Записки научных
семинаров ПОМИ
Том 479, 2019 г.

А. Г. Мошонкин, И. М. Хамитов

НОВЫЙ ВЕРОЯТНОСТНЫЙ ТЕСТ НА ПРОСТОТУ НАТУРАЛЬНЫХ ЧИСЕЛ

1. Введение. В этой работе мы даем краткий обзор наиболее известных тестов на простоту и представляем новый эффективный общий вероятностный тест. Мы также приводим некоторую оценку эффективности нового теста. Эта оценка уступает, например, оценке эффективности для теста Миллера–Рабина. Однако, как будет показано, есть веские основания полагать, что наша оценка весьма груба и реальная эффективность нового теста намного выше.

Построение больших простых чисел является древнейшей задачей. С появлением криптографии с публичным ключом эта задача приобрела практическое значение. Для многих криптосистем требуются "случайные" большие простые числа. Большими простыми числами в криптографических приложениях в настоящее время считаются числа размером в несколько тысяч битов (двоичных разрядов). Вычислительная практика показывает, что для чисел такого размера асимптотический закон распределения простых чисел служит хорошим приближением для плотности простых чисел в натуральном ряду ($\sim 1/\ln n$), то есть, что они довольно "плотно" располагаются в натуральном ряду "в среднем". По этой причине для нахождения простых чисел достаточно научиться их распознавать. Действительно, можно "случайно" выбирать натуральные числа нужного размера, и проверять их на простоту. Вероятность того, что нам будут встречаться только составные числа, с ростом числа попыток падает экспоненциально, поэтому простое число будет найдено довольно быстро. Часто "случайно" выбирают только первое число и проверяют на простоту его и все последующие за ним числа. На практике такой метод тоже работает вполне удовлетворительно. В связи с этим, тесты на простоту приобрели важное практическое значение, так как криптография с публичным ключом широко применяется в области информационных технологий, которые неумолимо проникают во все сферы нашей жизни.

Ключевые слова: простое число, тест на простоту.

Тесты на простоту (как и другие алгоритмы) бывают детерминированными и вероятностными. Детерминированный тест дает однозначный ответ на вопрос, является число простым или составным. До августа 2002 года не было известно общих, то есть, работающих с любыми натуральными числами, детерминированных тестов на простоту полиномиальной сложности. Были известны лишь специальные тесты, которые работали с числами специального вида. Например, тест Люка–Лемера работает с числами Мерсена, а тест Пепина – с числами Ферма [1]. Оба теста детерминированы и весьма эффективны. В работе индийских математиков Агравала, Каяла и Саксены [2] представлен общий детерминированный алгоритм проверки на простоту полиномиальной сложности. К сожалению, он недостаточно эффективен, и на практике пока используются эффективные вероятностные тесты.

Говоря о вероятностных тестах полиномиальной сложности, мы подразумеваем две различные ситуации. С одной стороны, время работы теста теоретически может быть экспоненциальным, хотя вероятность этого исчезающе мала. В этом случае мы говорим об ожидаемом времени работы. С другой стороны, тест может не давать однозначного ответа, он может лишь дать ответ, который будет верным с некоторой вероятностью. Возможно, конечно, и то и другое одновременно. Приведем примеры.

Специальный тест Люка [3] работает с числами n, для которых известно разложение $n-1$ на простые множители. Можно установить, будет ли n простым или составным, основываясь на следующем простом наблюдении. Число n будет простым тогда, и только тогда, когда найдется натуральное b, такое, что $b^{n-1} \equiv 1 \pmod{n}$, но $b^{(n-1)/p} \not\equiv 1 \pmod{n}$ для любого простого делителя p числа $n-1$, причем b в этом случае будет первообразным корнем по модулю n. Первообразных корней по модулю простого числа n достаточно много, а именно $\varphi(n-1)$, где φ – функция Эйлера, поэтому на практике для простого числа n мы довольно быстро методом проб и ошибок найдем первообразный корень. Теоретически нам может потребоваться экспоненциальное число проб, однако вероятность этого исчезающе мала. Таким образом, здесь идет речь о вероятностном тесте, с полиномиальным ожидаемым временем работы, который определяет простоту числа наверняка. Используя этот тест, можно строить большие простые числа следующим образом. Пусть мы уже построили доказуемо простые числа

$p_1, p_2, \cdots, p_k$. По теореме Дирихле о простых числах в арифметической прогрессии, прогрессия $1 + a \cdot \prod p_i$ при $a > 0$ содержит бесконечно много простых чисел, причем весьма вероятно, что наименьшее из них получается при небольшом значении a. Пусть теперь $n = 1 + a \cdot \prod p_i$ для небольшого положительного a. Так как a невелико, то нетрудно разложить его на простые множители, получить каноническое разложение для числа $n - 1$, и применить тест Люка.

