Отчет по лабораторной работе №5

Дисциплина: Математические основы защиты информации и информационной безопасности

Выполнила Дяченко Злата Константиновна, НПМмд-02-22

Содержание

# 1 Цель работы

Ознакомится и реализовать вероятностные алгоритмы проверки чисел на простоту.

# 2 Задание

Реализовать алгоритм, реализующий тест Ферма; алгоритм вычисления символа Якоби; алгоритм, реализующий тест Соловэя-Штрассена; алгоритм, реализующий тест Миллера-Рабина.

# 3 Теоретическое введение

Пусть – целое число. Числа ±1, ± называются тривиальными делителями числа . Целое число называется простым, если оно не является делителем единицы и не имеет других делителей, кроме тривиальных. В противном случае число называется составным. Проверка чисел на простоту является составной частью алгоритмов генерации простых чисел, применяемых в криптографии с открытым ключом. Алгоритмы проверки на простоту можно разделить на вероятностные и детерминированные. Детерминированный алгоритм всегда действует по одной и той же схеме и гарантированно решает поставленную задачу (или не дает никакого ответа). Вероятностный алгоритм использует генератор случайных чисел и дает не гарантированно точный ответ. Вероятностные алгоритмы в общем случае не менее эффективны, чем детерминированные (если используемый генератор случайных чисел всегда дает набор одних и тех же чисел, зависящих от входных данных, то вероятностный алгоритм становится детерминированным).

Тест Ферма основан на малой теореме Ферма: для простого числа и произвольного числа , 1 ≤ ≤ p − 1, выполняется сравнение

Следовательно, если для нечетного существует такое целое , что 1 ≤ < n, НОД() = 1 и , то число составное. Отсюда получаем следующий вероятностный алгоритм проверки числа на простоту.

1. Алгоритм, реализующий тест Ферма.  
   *Вход*. Нечетное целое число .  
   *Выход*. «Число , вероятно, простое» или «Число составное».
   1. Выбрать случайное целое число , 2 ≤ ≤ n − 2.
   2. Вычислить .
   3. При результат: «Число , вероятно, простое». В противном случае результат: «Число составное».

Тест Соловэя-Штрассена. Основан на критерии Эйлера: нечетное число ݊является простым тогда и только тогда, когда для любого целого числа , 1 ≤ ≤ n − 1, взаимно простого с , выполняется сравнение:

где – символ Якоби. Пусть , где и числа простые (не обязательно различные). Символ Якоби определяется равенством

1. Алгоритм вычисления символа Якоби.  
   *Вход*. Нечетное целое число , целое число , 0 ≤ < n.  
   *Выход*. Символ Якоби .
   1. Положить .
   2. При результат: 0.
   3. При результат: g.
   4. Представить в виде , где число нечетное.
   5. При четном положить , при нечетном положить , если ; положить , если .
   6. При результат: g\*s.
   7. Если и , то .
   8. Положить , , и вернуться на шаг 2.
2. Алгоритм, реализующий тест Соловэя-Штрассена.  
   *Вход*. Нечетное целое число .  
   *Выход*. «Число , вероятно, простое» или «Число составное».
   1. Выбрать случайное целое число , 2 ≤ < n − 2.
   2. Вычислить .
   3. При и результат: «Число составное».
   4. Вычислить символ Якоби
   5. При результат: «Число составное». В противном случае результат: «Число , вероятно, простое».

На сегодняшний день для проверки чисел на простоту чаще всего используется тест Миллера-Рабина, основанный на следующем наблюдении. Пусть число нечетное и ,где – нечетное. Если простое, то для любого ≥ 2, взаимно простого с , выполняется условие .

1. Алгоритм, реализующий тест Миллера-Рабина.  
   *Вход*. Нечетное целое число .  
   *Выход*. «Число , вероятно, простое» или «Число составное».
   1. Представить в виде , где число нечетное.
   2. Выбрать случайное целое число , 2 ≤ < n − 2.
   3. Вычислить .
   4. При и выполнять следующие действия:
      1. Положить .
      2. Если и :
         1. Положить .
         2. При результат: «Число составное».
         3. Положить
      3. При результат: «Число составное».
   5. Результат: «Число , вероятно, простое».

# 4 Выполнение лабораторной работы

## 4.1 Шаг 1

Ознакомилась с предоставленными теоретическими данными. Для выполнения задания решила использовать язык Python. Написала функцию, выполняющую проверку числа на простоту с использованием теста Ферма. Код функции и результат ее использования представлен на Рисунке 1 (рис. - fig. 1). Функция принимает на вход число . При условии, что реализуется алгоритм, представленный в теоретическом введении и функция возвращает отчет. Если условие не выполняется, будет выведено соответствующее сообщение.



Figure 1: Реализация алгоритма, реализующего тест Фирма

## 4.2 Шаг 2

На Рисунке 2 (рис. - fig. 2) представлен код функции, реализующий алгоритм вычисления символа Якоби. Пример выполнения показан на Рисунке 3 (рис. - fig. 3).



Figure 2: Реализация алгоритма вычисления символа Якоби



Figure 3: Пример работы алгоритма вычисления символа Якоби

## 4.3 Шаг 3

На Рисунке 3 (рис. - fig. 3) представлен код функции, реализующий тест Соловэя-Штрассена.

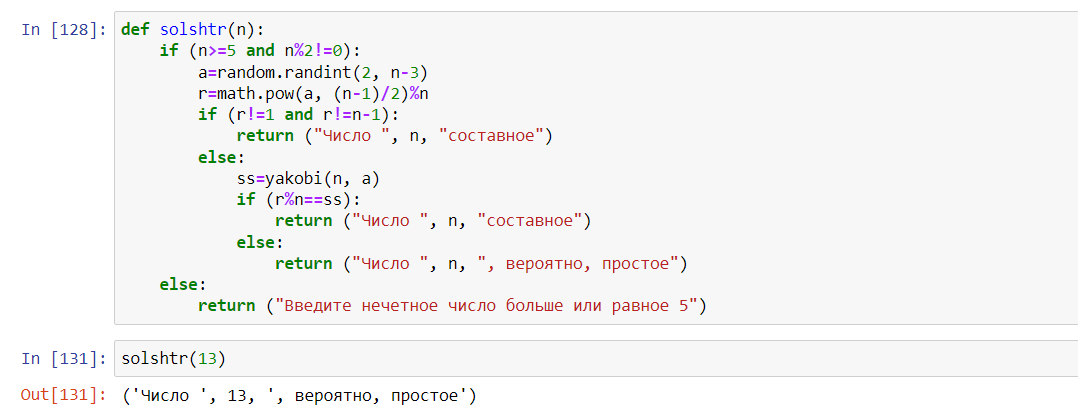


Figure 4: Реализация теста Соловэя-Штрассена

## 4.4 Шаг 4

На Рисунке 4 (рис. - fig. 4) представлен код функции, реализующий тест Миллера-Рабина, и пример выполнения.



Figure 5: Реализация теста Миллера-Рабина

# 5 Выводы

Я ознакомилась с алгоритмами проверки чисел на простоту и реализовала их. Результаты работы находятся в [репозитории на GitHub](https://github.com/ZlataDyachenko/workD), а также есть [скринкаст выполнения лабораторной работы](https://www.youtube.com/watch?v=Sg-IUSGijSI).