

Лабораторная работа №5 — Вероятностные алгоритмы проверки чисел на простоту

Фатеева Елизавета Артёмовна — НПИмд-01-24 МОЗИиИБ |

Руководитель: Кулябов Д.С. — 7 ноября 2025 г.



Зачем проверять простоту в криптографии?

Криптография опирается на большие простые числа (обычно ≥ 1024 бит). Полный перебор делителей экспоненциален и невыполним на практике. Вероятностные тесты дают практическое решение:

- Быстро — работают за полиномиальное время
- Надёжно — контролируемая вероятность ошибки
- Практично — применяются в OpenSSL, GMP, Java



Основные свойства вероятностных тестов

1

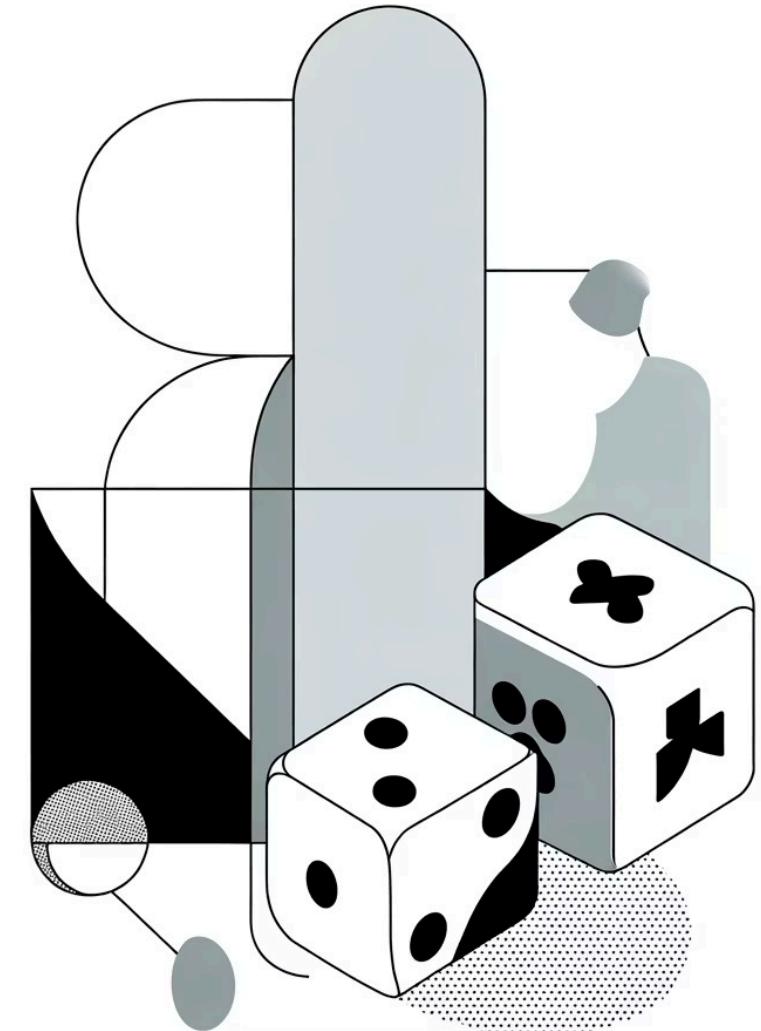
Детерминированный отказ

Если тест не пройден — число точно составное. Это важное свойство для фильтрации.

2

Вероятное принятие

Если тест пройден — число «вероятно простое» с оценимой ошибкой, уменьшаемой повторениями.



Реализованные тесты — обзор

Ферма

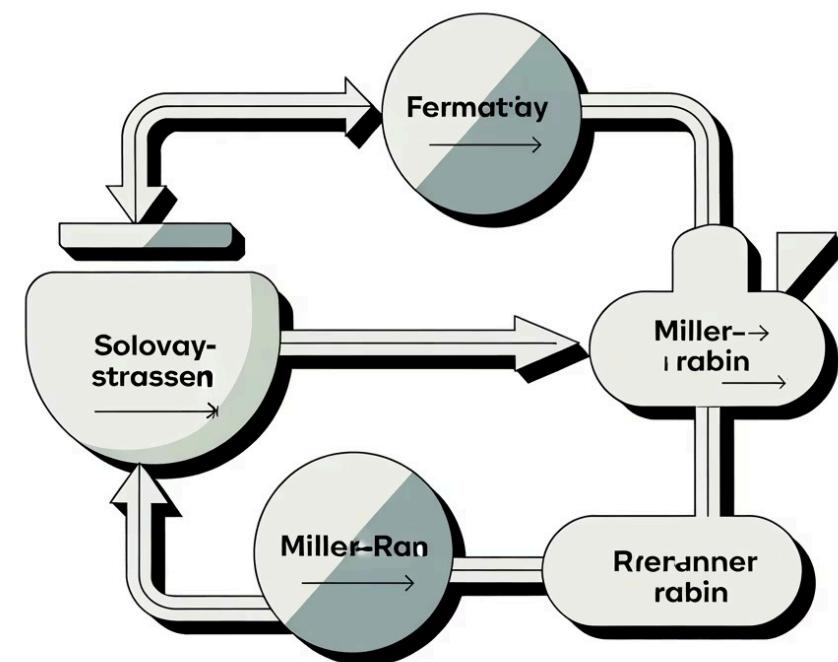
$a^{n-1} \equiv 1 \pmod{n}$. Быстрый фильтр, но не гарантированный: могут пройти Carmichael числа.