Все же большинство вероятностных тестов на простоту легче работают с составными числами, нежели с простыми. Например, согласно малой теореме Ферма для простого числа n и произвольного целого числа a выполняется условие $a^n \equiv a \pmod{n}$, поэтому если мы найдем хотя бы одно a, для которого это условие не выполняется, то число n заведомо составное. Однако таким образом не для всякого составного числа удастся доказать, что оно является составным. Составные числа $561 = 3 \cdot 11 \cdot 17$, $1105 = 5 \cdot 13 \cdot 17$, $1729 = 7 \cdot 13 \cdot 19$ и еще бесконечно много других (числа Кармайкла) [4] обладают тем свойством, что $a^n \equiv a \pmod{n}$ для всех целых чисел a. Поэтому тест на простоту, основанный на малой теореме Ферма, нельзя считать надежным. Более сильная версия теоремы Ферма, свободная от этого дефекта, утверждает, что $a^{(n-1)/2} \equiv (\frac{a}{n}) = \pm 1 \pmod{n}$ для всех нечетных простых чисел n и всех целых чисел $a \not\equiv 0 \pmod{n}$. Здесь $(\frac{a}{n})$ обозначает символ Якоби для целого a и нечетного положительного n. Можно доказать, что для нечетного составного n не более половины, и обычно значительно меньше вычетов по модулю n удовлетворяют этому условию. Значения $a^{(n-1)/2}$ и $(\frac{a}{n})$ могут быть эффективно вычислены: первое – повторным возведением в квадрат и умножением по модулю n с использованием двоичного представления $(n-1)/2$, а второе – посредством закона взаимности для символов Якоби. Отсюда мы приходим к простому практическому способу распознавания непростоты: выбираем случайным образом вычет a по модулю n до тех пор, пока не встретится такой, который не удовлетворяет вышеприведенному условию. Если было испробовано несколько сотен значений a, и для всех из них условие выполняется, то вероятность того, что n составное будет исчезающе малой. Это тест Соловэя–Штрассена [5].

Тест Миллера–Рабина [6] более эффективен. Пусть n нечетно, и $n = 2^s \cdot r + 1$, где r нечетно. Выберем случайный ненулевой вычет b по модулю n, и пусть $c = b^r \pmod{n}$. Рассмотрим последовательность $c_1, c_2, \ldots, c_{s+1}$, в которой $c_1 = c$, и $c_{i+1} = {c_i}^2 \pmod{n}$. Для простого n

в этой последовательности обязательно встретится 1, после которой, очевидно, будут следовать только единицы, причем, если c_i первая единица в этом ряду, то либо $i = 1$, либо $c_{i-1} = -1 \pmod n$. Можно доказать, что для составного n не более четверти вычетов удовлетворяют этому условию. Если справедлива расширенная гипотеза Римана, то тест Миллера–Рабина будет детерминированным, потому что по модулю этой гипотезы для составного n наименьший вычет, который установит его непростоту не будет превосходить $2(\ln n)^2$.

2. Новый эффективный вероятностный тест. Здесь мы приводим новый эффективный общий вероятностный тест на простоту. Основная идея состоит в следующем. Пусть n – нечетное натуральное число большее 1. На предварительном этапе мы проверяем, является ли n степенью натурального числа, то есть представляется ли оно в виде

$$n = a^b \quad \text{для} \quad a, b \in \mathbb{N}, \quad b \geqslant 2. \tag{1}$$

Из условия (1) следует, что $b \leqslant \log_2 n - 1$, поэтому для каждого b из промежутка $[2, \log_2 n - 1]$ достаточно проверить, является ли n b-той степенью натурального числа. Для каждого b такая проверка может быть эффективно осуществлена, например, методом половинного деления. Если n представимо в виде (1), то оно заведомо составное (мы даже получаем частичное разложение), и тест завершает работу. Далее будем считать, что n не представимо в виде (1).

На следующем этапе мы выбираем случайным образом ненулевой вычет $r \pmod n$. Если r и n не взаимно просты, то мы так же получаем частичное разложение для n и тест завершает работу.

