МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Тольяттинский государственный университет»

Институт математики физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование кафедры)

02.03.03 Математическое обеспечение и администрирование информационных систем

(код и наименование направления подготовки, специальности)

Технология программирования

(направленность (профиль)/специализация)

БАКАЛАВРСКАЯ РАБОТА

на тему Алгоритмы проверки простоты целых чисел

Студент	Ф.Д. Давлатбеков	
	(И.О. Фамилия)	(личная подпись)
Руководитель	Г.А. Тырыгина	
	(И.О. Фамилия)	(личная подпись)
Консультанты	Т.В. Маркелова	
	(И.О. Фамилия)	(личная подпись)

Допустить к защите

Заведующий кафедрой к.т.м, доцент, А.В. Очеповский _____

(ученая степень, звание, И.О. Фамилия) (личная подпись)

«_____»_____20_____г.

Тольятти 2018

Аннотация

Название бакалаврской работы – «Алгоритмы проверки простоты целых чисел».

Объект работы: простые целые числа.

Предмет исследования: критерии проверки простоты целых чисел.

Целью данной работы является сравнение критерии проверки простоты целых чисел.

Для реализации эти цели формулируются задачи:

1) Описание теоретических основ критерии проверки простоты целых чисел;

2) Построение алгоритмов проверки простоты целых чисел;

3) Сравнение критерии проверки простоты целых чисел;

Во введении обосновывается актуальность темы исследования, формулируются цели и задачи.

Теоретическим основам критериев простоты целых чисел посвящена первая глава.

Во второй главе исследуются разные алгоритмы для проверки простоты целых чисел.

В третьей главе сравниваются тести распознавание простоты целых чисел.

В бакалаврской работе пояснительная записка объемом 40 страниц, в которой 6 рисунков, список использованной литературы из 22 источников.

ABSTRACT

The title of the graduation work is "Algorithms for testing the simplicity of integers".

The object of the work: simple integers.

The subject of the study: criteria for verifying the simplicity of integers.

The goal of this paper is to compare the criteria for verifying the simplicity of integers.

To implement these goals, we formulate the tasks:

- Description of the theoretical bases of the criteria for testing the simplicity of integers;
- Construction of algorithms for testing the simplicity of integers;
- Comparison of the criteria for testing the simplicity of integers;

In the introduction, the urgency of the research topic is substantiated, goals and tasks are formulated, which must be solved to achieve the goal.

The first chapter describes the theoretical foundations of the criterion for testing the simplicity of integers.

The second chapter explores different algorithms for testing the simplicity of integers.

The third chapter compares the test to the recognition of the simplicity of integers.

Graduation work contains an explanatory note in the volume of 40 pages, includes 6 figures and a list of used literature, consisting of 20 sources.

Оглавление

Введение

Проблема защиты информации путем ее преобразования давно беспокоила человеческий разум. Криптография - того же возраст, что и история человеческого языка. Изначально сама запись была своего рода криптографической системой, поскольку в древних обществах она принадлежала только элите.

Простые числа применяются в криптосистемах с открытым ключом, поэтому вопрос об определении простоты числа является актуальной задачей.

Объект исследования – простые целые числа.

Предмет исследования – критерии простоты целых чисел.

Целью данной работы являются алгоритмы проверки простоты целых чисел.

Для реализации эти цели формулируются задачи:

1. Описание теоретических основ критериев простоты целых чисел;
2. Построение алгоритмов проверки простоты целых чисел;
3. Реализация алгоритмов проверки простоты;

Во введении обосновывается актуальность темы исследования, формулируется цель, а также задачи, вытекающие из постановки цели.

Первая глава повешена теоретическим основам критериев простоты целых чисел.

Во второй главе рассматриваются тесты простоты целых чисел.

В третьей главе реализуются алгоритмы тестов простоты целых чисел.

Глава 1. Теоретические основы критериев простоты целых чисел.

1.1 Распределение простых чисел в натуральном ряду

Использование простых чисел используются в алгоритмах защиты информации. Простые числа широко используются в криптографических алгоритмах с ассиметричным ключом, в алгоритмах электронной цифровой подписи.([15], [16], [11], [4] ,[8])

Простые числа играют большую роль в прикладных задачах, поэтому свойства множества всех простых чисел всегда привлекали интерес математиков, например, распределение простых чисел в натуральном ряду. ([17]-[22]) Как видно из таблиц простых чисел, простые числа распределены во множестве N неравномерно. Среди первых ста натуральных чисел встречается 25 простых чисел, в следующих ста числах насчитывается 21, в сорок девятой сотне - 8, в пятидесятой сотне- 15. Наблюдается тенденция постепенного уменьшение плотности распределения простых чисел. Для каждого натурального числа n можно указать n последовательных составных чисел:

$$(n + 1)! + 2, \ldots, (n + 1)! + (n + 1). \quad (1.1)$$

Французский математик Бертран предположил, что для всякого $n > 1$ между n и $2n-2$ найдется хотя бы одно простое число. Этот постулат Бертрана был доказан русским математиком П. Л. Чебышевым. При исследовании простых чисел пытались найти явные формулы, описывавшие множество простых чисел. Известно, что значение многочлена с целыми коэффициентами в целых точках не могут состоять только из простых чисел. Однако, был выстроен многочлен степени 25 из 26 переменных, для которых множество положительных значений в целых точках совпадает с множеством всех простых чисел.([14], [13], [12], [1], [9], [10])

В проблеме распределении простых чисел важные место занимает задача описания числовой функции π: $(1; +\infty)$, где число простых чисел на отрезке $[1, x]$ равно $\pi(x)$. Из бесконечности множества простых чисел

вытекает, что $\pi(x) \to \infty$, при $x \to \infty$ (теорема Евклида). Важной задачей является нахождение оценки порядка роста функции $\pi(x)$.

