СРАВНЕНИЕ АЛГОРИТМОВ ТЕСТИРОВАНИЯ ЧИСЕЛ НА ПРОСТОТУ. ПОСТРОЕНИЕ ОПТИМАЛЬНОГО АЛГОРИТМА

Кривошеев А. В., Совпель Д. С.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Стройникова Е. Д. — ассистент кафедры информатики

Целью работы является рассмотрение различных видов алгоритмов определения простоты чисел, выявление оптимальных алгоритмов для конкретных наборов данных, нахождение оптимизационных шагов для конкретных тестов на простоту. Основные объекты изучения — истинные и вероятностные тесты на простоту, их различия и специфика полученных результатов для родственных тестов.

Большие простые числа имеют весомое практическое применение в криптографических алгоритмах. Эффективность множества современных шифров зависит от того, насколько простое число велико. Например, для шифра RSA, на данный момент использующего открытый 2048-битный ключ, состоящий из произведения двух простых чисел. Увеличение мощности технических средств может позволить факторизовать открытый ключ (на данный момент максимальное большое факторизованное число RSA-768 содержит 232 цифры) [1], что приведет к небезопасности шифров, основанных на открытом ключе. Поэтому эффективный поиск больших простых чисел способен улучшить уже существующие шифры.

Тесты на простоту — вид алгоритмов, целью которых является определение простоты заданного числа. Среди тестов различают два подвида: истинные и вероятностные тесты.

Истинные тесты позволяют точно определить простоту числа. Однако они обладают рядом существенных недостатков, ограничивающих их применение. Среди них: условность — тест работает лишь для ряда заранее определенных чисел (тесты для чисел Мерсенна, Ферма, Прота), малая скорость обработки данных — полиномиальное время. Среди этих тестов в нашей работе были рассмотрены и оптимизированы следующие тесты:

- Тест Люка, основанный на одноименной теореме, работающий за полиноминальное время. В данном тесте определяется простота числа $2^p - 1$ за полиномиальное время от битовой длины числа p. Именно благодаря данному тесту числа Мерсенна почти всегда были самыми большими из найденных простых чисел.
- Тест Адлемана — Померанса — Румели — универсальный тест, работающий за $O(\log n^{c * \log\log n})$. Данный алгоритм может быть применим на практике в связи с медленнорастущей сложностью.
- Тест Агравала — Каяла — Саксены — универсальный тест, основанный на обобщенной теореме Ферма, работающий за $O((\log n)^6)$. Главная особенность данного алгоритма состоит в том, что он полиномиален, универсален, детерминирован и безусловен в отличие от всех предыдущих алгоритмов [2].
- Тесты с использованием эллиптических кривых. Данные тесты являются одними из самых быстрых из известных методов проверки числа на простоту. В худшем случае сложность такого теста будет составлять $O((\log n)^{5+\sum})$. И для некоторых случаев этот показатель может быть улучшен до $4 + \sum$.

В свою очередь результатом вероятностных тестов служит лишь вероятность простоты числа. Главным преимуществом таких тестов над истинными является их скорость обработки данных (например алгоритм Миллера — Рабина со сложностью $O((\log n)^3)$ [3]. Однако в силу вероятности результата ответом могут стать так называемые псевдопростые числа — числа, которые прошли тест на простоту, но таковыми не являются. Для уменьшения частоты появления подобных чисел используются раунды — повторные проверки числа этим же тестом. Примером псевдопростых чисел являются числа Кармайкла. В работе были рассмотрены следующие вероятностные тесты:

- Тест Ферма, основанный на одноименной теореме, работающий за $O(k * \log n * \log n)$.
- Тест Миллера — Рабина, использующий теорему Рабина о свидетелях простоты, с минимальной скоростью работы $O(k * (\log n)^3)$.
- Тест Соловея — Штрассена, основанный на символе Якоби, со скоростью работы $O((\log n)^3)$.

Результатом нашей работы стал универсальный оптимизационный алгоритм, полученный в процессе анализа специфики каждого теста для определенных числовых данных. Данный алгоритм аналогичен тесту BPSW, в основе которого лежит использование нескольких типов тестов. На рисунке 1 представлен график зависимости скорости работы теста BPSW от размера простого числа. Данная методика позволяет добиться уменьшения вычислительной сложности.