Известно несколько эффективных (вероятностных) алгоритмов для нахождения квадратного корня из квадратичного вычета по модулю простого числа. Применим какой либо из них к элементу $r^2 \pmod n$, не обращая внимание на то, что n может быть составным. Для простого n алгоритм всегда доработает до конца, и на выходе будет $\pm r \pmod n$. Давайте посмотрим, что может случиться при составном n. Во первых, алгоритм может не пройти какой то из своих шагов, например, некоторые шаги могут быть корректно определены только для простых n. В этом случае n составное, и тест завершает работу. Во вторых, алгоритм дойдет до конца, но на выходе будет элемент, отличный от $\pm r \pmod n$. В этом случае n так же составное, и тест завершает работу. Наконец, алгоритм дойдет до конца, и на выходе будет $\pm r$

$(\mathrm{mod}\ n)$. В этом случае n может быть как простым, так и составным. Покажем, что последняя возможность для составного n, не представимого в виде (1), может реализоваться не более чем для половины элементов $r\ (\mathrm{mod}\ n)$, взаимно простых с n.

Действительно, если n составное, не представимое в виде (1), то оно представимо в виде

$$n = k \cdot l \text{ для некоторых взаимно простых } k,\ l \geqslant 2. \tag{2}$$

В этом случае, согласно Китайской теореме об остатках

$$\mathbb{Z}/n\mathbb{Z} \cong \mathbb{Z}/k\mathbb{Z} \times \mathbb{Z}/l\mathbb{Z}. \tag{3}$$

Элемент r^2 будет квадратом как в $\mathbb{Z}/k\mathbb{Z}$, так и в $\mathbb{Z}/l\mathbb{Z}$, причем в каждом из этих колец у него будет по крайней мере два различных квадратных корня. Комбинация каждого из квадратных корней в $\mathbb{Z}/k\mathbb{Z}$ с каждым из квадратных корней в $\mathbb{Z}/l\mathbb{Z}$ при подъеме по изоморфизму (3) дает квадратный корень в $\mathbb{Z}/n\mathbb{Z}$, причем все эти корни будут различны, так что в $\mathbb{Z}/n\mathbb{Z}$ имеется не менее четырех различных корней.

При формальном описании теста мы будем использовать алгоритм извлечения квадратного корня из квадратичного вычета по модулю простого числа p, каждый шаг которого определен для любого нечетного n, это делается для того, чтобы изложение было более прозрачным. Сначала опишем этот алгоритм.

Пусть p – нечетное простое число, и c – квадратичный вычет по модулю p. Найдем b такое, что $b^2 - 4c$ будет квадратичным невычетом. Этот шаг имеет вероятностный характер, так как теоретически нам может потребоваться экспоненциальное число попыток, однако вероятность этого исчезающе мала, потому, что $b^2 - 4c$ будет квадратичным невычетом не менее, чем для половины ненулевых вычетов b. Для такого b многочлен $f(x) = x^2 + bx + c$ будет неприводим над $\mathbb{F}_p$. Присоединение корня этого многочлена является квадратичным расширением, изоморфным $\mathbb{F}_p[x]/f(x)$, причем класс вычетов, содержащий многочлен x будет корнем $f(x)$ в этом расширении. Другим корнем будет сопряженный элемент, то есть класс вычетов, содержащий многочлен x^p. По теореме Виета произведение корней равно c, то есть $x^{p+1} \equiv c\ (\mathrm{mod}\ f(x))$. Это значит, что $x^{(p+1)/2}\ (\mathrm{mod}\ f(x))$ будет константой, квадрат которой равен c.

Теперь приведем формальное описание нашего теста.

Пусть n – нечетное число.

1. Если n представимо в виде (1), то оно составное, и тест завершает работу.

2. Берем случайный ненулевой вычет r по модулю n. Если он не взаимно прост с n, то n составное, и тест завершает работу.

3. Пусть $c = r^2 \pmod n$. Находим b такое, чтобы выполнялись условия $b^2 - 4c \neq 0 \pmod n$, и $\left(\frac{b^2-4c}{n}\right) = 0$ или -1, где $(\ )$ – символ Якоби. Если $\left(\frac{b^2-4c}{n}\right) = 0$, то n составное, и тест завершает работу. В противном случае переходим к следующему пункту. (Этот пункт имеет вероятностный характер. Теоретически, для нахождения b нам может потребоваться экспоненциальное число попыток, однако вероятность этого исчезающе мала, потому, что подходящих нам b не менее половины).

4. Пусть $f(x) = x^2 + bx + c$. Если $x^{(n+1)/2} \not\equiv \pm r \pmod{f(x)}$, то n составное, и тест завершает работу. В противном случае возвращаемся к пункту 2.

Если мы проверили несколько сотен r, и ни одно из них не показало, что n составное, то вероятность того, что оно действительно составное будет исчезающе малой.