Миллер–Рабин

Разложение $n-1 = 2^s \cdot d$; последовательно проверяются квадраты. Практически стандарт — высоко надёжен.

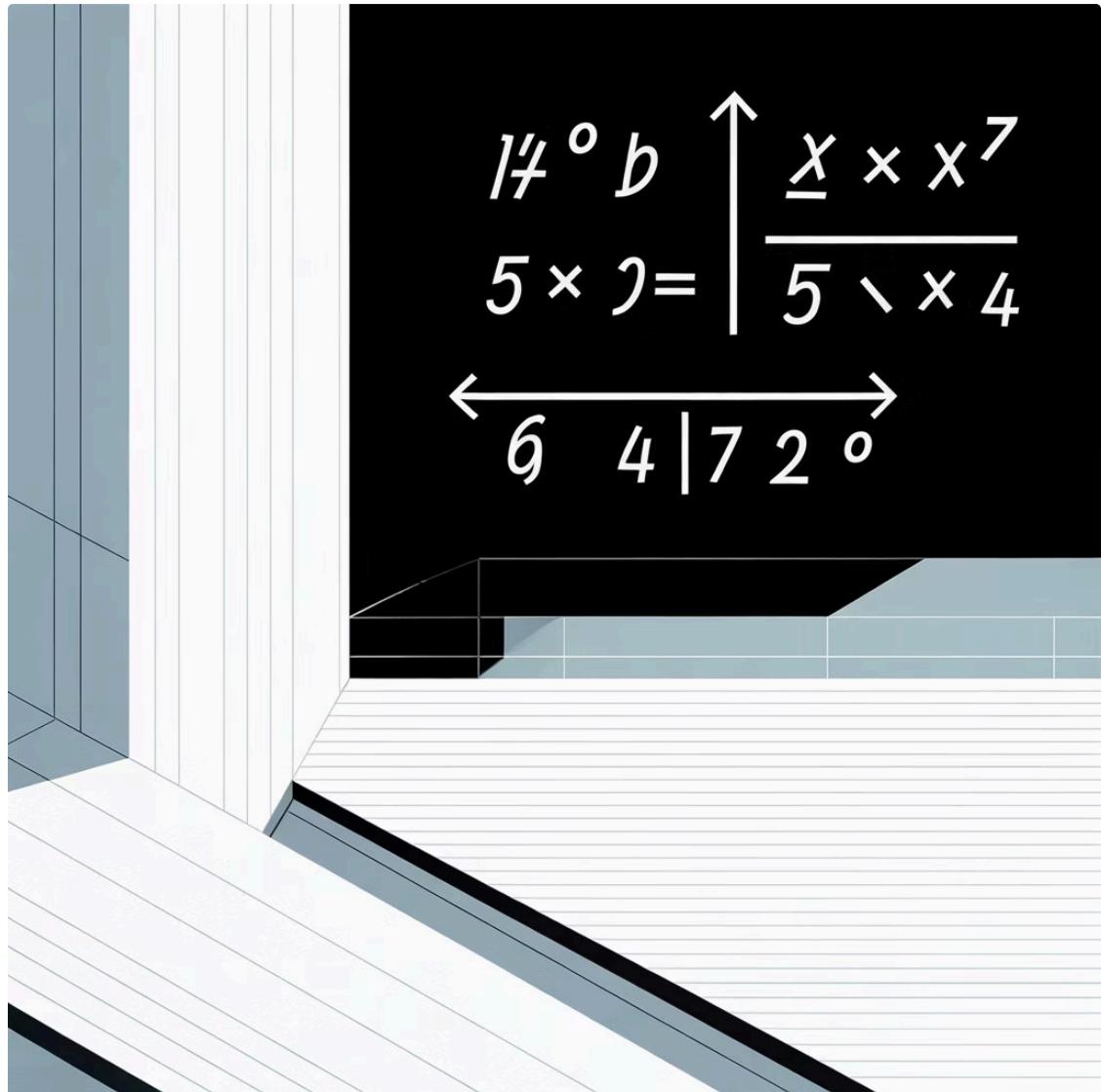
Соловей–Штрассен

Использует символ Якоби: $a^{(n-1)/2} \equiv (a|n) \pmod{n}$.
Лучше Ферма, но реже применяется на практике.



