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

Вероятностные алгоритмы проверки чисел на простоту

Выполнил: Мануэл Марсия Педру,
НФИмд-02-25, 1032255503

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Изучить шифры простой замены и научиться их реализовывать.

2 Выполнение лабораторной работы

#2.1 Реализация вычисления символа Якоби Символ Якоби — теоретико-числовая функция двух аргументов, введённая К. Якоби в 1837 году. Является квадратичным характером в кольце вычетов. 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.

#2.2 Реализация алгоритма, реализующего тест Ферма Тест простоты Ферма в теории чисел — это тест простоты натурального числа ⊠, основанный на малой теореме Ферма.

#2.3 Реализация алгоритма, реализующего тест

Соловэя-Штрассена 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 состав- ных чисел с некоторой вероятностью он может дать неверный ответ. Основное 8 преимущество теста заключается в том, что он, в отличие от теста Ферма, распо- знает числа Кармайкла как составные.

#2.4 Реализация алгоритма, реализующего тест Миллера-Рабина Тест Миллера—Рабина — вероятностный полиномиальный тест простоты. Тест Миллера—Рабина, наряду с тестом Ферма и тестом Соловея—Штрассена, позво- ляет 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.

Выполним реализацию этого алгоритма на языке Python (рис. 2.1):

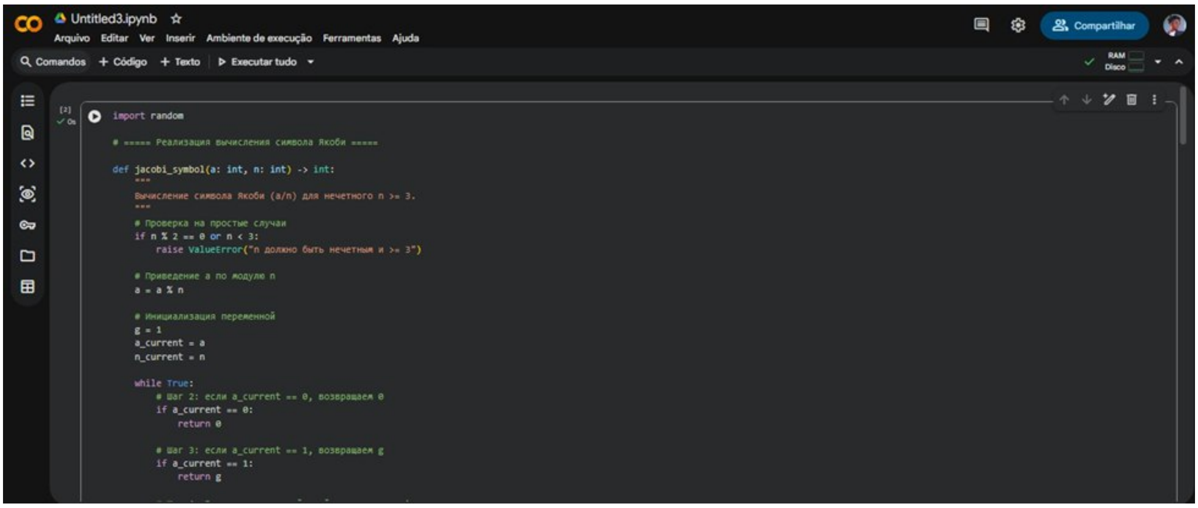

Рис. 2.1: Реализация вычисления символа Якоб

Проверим работу алгоритма (рис. 2.2):

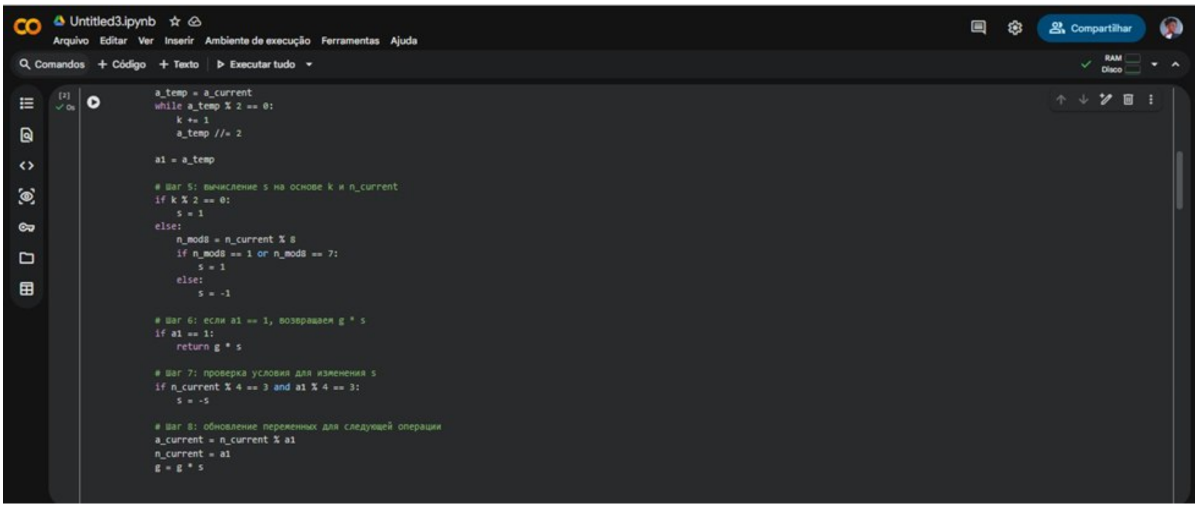

Рис. 2.2: Тест Соловея-Штрассена

Проверим работу алгоритма (рис. **??**):

