymboleJacobi(n,a)
17        if r % n == s:
18            print(f'Число {n} составное')
19        else:
20            print(f'Число {n} ,вероятно, простое')
```

```
|: 1 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 вычисления Миллера - Рабина как функций в python

```

: 1  def Algorithm_4(n):
2
3      s=0
4      #print("faite entrez un reel a superieur a 5 ")
5
6      n_=n-1
7
8
9      while(n_%2==0):
10
11          n_=n_/2
12
13          s+=1
14
15      r=int(n_)
16
17      # print("s est egal a\n",s)
18
19      #print(" r est egal a\n", r )
20
21
22      a=int(input("faite entrez un reel a superieur a 2 et inferieur a n-2\n"))
23
24      a1=pow(a,r)
25
26      y=a1%n
27
28      print("a1 est =",a1,"y := ",y)
29
30      if(y!= 1 and y!=n-1):
31
32          j=1
33
34          while( j<=s-1 and y!=n-1 ):
35
36              j=1
37
38              while( j<=s-1 and y!=n-1 ):
39
40                  y=y**2%n
41
42                  if y==1:
43                      print("Число n составное")
44                      j+=1
45
46                  print("y := \n",y)
47
48                  if(y!=n-1):
49                      print("Число n составное ")
50
51
52          print("Число", n,"вероятно , простое ")
53
54      # print("a est pair",a,"la valeur de k est = ",k ,"la valeur de a1 =", a1)
55

```

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

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