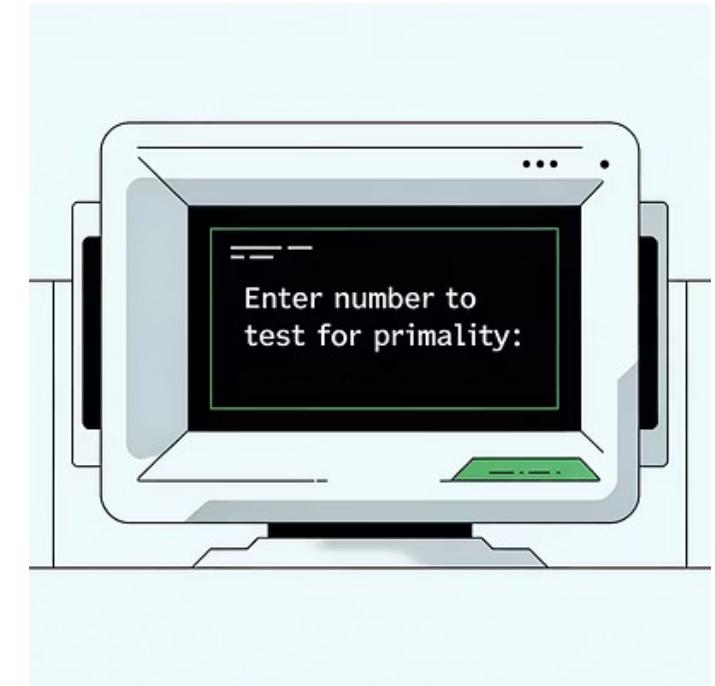
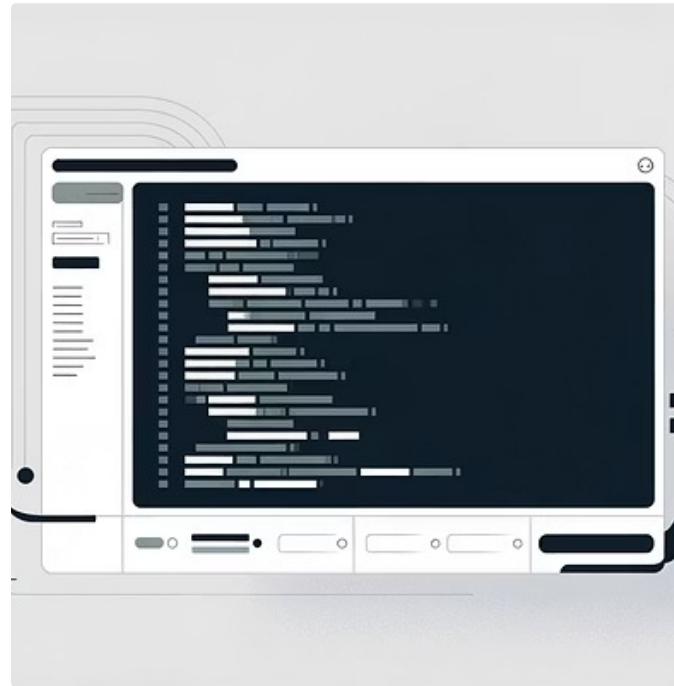
Символ Якоби — математическая подсказка

Символ Якоби — обобщение символа Лежандра для составных модулей. Возвращает значения -1 , 0 или $+1$ и вычисляется без факторизации, используя операции взаимности и деление на 2 . Быстро и детерминированно, служит «мотором» Соловея-Штрассена.

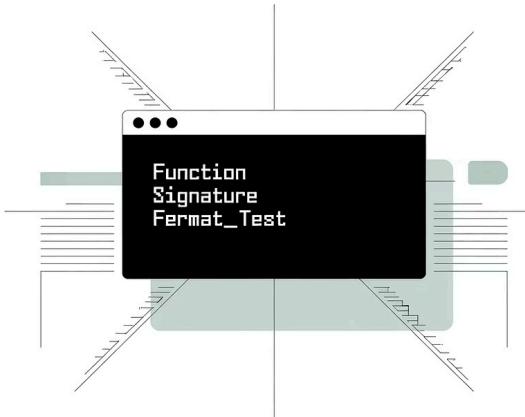


Архитектура кода — интерактивный тестер

Интерактивный тестер реализует последовательную верификацию: быстрые фильтры → более строгие тесты. Интерфейс содержит заголовки функций и логику вызовов без детального кода, для учебного демонстрационного примера.



