

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

СОДЕРЖАНИЕ

1	Задание лабораторной работы	3
2	Теоретическая часть	4
3	Практическая часть.....	6
3.1	Пример работы алгоритма.....	6
3.2	Код программы, реализующей рассмотренный алгоритм	6

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, ..., 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 α, β — 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:

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

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

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

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

$D = 2^2 - 4 \cdot (-2) = 12$, 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.

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

$$V_{N+1} \equiv 2(1 + \sqrt{3})^{N+1} = 2(1 + \sqrt{3})(1 + 3^{\frac{N-1}{2}} \sqrt{3}) \equiv 2(1 - 3) = -4 \pmod{N},$$

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

```
Построить большое простое число с помощью Критерия Лока - \enter
Выход из программы - 2
Введите значение:
Случайно было выбрано простое число p = 31
Построено простое число: 2147483647

Построить большое простое число с помощью Критерия Лока - \enter
Выход из программы - 2
Введите значение:
Случайно было выбрано простое число p = 19
Построено простое число: 524287

Построить большое простое число с помощью Критерия Лока - \enter
Выход из программы - 2
Введите значение:
Случайно было выбрано простое число p = 1279
Построено простое число: 10407932194668399081925240327364085538615262247266704805319112350403608059673360298012239441732324184842421613954281007791383566248323464908139906605677320762924129509389220345773183349661583550472
959420547689811211693677147548478866962501384438260291732348885311160828538416585028255604666224831890918801847068222203140521026698435488732958028878050809736186900714720710555703168729087

Построить большое простое число с помощью Критерия Лока - \enter
Выход из программы - 2
Введите значение:
Случайно было выбрано простое число p = 89
Построено простое число: 618970019642690137449562111

Построить большое простое число с помощью Критерия Лока - \enter
Выход из программы - 2
Введите значение:
Случайно было выбрано простое число p = 3
Построено простое число: 7

Построить большое простое число с помощью Критерия Лока - \enter
Выход из программы - 2
Введите значение:
Случайно было выбрано простое число p = 107
Построено простое число: 162259276829213363391578010288127
```

Рисунок 1

3.2 Код программы, реализующей рассмотренный алгоритм

```
1 import random
2
3 def sieve_of_eratosthenes(n):
4     sieve = list(range(2,n))
5     i = 0
6     while True:
7         cur_elem = sieve[i]
8         cur_elem_in_power = cur_elem ** 2
9         if cur_elem_in_power <= n:
10             start_position = min(filter(lambda x : x >= cur_elem_in_power,
11                                         ↪ sieve))
12             for elem in sieve[sieve.index(start_position):]:
13                 if not elem % cur_elem:
14                     sieve.remove(elem)
15             i += 1
16         else:
17             return sieve
18
19 def power(a, n):
20     return (1 if n == 0
21             else power(a * a, n // 2) if n % 2 == 0
22             else a * power(a, n - 1))
23
```

```

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 -
40             ↪ \enter')
41         print('Выход из программы - 2')
42
43         try:
44             value = int(input('Введите значение: '))
45
46         except ValueError:
47             value = 1
48
49         if value == 1:
50             while True:
51                 p = sieve_of_eratosthenes(2000)
52                 a = p[random.randrange(0, len(p))]
53                 if a not in exists:
54                     exists.append(a)
55
56                 m = lucas_lehmer_test(a)
57                 if m:
58                     print(f'Случайно было выбрано простое число p =
59                         ↪ {a} ')
60                     print(f'Построено простое число: {m} ')
61                     break
62
63         elif value == 2:

```

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