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

АЛГОРИТМЫ АЛГЕБРЫ И ТЕОРИИ ЧИСЕЛ

ЛАБОРАТОРНАЯ РАБОТА №7

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 Рабина-Миллера

2 Теоретическая часть

Формулировка теста

Пусть n — простое число и $n-1=2^s d$, где d — нечётно. Тогда $\forall a \in \mathbf{Z}_n$ ыполняется хотя бы одно из условий:

$$1.\ a^d \equiv 1 \pmod{n}$$

2. Существует целое число $r < s$ такое что $a^{2^r d} \equiv -1 \pmod{n}$

Доказательство

Если это утверждение (условие 1 или 2) выполняется для некоторых чисел a и n (не обязательно простого), то число a называют *свидетелем простоты* числа n по Миллеру, а само число n — вероятно простым.

(При случайно выбранном a вероятность ошибочно принять составное число за простое составляет 25%, но её можно уменьшить, выполнив проверки для других a.

В случае когда выполняется контрапозиция доказанного утверждения, то есть если найдётся число a такое, что:

$$a^d \not\equiv 1 \pmod{n} \text{ и } \forall r:\ 0 \le r \le s-1:\ a^{2^r d} \not\equiv -1 \pmod{n},$$

то число n не является простым. В этом случае число a называют свидетелем того, что число n составное.

Идея теста заключается в том, чтобы проверять для случайно выбранных чисел $a < n$, являются ли они свидетелями простоты числа n. Если найдётся свидетель того, что число составное, то число действительно является составным. Если было проверено k чисел, и все они оказались свидетелями простоты, то число считается простым. Для такого алгоритма вероятность принять составное число за простое будет меньше $(1/4)^k$.

3 Практическая часть

3.1 Пример работы алгоритма

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

Введите число n: 300973
Введите количество раундов теста: 5
Число 300973 является простым с вероятностью 0.9990234375.

Проверить число на простоту тестом Миллера-Рабина - \enter
Выход из программы - 2
Введите значение:

Введите число n: 3003
Введите количество раундов теста: 4
Число 3003 не является простым.

Проверить число на простоту тестом Миллера-Рабина - \enter
Выход из программы - 2
Введите значение:

Введите число n: 78997221
Введите количество раундов теста: 5
Число 78997221 не является простым.

Проверить число на простоту тестом Миллера-Рабина - \enter
Выход из программы - 2
Введите значение: 2
Работа программы завершена
```

Рисунок 1

3.2 Код программы, реализующей рассмотренный алгоритм

```
1  import random
2
3  def representation(p):
4
5      twos = 0
6
7      while not p % 2:
8          twos += 1
9          p //= 2
10
11     return twos, p
12
13
14 def modula_pow(number, power, modula):
15
16     number_save = number
17
```

```
18      for _ in range(power - 1):
19          number = number * number_save % modula
20      return number
21
22
23  def miller_rabin_test(n, k):
24
25      s, t = representation(n - 1)
26
27      for _ in range(k):
28
29          a = random.randint(2, n - 2)
30
31          x = modula_pow(a, t, n)
32
33          if x == 1 or x == n - 1:
34              continue
35
36          flag = False
37          for _ in range(s - 1):
38
39              x = x * x % n
40
41              if x == 1:
42                  return False
43
44              if x == n - 1:
45                  flag = True
46                  break
47
48          if flag:
49              continue
50
51          return False
52
53      return True
54
55
56  def main():
57
58      while True:
```

```
59
60          print('\nПроверить число на простоту тестом Миллера-Рабина - \enter')
61          print('Выход из программы - 2')
62
63          try:
64              value = int(input('Введите значение: '))
65
66          except ValueError:
67              value = 1
68
69          if value == 1:
70
71              n = int(input("\nВведите число n: "))
72              k = int(input("Введите количество раундов теста: "))
73
74              if (miller_rabin_test(n, k)):
75                  print(f'Число {n} является простым с вероятностью {1 - (1 /
                        ↪  4) ** k}.')
76              else:
77                  print(f'Число {n} не является простым.')
78
79          elif value == 2:
80              print('Работа программы завершена')
81              return
82
83
84  if __name__ == "__main__":
85      main()
86
87      # numbers = [i for i in range(17, int(1e4)) if miller_rabin_test(i, 3)]
88      # print(numbers)
89
90
91
```