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

АЛГОРИТМЫ АЛГЕБРЫ И ТЕОРИИ ЧИСЕЛ

ЛАБОРАТОРНАЯ РАБОТА №6

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 проверку чисел на простоту с помощью теста Соловея-Штрассена.

2 Теоретическая часть

2.1 Символ Якоби и его свойства

Пусть P — нечётное, большее единицы число и $P = p_1 p_2 \dots p_n$ — его разложение на простые множители (среди $p_1, \ \dots, \ p_n$ могут быть равные). Тогда для произвольного целого числа a символ Якоби определяется равенством:

$\left(\frac{a}{P}\right) = \left(\frac{a}{p_1}\right)\left(\frac{a}{p_2}\right)\cdots\left(\frac{a}{p_n}\right)$, где $\left(\frac{a}{p_i}\right)$ — символы Лежандра.

Свойства символа Якоби

1. Мультипликативность: $\left(\frac{ab}{P}\right) = \left(\frac{a}{P}\right)\left(\frac{b}{P}\right)$. В частности, $\left(\frac{a^2 b}{P}\right) = \left(\frac{b}{P}\right)$.
2. Периодичность: если $a \equiv b \pmod P$, то $\left(\frac{a}{P}\right) = \left(\frac{b}{P}\right)$.
3. $\left(\frac{1}{P}\right) = 1$.
4. $\left(\frac{-1}{P}\right) = (-1)^{\frac{P-1}{2}}$.
5. $\left(\frac{2}{P}\right) = (-1)^{\frac{P^2-1}{8}}$.
6. Если Q — нечётное натуральное число, взаимно простое с P, то $\left(\frac{Q}{P}\right)\left(\frac{P}{Q}\right) = (-1)^{\frac{P-1}{2}\cdot\frac{Q-1}{2}}$
7. Если P и Q взаимно простые и нечётные, то $\left(\frac{Q}{P}\right) = (-1)^{\frac{P-1}{2}\cdot\frac{Q-1}{2}}\left(\frac{P}{Q}\right)$.

2.2 Тест Соловея-Штрассена

Теорема Соловея-Штрассена

Пусть n нечетно, тогда для того чтобы n было простым необходимо и достаточно, чтобы для каждого $a \in Z_n^*$ было выполнено $a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right) \pmod n$.

Доказательство

Необходимость следует из критерия Эйлера для символа Лежандра. Докажем достаточность методом от противного.

Пусть $\forall a \in \mathbb{Z}_n^* : a^{\frac{n-1}{2}} \equiv \frac{a}{n} \pmod n$, но n — составное.

$a^{n-1} = (a^{\frac{n-1}{2}})^2 \equiv \left(\frac{a}{n}\right)^2 \pmod n$

$\left(\frac{a}{n}\right)^2 = 1 \Rightarrow a^{n-1} \equiv 1 \pmod n$

Таким образом, n — число Кармайкла. $\Rightarrow n = p_1 \times p_2 \times \ldots \times p_s, \ \varphi(p_i) = p - 1, \ i = \overline{1, s}$.

Рассмотрим такое b, что $\left(\frac{b}{p_1}\right) \equiv 1 \pmod n$

Найдем такое a, что:

$$\begin{cases} a \equiv b \pmod{p_1} \\ a \equiv 1 \pmod{p_i},\ i = \overline{2, s} \end{cases} \tag{1}$$

Такое a существует по Китайской теореме об остатках и принадлежит $\mathbb{Z}_n^*$ (так как взаимнопросто с n).

$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right) \times \left(\frac{a}{p_2}\right) \times \ldots \times \left(\frac{a}{p_s}\right) = \left(\frac{a}{p_1}\right) = \left(\frac{b}{p_1}\right) = -1;$

$a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right) \pmod{n};$

$\left(\frac{a}{n}\right) = -1 \Rightarrow a^{n-1} \equiv -1 \pmod{n};$

$a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right) = -1 \pmod{p_1};$

$a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right) = -1 \pmod{p_2} \Rightarrow$ получили противоречие с тем, что $a \equiv 1 \pmod{p_i},\ i = \overline{2, s}$. А, значит, неверно и предположение о том, что n — составное.

