Лабораторна работа № 6

Вероятностные проверки чисел на простоту

Покрас Илья Михайлович

Содержание

# 1 Цель работы

Реализовать алгоритмы вероятностной проверки чисел на простоту.

# 2 Задание

* Реализовать алгоритм теста Ферма;
* Реализовать алгоритм вычисления символа Якоби;
* Реализовать алгоритм теста Соловэя-Штрассена;
* Реализовать алгоритм теста Миллера-Рабина.

# 3 Теоретическое введение

Тест простоты Ферма в теории чисел — это тест простоты натурального числа n, основанный на малой теореме Ферма. Малая теорема Ферма - теорема теории чисел, которая утверждает, что: если p - простое число и а - целое число, не делящееся на p, то a^(p-1) - 1 делится на p

Символ Якоби — теоретико-числовая функция двух аргументов, введённая К. Якоби в 1837 году. Символ Якоби обобщает символ Лежандра на все нечётные числа, большие единицы. Символ Кронекера — Якоби, в свою очередь, обобщает символ Якоби на все целые числа, но в практических задачах символ Якоби играет гораздо более важную роль, чем символ Кронекера — Якоби.

Тест Соловея — Штрассена — вероятностный тест простоты, открытый в 1970-х годах Робертом Мартином Соловеем совместно с Фолькером Штрассеном. Тест всегда корректно определяет, что простое число является простым, но для составных чисел с некоторой вероятностью он может дать неверный ответ. Основное преимущество теста заключается в том, что он, в отличие от теста Ферма, распознает числа Кармайкла как составные.

Тест Миллера — Рабина — вероятностный полиномиальный тест простоты. Тест Миллера — Рабина, наряду с тестом Ферма и тестом Соловея — Штрассена, позволяет эффективно определить, является ли данное число составным. Однако, с его помощью нельзя строго доказать простоту числа. Тем не менее тест Миллера — Рабина часто используется в криптографии для получения больших случайных простых чисел.

# 4 Выполнение лабораторной работы

## 4.1 Алгоритм теста Ферма

Я создал функцию алгоритма теста Ферма для проверки числа на простоту с входными данными в виде числа и с выводом сообщение о том, каким число является, в зависимости от того, выполненяется ли условие. Далее вызвал функцию, входным аргументом котороой будет являться переменная, которую я задал до этого (рис. 1).

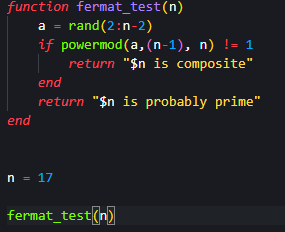


Рис. 1: Функция алгоритма теста Ферма

И получил следующее сообщение (рис. 2):

Результат выполненного кода (1)

Рис. 2: Результат выполненного кода (1)

## 4.2 Алгоритм вычисления символа Якоби

Далее я создал функцию алгоритма вычисления символа Якоби для работы которого требуются входные параметры а и n, такие, что n >= 3, 0 <= а < n. Далее я задал переменные, которые будут являться этими входными данными и используя их вызвал функцию (рис. 3).

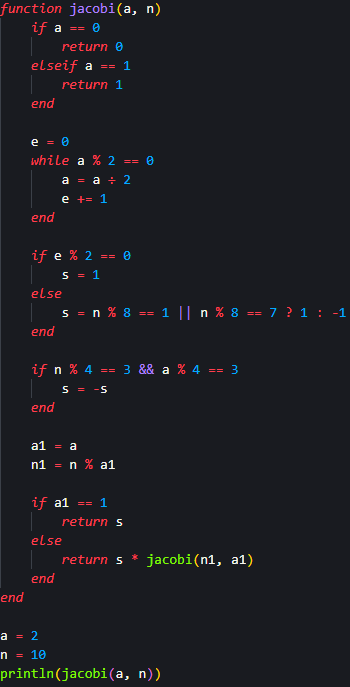


Рис. 3: Функция алгоритма вычисления символа Якоби

И получил следующий результат (рис. 4):

Результат выполненного кода (2)

Рис. 4: Результат выполненного кода (2)

## 4.3 Алгоритм теста Соловэя-Штрассена

После я создал функцию алгоритма теста Соловэя-Штрассена, который так же вызывает функцию вычисления символа Якоби, который был реализован ранее. Будет выводиться сообщение о том, каким число является, в зависимости от того, выполненяется ли условие. Далее вызвал функцию, входным аргументом котороой будет являться переменная, которую я задал до этого (рис. 5).

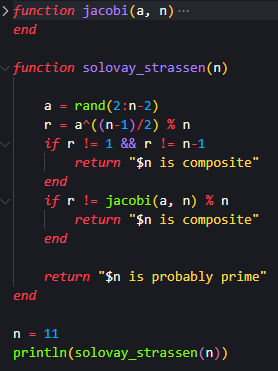


Рис. 5: Функция алгоритма теста Соловэя-Штрассена

И получил следующий результат (рис. 6):

Результат выполненного кода (3)

Рис. 6: Результат выполненного кода (3)

## 4.4 Алгоритм теста Миллера-Рабина

Я создал функцию алгоритма теста Миллера-Рабина, требующую число n >= 5, которое провряется на простоту. Будет выводиться сообщение о том, каким число является, в зависимости от того, выполненяется ли условие. Далее я создал переменную, которая будет являться входным параметром вызываемой функции (рис. 7):

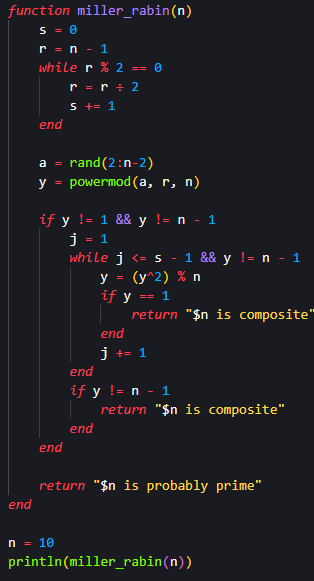


Рис. 7: Функция алгоритма теста Миллера-Рабина

И получил следующий результат (рис. 8):

Результат выполненного кода (4)

Рис. 8: Результат выполненного кода (4)

# 5 Выводы

Я реализовал алгоритмы вероятностной проверки чисел на простоту.

# Список Литературы

1. [Julia - Control Flow](https://docs.julialang.org/en/v1/manual/control-flow/)
2. [Julia - Mathematical Operations](https://docs.julialang.org/en/v1/manual/mathematical-operations/)
3. [Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone - Handbook of Applied Cryptography](https://cacr.uwaterloo.ca/hac/)