МИНОБРНАУКИ РОССИИ
ФГБОУ ВО «СГУ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

АЛГОРИТМЫ АЛГЕБРЫ И ТЕОРИИ ЧИСЕЛ

ЛАБОРАТОРНАЯ РАБОТА №9

студента 4 курса 431 группы
направления 10.05.01 — Компьютерная безопасность
факультета КНиИТ
Никитина Арсения Владимировича

Проверил
доцент _____ А. С. Гераськин

Саратов 2022

СОДЕРЖАНИЕ

1 Задание лабораторной работы

Осуществить построение большого простого числа с использованием критерия Люка.

2 Теоретическая часть

Тест основывается на следующем критерии простоты чисел Мерсенна:

Пусть p — простое нечётное. Число Мерсенна $M_p = 2^p - 1$ простое тогда и только тогда, когда оно делит нацело $(p-1)$-й член последовательности: $4,\ 14,\ 194,\ 37634,\ \ldots$, которая задается рекуррентно:

$$S_k = \begin{cases} 4 & k = 1, \\ S_{k-1}^2 - 2 & k > 1. \end{cases}$$

Доказательство

Один из подходов к доказательству основан на использовании функций Люка:

$$V_n(P, Q) = \alpha^n + \beta^n,$$

$$U_n(P, Q) = \frac{\alpha^n - \beta^n}{\alpha - \beta},$$

где $\alpha,\ \beta$ — корни квадратного уравнения:

$$x^2 - Px + Q = 0$$

с дискриминантом $D = P^2 - 4Q$, причём P и Q взаимно просты.

В частности, при доказательстве используются некоторые свойства этих функций, а именно:

1. $V_n^2 - DU_n^2 = 4Q^n$
2. $V_{2n} = V_n^2 - 2Q^n, \quad U_{2n} = U_n V_n$
3. $\frac{V_n + \sqrt{D}U_n}{2} = \left(\frac{P + \sqrt{D}}{2}\right)^n$
4. Если $P' \equiv P \pmod N, Q' \equiv Q \pmod N, (Q, N) = 1$ и $QP' = P^2 - 2Q \pmod N$, то:

$$\begin{cases} Q^n V_n(P', 1) \equiv V_{2n}(P, Q) \pmod N \\ PQ^{n-1} U_n(P', 1) \equiv U_{2n}(P, Q) \pmod N \end{cases}$$

5. Если p — простое, такое, что $2DQ$ взаимно просто с p, то p делит нацело $U_{\Phi(p)}(P, Q)$, где $\Phi(p) = p - \left(\frac{D}{p}\right)$, а $\left(\frac{D}{p}\right)$ — символ Лежандра.

Необходимость

Из свойства 4. по модулю $N = M_p$ при $P = 2$, $Q = -2$, следует:

$2^n V_n(-4, 1) \equiv V_{2n}(2, -2) \pmod N$, а по свойству 2.

$V_{2n}(-4, 1) = V_n^2(-4, 1) - 2$, поэтому

$S_{p-1} \equiv V_{\frac{N+1}{4}}(-4, 1) \pmod N$ и

$V_{\frac{N+1}{2}}(2, -2) \equiv 2^{\frac{N+1}{4}} S_{p-1} \pmod N$

$D = 2^2 - 4 \cdot (-2) = 12$, поэтому если N — простое, то $\left(\frac{D}{N}\right) = -1$ и из последних двух свойств N делит $U_{N+1}(2, -2) = V_{\frac{N+1}{2}}(2, -2) U_{\frac{N+1}{2}}(2, -2)$

Далее, из свойств 1. и 2.

$V_{N+1} = V_{\frac{N+1}{2}}^2 - 2 \cdot (-2)^{\frac{N+1}{2}} \equiv 8 + 4 = 12 \pmod N$, но по свойству 3 имеем:

$V_{N+1} \equiv 2(1+\sqrt{3})^{N+1} = 2(1+\sqrt{3})(1+3^{\frac{N-1}{2}}\sqrt{3}) \equiv 2(1-3) = -4 \pmod N$, то есть N делит $V_{\frac{N+1}{2}}(2, -2)$, а значит и S_{p-1}.

Достаточность

Если N делит S_{p-1}, то из доказательства необходимости следует, что оно делит и $V_{\frac{N+1}{2}}$. N взаимно просто с $U_{\frac{N+1}{2}}$ по свойству 1., а по свойству 2. — делит U_{N+1}. Но тогда каждый простой делитель числа N представим в виде $\pm 1 + k2^p > \sqrt{N}$, то есть $N = M_p$ — простое.

3 Практическая часть

3.1 Пример работы алгоритма

Рисунок 1

3.2 Код программы, реализующей рассмотренный алгоритм

```
import random

def sieve_of_eratosthenes(n):
    sieve = list(range(2,n))
    i = 0
    while True:
        cur_elem = sieve[i]
        cur_elem_in_power = cur_elem ** 2
        if cur_elem_in_power <= n:
            start_position = min(filter(lambda x : x >= cur_elem_in_power,
            ↪  sieve))
            for elem in sieve[sieve.index(start_position):]:
                if not elem % cur_elem:
                    sieve.remove(elem)
            i += 1

        else:
            return sieve

def power(a, n):
    return (1 if n == 0
            else power(a * a, n // 2) if n % 2 == 0
            else a * power(a, n - 1))

```

```
24
25 def lucas_lehmer_test(p):
26     s = 4
27     k = 1
28     m = power(2, p) - 1
29     while k != p - 1:
30         s = (s * s - 2) % m
31         k += 1
32     return m if not s else False
33
34
35 def main():
36     exists = []
37     while True:
38
39         print('\nПостроить большое простое число с помощью Критерия Люка -
           ↪ \enter')
40         print('Выход из программы - 2')
41
42         try:
43             value = int(input('Введите значение: '))
44
45         except ValueError:
46             value = 1
47
48         if value == 1:
49             while True:
50                 p = sieve_of_eratosthenes(2000)
51                 a = p[random.randrange(0, len(p))]
52                 if a not in exists:
53                     exists.append(a)
54
55                     m = lucas_lehmer_test(a)
56                     if m:
57                         print(f'Случайно было выбрано простое число p =
                           ↪ {a}')
58                         print(f'Построено простое число: {m}')
59                         break
60
61
62         elif value == 2:
```

```
63              print('Работа программы завершена')
64              return
65
66
67  if __name__ == "__main__":
68      main()
```