3. Оценка эффективности. Предварительный этап теста (пункт 1) проходит очень быстро. Для каждого $b \in [2,\ \log_2 n - 1]$ мы должны проверить, является ли n b-той степенью натурального числа. Эту проверку для конкретного b можно эффективно сделать методом половинного деления, при этом нам потребуется не более $\log_2 n - 1$ возведений в небольшую степень b.

Пункт 3 так же проходит очень быстро, в среднем нам потребуется не более двух вычислений символа Якоби, которые могут быть проделаны посредством закона взаимности для символов Якоби. Таким образом, эффективность теста определяется эффективностью выполнения пункта 4.

Возведение в степень можно осуществить бинарным методом, который требует m возведений в квадрат и k умножений линейных многочленов по модулю $f(x)$, где m – количество знаков в двоичном представлении $(n+1)/2$, а k – количество единиц в таком представлении. Обычное возведение в квадрат линейного многочлена требует трех модулярных (по модулю n) умножений. Для приведения результата по

модулю $f(x)$ потребуется еще два модулярных умножения. Произведение требует четырех модулярных умножений. Для приведения результата по модулю $f(x)$ так же потребуется еще два модулярных умножения. В итоге получаем, что выполнение алгоритма требует $5m + 6k$ модулярных умножений (для алгоритма Чиполлы [7] требуется $4m+6k$ таких операций).

При чуть более аккуратном рассмотрении можно заметить, что для возведения в квадрат линейного многочлена требуется менее $5m$ модулярных умножений. Мы считали, что обычное возведение в квадрат, это три модулярных умножения, то есть три обычных умножения и три приведения результатов по модулю n. Но в действительности два из этих умножений являются возведениями в квадрат, а возведение в квадрат асимптотически вдвое эффективнее, чем умножение. Мы не учитывали здесь операции модулярного сложения, тат как их выполнение требует значительно меньшего времени, чем выполнение операций модулярного умножения.

Приведенный нами алгоритм извлечения квадратного корня можно еще несколько ускорить, если вместо многочлена $x^2 + bx + c$ использовать многочлен $x^2 + x + b^{-2}c$. Это соответствует замене $x \leftrightarrow bx$, что приведет к нахождению квадратного корня из b^2c. Преимущество состоит в том, что у этого многочлена уже два из трех коэффициентов равны 1, и приведение квадратного многочлена по модулю $x^2+x+b^{-2}c$ потребует всего одного модулярного умножения. Общее же число операций модулярного умножения сократится до $4m + 5k$.

Дальнейшее усовершенствование возможно, если воспользоваться интерполяцией. Все многочлены, с которыми мы здесь работаем, имеют степень не больше двух, поэтому однозначно определяются своими значениями в любых трех различных точках. Удобнее всего задавать такой многочлен g тремя значениями – $g(-1), g(0)$ и $g(1)$. Заметим, что поскольку степень g не превосходит двух, то по данному представлению сразу же определяется коэффициент при x^2 в обычном представлении, который равен $(g(-1) + g(1) - 2g(0))/2$. Умножение линейных многочленов в таком представлении требует трех модулярных умножений, и приведение результата (у которого известен старший коэффициент) по модулю $x^2+x+b^{-2}c$ еще одного модулярного умножения. В итоге общее число необходимых операций модулярного умножения сократится до $4m + 4k$. Для простых чисел n вида $4t + 3$ квадратный корень из квадратичного вычета a может быть получен очень быстро

прямым вычислением $a^{(n+1)/4} \pmod n$, для таких n один цикл нашего теста требует числа операций, эквивалентного числу операций одного цикла теста Миллера–Рабина.

Эффективность теста на простоту определяется не только временем прохождения одного цикла, но и вероятностью того, что для составного числа один цикл определит, что это число действительно составное. Для теста Миллера–Рабина эта вероятность не меньше $3/4$, для теста Соловэя–Штрассена и для нашего теста, эта вероятность не менее $1/2$.

4. Обсуждение. Как уже говорилось, пункты 3 и 4 можно заменить любым другим алгоритмом извлечения квадратного корня, возможно, с проверкой корректности некоторых его шагов, поэтому нахождение более эффективных методов извлечения квадратного корня будет способствовать увеличению эффективности нашего теста.