Ключевые функции (обзор)



fermat_test(n, trials=5)

Работает с Int (для демонстрации),
использует powermod, возвращает
Bool, учитывает граничные случаи и
малые делители.



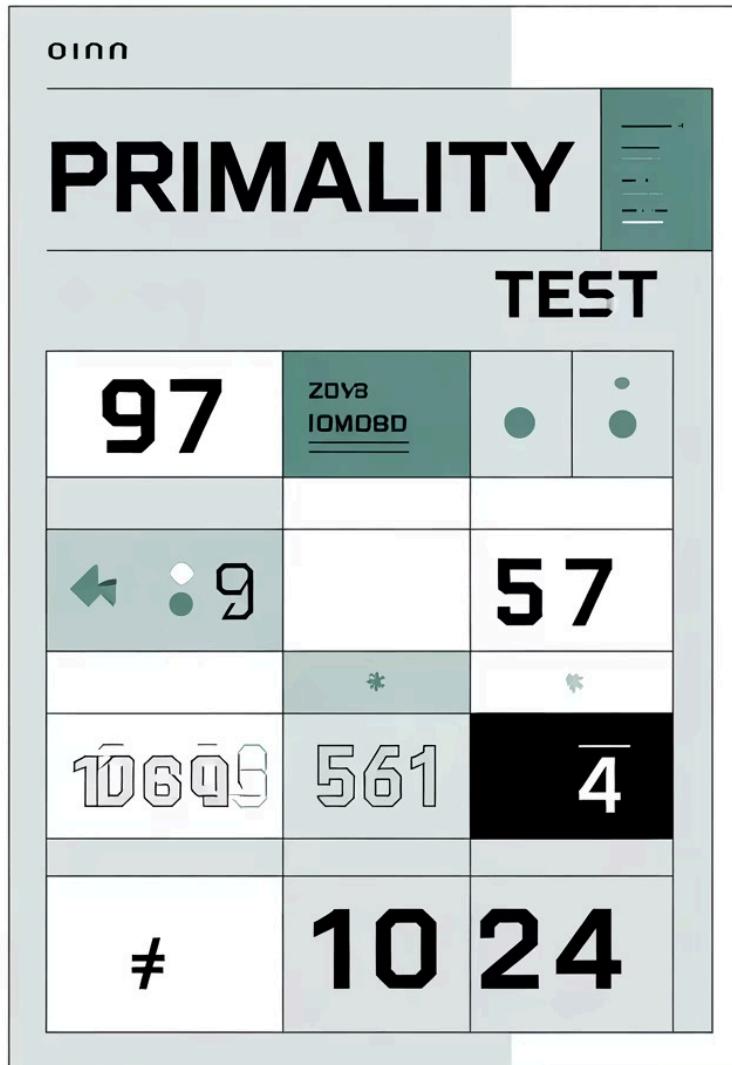
solvay_strassen_test(n, trials=5)

Вычисляет символ Якоби,
сравнивает с степенной проверкой,
детерминированные негативные
ответы.



miller_rabin_test(n, trials=5)

Разложение $n-1=2^s \cdot d$,
последовательные проверки
оснований, высокая практическая
надёжность.



Тестирование — набор чисел и результаты

Использовались контрольные множества: простые (97, 1009), обычные составные (9, 77), Carmichael (561), чётные (4, 1024). Результаты сводятся к понятным диагнозам: «составное» или «вероятно простое».

Демонстрация: случай 561 (Carmichael)

Пять итераций, сводка поведения тестов:

Итерации	Тест Ферма	Соловей–Штрассен	Миллер–Рабин
1–5	✓ вероятно простое	✗ составное	✗ составное

Вывод: 561 составное; Ферма уязвим к числам Кармайкла, Соловей–Штрассен и Миллер–Рабин обнаруживают составность.

Выводы и дальнейшие шаги

Корректность реализации

Все три теста реализованы корректно: обработка граничных случаев, powermod, модульность.

Практическая рекомендация

Миллер–Рабин — стандарт для практических приложений; комбинируйте с быстрыми фильтрами.

Академическая ценность

Код годится для учебных целей: читаемость, тесты, расширяемость до BigInteger при необходимости.

Спасибо за внимание! Вопросы? Код и отчёт — в приложении; готова принять замечания и внести доработки.