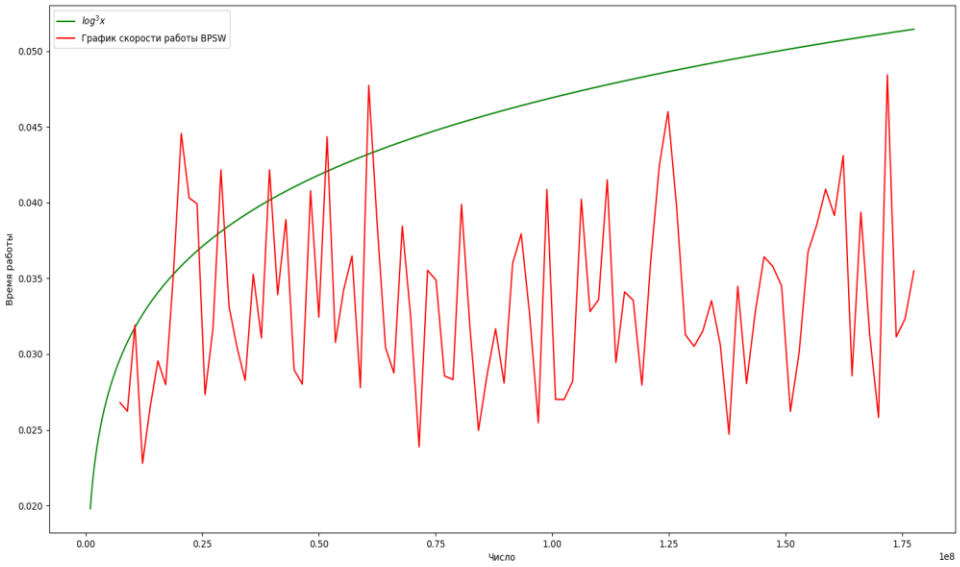

Рисунок 1 — График зависимости скорости работы алгоритма BPSW от размера простого числа

Список использованных источников:

1. Factorization of a 768-bit RSA Modulus / Th. Kleinjung [et al.] // Advances in Cryptology — CRYPTO 2010 : 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15—19, 2010. — Santa Barbara, 2010. — P. 333—350.
2. Бараш, *Л. Ю.* Алгоритм AKS проверки чисел на простоту и поиск констант генераторов псевдослучайных чисел / Л. Ю. Бараш // Безопасность информационных технологий. — 2005. — №2. — С. 27—38.
3. Шнайер, Б. Прикладная криптография / Б. Шнайер ; пер. с англ. — М. : Триумф, 2013. — 816 с.

ВИЗУАЛИЗАЦИЯ РЕШЕНИЙ НЕКОТОРЫХ МАТЕМАТИЧЕСКИХ ЗАДАЧ В MAPLE

Кузнечик В. А., Милинкевич М. И.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Калугина М. А. - к. ф.-м. н., доцент

В работе представлены результаты изучения средств и технологий системы компьютерной алгебры Maple и их применения для визуализации решений некоторых математических проблем. Перечислены примеры задач, для которых созданы процедуры поэтапного решения с элементами анимации: приведение уравнений кривых и поверхностей второго порядка к каноническому виду, разложение функции в ряд, решения систем дифференциальных уравнений, решение краевой задачи на колебания поверхности барабана и построение векторного поля.

Наглядность поставленной задачи является важной частью не только для понимания процесса решения, но и для исследования его в динамике, которая легче воспринимается зрительно. Анимация этапов решения, кроме этого, даёт выразительное представление о скорости явления. Студентам визуализация позволяет находить закономерности, систематизировать найденные решения и моделировать определённую ситуацию при исследовательской работе.

Одной из задач, геометрический смысл которой полезен при обучении и на практике, является задача приведения кривых и поверхностей второго порядка к каноническому виду. Ниже приведено ее решение на примере эллипса, заданного уравнением $5x^2 - 6xy + 5y^2 - 24x - 32 = 0$ (рис. 1 а), реализованное с помощью созданной процедуры. Для решения этой задачи в Maple можно построить матрицу квадратичной формы левой части уравнения, затем с помощью команд *EigenVectors* и *GramSchmidt* из пакета *LinearAlgebra* найти ее собственные векторы, из которых построить ортонормированный базис. Таким образом, получается базис, образованный поворотом исходного на некоторый угол (рис. 1 б). Ещё одним действием можно продемонстрировать параллельный перенос