Как отмечалось в предыдущем разделе, вероятность того, что для составного числа один цикл нашего теста определит, что оно действительно составное с вероятностью не меньшей $1/2$. Однако есть основания полагать, что эта вероятность существенно выше. Дело в том, что $1/2$ – это не просто оценка снизу для вероятности обнаружения того, что число составное, а оценка для условной вероятности при условии наступления события A, которое состоит в том, что цикл нашего теста прошел пункт 4, и на выходе этого пункта получился элемент, равный квадратному корню из c. Однако если бы вероятность события A была сколько-нибудь существенной, то это давало бы для данного составного n эффективный способ извлекать квадратные корни по модулю n. Так как извлечение квадратного корня полиномиально эквивалентно факторизации, то мы имели бы эффективный алгоритм факторизации данного “случайного” большого составного числа n. Такой сказочный бонус в виде алгоритма факторизации в дополнение к скромному тесту на простоту представляется незаслуженной удачей, на которую надеяться не представляется возможным при текущем уровне знаний в этой области математики.

5. Усовершенствование. Приведенный тест можно усовершенствовать для нечетных чисел n, у которых “нечетная” часть числа $n-1$ достаточно велика, то есть для чисел вида

$$n = 2^s r + 1, \tag{4}$$

где r – достаточно большое нечетное число (скажем, его запись в двоичной системе счисления состоит из нескольких сотен битов, конечно, "случайные" большие нечетные числа удовлетворяют этому условию).

Это усовершенствование возможно благодаря тому, что для простых чисел вида (4) можно очень быстро вычислять квадратные корни хотя и не из всех квадратичных вычетов, но из достаточно большого (r штук) их количества.

Пусть нечетное простое число n имеет вид (4). Рассмотрим цепочку гомоморфизмов групп

$$(\mathbb{Z}/n\mathbb{Z})^* \to V_{s-1} \to V_s,$$

здесь V_{s-1} – множество 2^{s-1}-тых степеней, а V_s – множество 2^s-тых степеней группы $(\mathbb{Z}/n\mathbb{Z})^*$. Первый гомоморфизм, это возведение в степень 2^{s-1}, а второй – возведение в квадрат.

Если x – "случайный" элемент $(\mathbb{Z}/n\mathbb{Z})^*$, то возведение его в степень 2^{s-1} даст "случайный" элемент V_{s-1}, квадрат которого попадет в V_s. Множество V_s содержит r элементов, и для любого элемента a этого множества корень квадратный из a находится очень быстро прямым вычислением $b = a^{(r+1)/2}$. Действительно, $b^2 = a^{(r+1)} = a^r \cdot a = a$ (здесь мы используем то, что $a \in V_s$, и поэтому $a^r = 1$).

Теперь ясно, как усовершенствовать наш тест на простоту для нечетных n, с достаточно большой нечетной частью числа $n-1$ (еще раз напомним, что "случайные" большие нечетные числа удовлетворяют этому условию). На вход каждого цикла теста нужно подавать 2^{s-1}-тую степень случайного элемента $(\mathbb{Z}/n\mathbb{Z})^*$, а квадратный корень из его квадрата вычислять так, как это описано в предыдущем абзаце.

Авторы выражают искреннюю благодарность Н. В. Проскурину за ряд полезных замечаний.

Список литературы

1. R. M. Robinson, *Mersenne and Fermat Numbers.* — Proc. Amer. Math. Soc. **5** (1954), 842–846.
2. M. Agrawal, N. Kayal, and N. Saxena, *Primes is in P.* — Ann. of Math. (2) **160**, No. 2 (2004), 781–793.
3. E. Lucas, *Theorie des fonctions numeriques simplement periodiques. (French)* — Amer. J. Math. **1**, No. 2–4 (1878), 184—240, 289–321.
4. W. R. Alford, A. Granville, and C. Pomerance, *There are infinitely many Carmichael numbers.* — Ann. of Math. **139** (1994), 703–722.

5. R. Solovay and V. Strassen, *A fast Monte-Carlo test for primality.* — SIAM J. Comput. **6** (1977), No. 1, 84–85.
6. M. O. Rabin, *Probabilistic algorithm for testing primality.* — J. Number Theory **12** (1980), No. 1, 128–138.
7. M. Baker, *Cipolla's algorithm for finding square roots mod p.* http://people.math.gatech.edu/ mbaker/pdf/cipolla2011.pdf

Moshonkin A. G., Khamitov I. M. New probabilistic primality test.

In this paper we present a new general probabilistic test for primality. The estimated efficiency of our test turns out to be inferior to that of the Miller–Rabin test. However, we provide some heuristic arguments that our estimation of efficiency is quite rough. This allows us to expect that the real efficiency of our test is much greater.

С.-Петербургский государственный университет,
С.-Петербург, Россия
E-mail: AMoshonkin@gmail.com

С.-Петербургский государственный университет,
С.-Петербург, Россия
E-mail: Ildarspb@gmail.com

Поступило 19 июня 2019 г.