[Тест Соловея-Штрассена] (рис. **??**):](image/3.png){ #fig:003 width=100% height=100% }

Выполним реализацию этого алгоритма на языке Python (рис. 2.3):

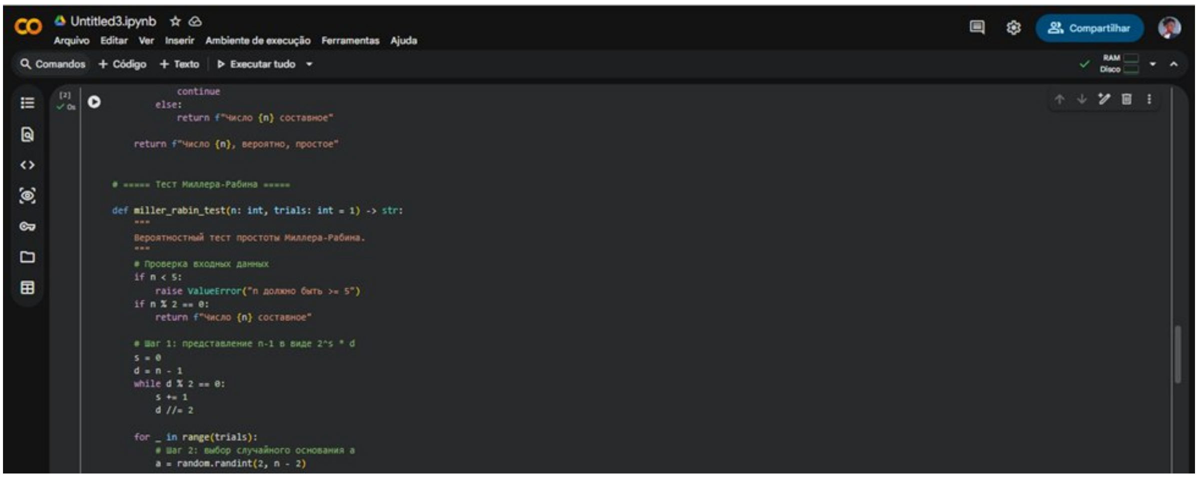

Рис. 2.3: Тест Миллера-Рабина

Проверим работу алгоритма (рис. 2.4):

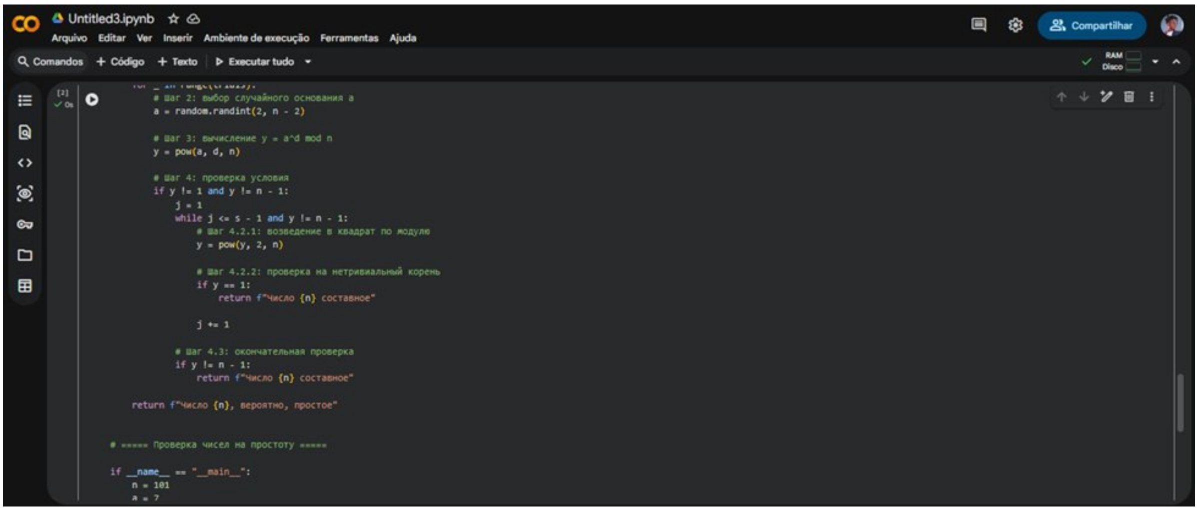

Рис. 2.4: Проверка чисел на простоту

Проверим работу алгоритма (рис. **??**):