3 Практическая часть

3.1 Пример работы алгоритма

```
Проверить число на простоту тестом Соловея-Штрассена - \enter
Выход из программы - 2
Введите значение:

Введите число, которое требуется проверить на простоту тестом Соловея-Штрассена: 3011
Введите количество тестов, которое требуется провести: 10
Число 3011 является простым с вероятностью 0.9999990463256836
Свидетелями его простоты являются числа [1377, 1996, 2695, 633, 1309, 632, 1985, 1031, 2523, 200]

Проверить число на простоту тестом Соловея-Штрассена - \enter
Выход из программы - 2
Введите значение:

Введите число, которое требуется проверить на простоту тестом Соловея-Штрассена: 3012341
Введите количество тестов, которое требуется провести: 8
Число 3012341 не является простым
Свидетелями его непростоты являются числа [1589836, 1430564, 3003989, 818175, 1275107, 1162803, 845195, 254931
9]

Проверить число на простоту тестом Соловея-Штрассена - \enter
Выход из программы - 2
Введите значение:

Введите число, которое требуется проверить на простоту тестом Соловея-Штрассена: 561
Введите количество тестов, которое требуется провести: 10
Число 561 не является простым
Свидетелями его непростоты являются числа [141, 279, 66, 144, 530, 312, 23, 435, 139, 434]

Проверить число на простоту тестом Соловея-Штрассена - \enter
Выход из программы - 2
Введите значение: 2
```

Рисунок 1

3.2 Код программы, реализующей рассмотренный алгоритм

```
import random

def gcd(a, n):

    if a == 0:
        return n
    else:
         return gcd(n % a, a)


def modula_power(a, power, modula):

    b = 1
    while power:
        if not power % 2:
            power //= 2
            a = (a * a) % modula
        else:
            power -= 1
            b = (b * a) % modula
```

```
    return b


def get_jacobi(a, n, res):

    if a == 0:
        return res

    twos = 0
    while not a % 2:
        a //= 2
        twos += 1

    if twos % 2:
        res *= -1 if ((n * n - 1) // 8) % 2 else 1


    res *=  -1 if ((a - 1) * (n - 1) // 4) % 2 else 1

    return get_jacobi(n % a, a, res)


def solovay_shtrassen_test(n, k):

    if not n % 2:
        return [], [2]

    power = (n - 1) // 2
    witnesses_of_simplicity = []
    witnesses_of_not_simplicity = []

    for _ in range(k):

        a = random.randrange(2, n - 1)
        if gcd(a, n) != 1:
            witnesses_of_not_simplicity.append(a)
        else:
            if modula_power(a, power, n) == get_jacobi(a, n, 1) % n:
                witnesses_of_simplicity.append(a)
            else:
                witnesses_of_not_simplicity.append(a)
```

```
62
63      return witnesses_of_simplicity, witnesses_of_not_simplicity
64
65
66  def main():
67
68      while True:
69
70          print('\nПроверить число на простоту тестом Соловея-Штрассена -
            ↪ \enter')
71          print('Выход из программы - 2')
72
73          try:
74              value = int(input('Введите значение: '))
75          except ValueError:
76              value = 1
77
78          if value == 1:
79
80              n = int(input('\nВведите число, которое требуется проверить на '\
81                  'простоту тестом Соловея-Штрассена: '))
82
83              number_of_tests = int(input('Введите количество тестов, которое '\
84                  'требуется провести: '))
85
86              witnesses_of_simplicity, witnesses_of_not_simplicity =
                ↪ solovay_shtrassen_test(n, number_of_tests)
87
88              if not witnesses_of_not_simplicity:
89
90                  print(f'Число {n} является простым с вероятностью {1 -
                    ↪ (1/4)**number_of_tests}')
91                  print(f'Свидетелями его простоты являются числа
                    ↪ {witnesses_of_simplicity}')
92
93              else:
94
95                  print(f'Число {n} не является простым')
96                  print(f'Свидетелями его непростоты являются числа
                    ↪ {witnesses_of_not_simplicity}')
97
```

```
98                  if witnesses_of_simplicity:
99                      print(f'Лжесвидетелями его простоты являются числа
                        ↪  {witnesses_of_simplicity}')
100
101         if value == 2:
102             break
103
104
105 if __name__ == '__main__':
106     main()
```