Математические основы защиты информации и информационной безопасности. Отчет по лабораторной работе №5

Вероятностные алгоритмы проверки чисел на простоту

Терентьев Егор Дмитриевич 1132236902

Содержание

# 1 Цель работы

Освоить на практике алгоритмы проверки чисел на простоту.

# 2 Выполнение лабораторной работы

Требуется реализовать:

1. Алгоритм, реализующий тест Ферма
2. Алгоритм вычисления символа Якоби
3. Алгоритм, реализующий тест Соловэя-Штрассена
4. Алгоритм, реализующий тест Миллера-Рабина.

## 2.1 Алгоритм, реализующий тест Ферма

Алгоритм основан на малой теореме Ферма, которая утверждает, что если n - простое число, то для любого целого числа a, не являющегося кратным n, выполняется a^(n-1) ≡ 1 (mod n). Алгоритм выбирает случайные значения a и проверяет условие. Если оно не выполняется для какого-либо a, то n считается составным. Если оно выполняется для всех выбранных a, то n вероятно является простым.

Реализация на Python предствлена на рисунке 1 fig. 1.

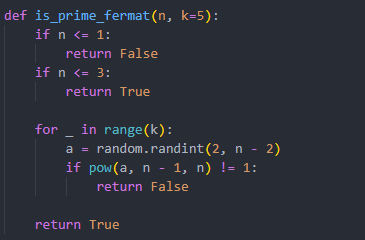


Figure 1: fermat

## 2.2 Символ Якоби

Символ Якоби обобщает символ Лежандра и используется для определения вычетов в кольце вычетов по модулю n. Для нечетного простого числа n и целого числа a, символ Якоби Jacobi(a, n) равен 1, если a является квадратичным вычетом по модулю n, -1, если a является квадратичным невычетом, и 0, если a кратно n. Символ Якоби используется в различных алгоритмах для проверки простоты и для решения квадратичных уравнений по модулю.

Реализация на Python предствлена на рисунке 2 fig. 2.

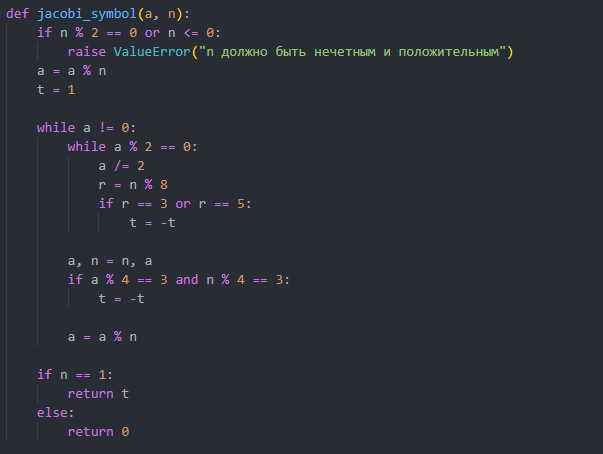


Figure 2: jacobi

## 2.3 Тест Соловея-Штрассена

Этот алгоритм использует символ Якоби и проверяет, является ли число простым. Алгоритм выбирает случайное целое число a и проверяет два условия: 1) a не делится на n, и 2) символ Якоби Jacobi(a, n) равен результату вычисления с использованием символа Лежандра. Если оба условия выполняются для всех выбранных a, то n вероятно является простым числом.

Реализация на Python предствлена на рисунке 3 fig. 3.

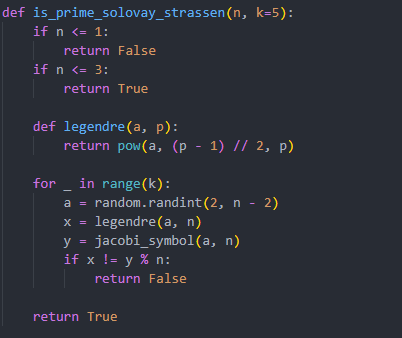


Figure 3: solovay\_strassen

## 2.4 Тест Миллера-Рабина

Этот алгоритм также использует вероятностный метод для проверки простоты числа. Алгоритм выбирает случайное целое число a и разлагает n - 1 на 2^s \* d, где s - четное, и d нечетное. Затем алгоритм проверяет условия Миллера-Рабина: 1) a^d ≡ 1 (mod n), и 2) для всех i от 0 до s-1, a(2i \* d) ≡ -1 (mod n) или a(2i \* d) ≡ 1 (mod n). Если оба условия выполняются для всех выбранных a, то n вероятно является простым числом.

Реализация на Python предствлена на рисунке 4 fig. 4.

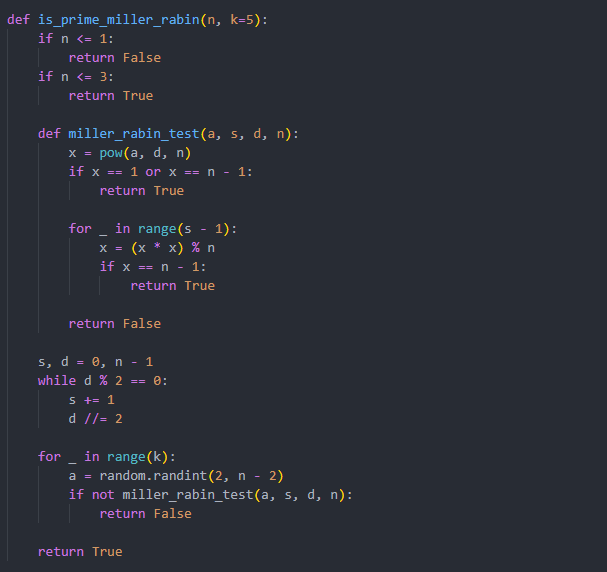


Figure 4: miller\_rabin

## 2.5 Результат работы программы

функция запуска fig. 5.

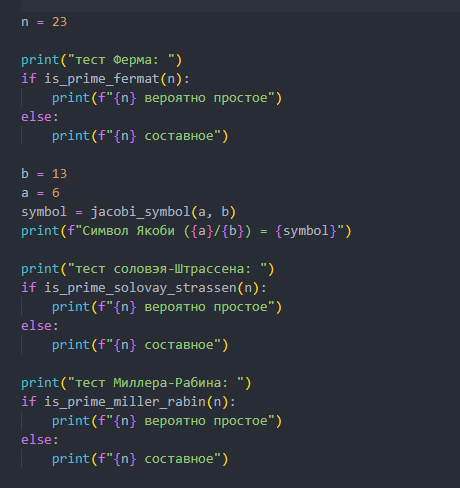


Figure 5: main

Выходные значения программы fig. 6.

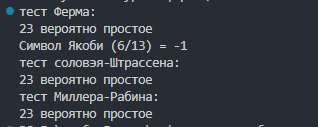


Figure 6: output

# 3 Выводы

В результате выполнения работы я освоил на практике применение алгоритмов проверки чисел на простоту.

# 4 Список литературы

1. Методические материалы курса