Французский математик А. М. Лежандр нашёл эмпирическую формулу

$$\pi(x) \approx \frac{x}{\mathrm{Ln}x - 1.08366} \qquad (1.2)$$

Опираясь на таблицы простых чисел Лежандр и Гаусс предположили, что функция π(x) асимптотически равна функции $\frac{x}{\mathrm{Ln}x}$. Эта гипотеза была доказана П. Л. Чебышевым. Из доказанных им теоремы следует, что если имеются пределы

$$\lim_{x\to\infty} \frac{\pi(x)}{\frac{x}{\mathrm{Ln}x}}, \lim_{x\to\infty} \frac{\pi(x)}{\int_2^x \frac{dt}{\mathrm{Ln}t}} \qquad (1.3)$$

то они равны 1.

Чебышев первым использовал функцию Эйлера $\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$, для исследования вопросов, связанных с распределением простых чисел. Функцию Эйлера (дзета-функцию) Г. Риман распространил на множество комплексных аргументов.

Приведем формулировку и доказательство теоремы Чебышева о приближении функции $\pi\ x$ к функцией $\frac{x}{\mathrm{Ln}x}$.

Теорема 1.1 (Чебышев).

Имеются подобные положительные числа $a < 1 < 2$, что для всякого $x \geq 2$ осуществляется неравенства

$$a\frac{x}{\mathrm{Ln}x} < \pi\ x\ < b\frac{x}{\mathrm{Ln}x}. \qquad (1.4)$$

Доказательство.

Приведем доказательство оценки сверху только для n. Покажем, что при всех $n \geq 2$ производится оценка сверху $\pi\ n\ < 1.7\frac{n}{\mathrm{Ln}\ n}$. Воспользуемся методом индукции по n. Несложно проверить это неравенство при $n \leq 1200$. Например, $\pi\ 1200\ = 196$.

$$1.7\frac{1200}{\mathrm{Ln}1200} = 287.7\ldots$$

Допустим, что указанное неравенство выполняется для всех $k \le n$. Проанализируем биномиальный коэффициент $\binom{2n}{n}$. Поскольку $\binom{2n}{n} = \frac{2n\ \ 2n-1\ \ \ldots (n+1)}{n!}$, то при $n < p < 2n$ простое число p разделит $\binom{2n}{n}$. Отсюда произведение $\prod_{n<p\le 2n} p$ делит $\binom{2n}{n}$. Получим неравенства

$$n^{\pi\ 2n\ -\pi(n)} \le \prod_{n<p\le 2n} p \le \binom{2n}{n} < 2^{2n}.$$

После логарифмирования принимаем

$$\pi\ 2n\ -\pi\ n\ \le \frac{2nLn2}{Ln\ n} < 1.39\frac{n}{Ln\ n}.$$

По предположению индукции $\pi\ n\ < 1.7\frac{n}{Ln\ n}$. Значит,

$$\pi\ 2n\ \le 3{,}09\frac{n}{Ln\ n} < 1.7\frac{2n}{Ln(2n)}$$

при $n > 1200$. Действительно, предоставленное неравенство эквивалентно неравенству $3.09Ln\ 2n\ < 3.4Ln(n)$, или $Ln\ n > \frac{3.09Ln2}{Ln\ n}$. Конечное неравенство выполняется при $n > 1200$.

Аналогично

$$\pi\ 2n+1\ \le \pi\ 2n\ +1 \le 3.09\frac{n}{Ln\ n} + 1 < 1.7\frac{2n+1}{Ln\ 2n+1}.$$

Полученный неравенство эквивалентно неравенству

$$3.09n + Ln\ n\ \ Ln\ 2n+1\ < (3.4n+1.7)Ln(n),$$

верному при $n = 1200$. Можно установить, что функция в правой части растет скорее, чем в левой. Следовательно, неравенство выполняется и при $n > 1200$.

Для всех $n \ge 2$ получена оценка сверху.

Дальше мы применим кое-какие признаки простоты для испытания простоты чисел особого вида.

Числами Ферма называются числа $F_n = 2^{2^n} + 1$. Для $n \in \{0, \ldots, 4\}$ числа F_n являются простыми, Ферма предположил, что числа F_n простые для всех n. Эйлер обнаружил, что число F_5 делится на 641. Известны составные числа Ферма и неизвестны простые числа Ферма при $n > 4$.

Определим главные качества чисел Ферма.

Утверждение.

1. Всякий натуральный делитель числа $\mathrm{F_n}$, $n > 1$, можно представить в виде $k2^{n+2} + 1, k \geq 0$.

2. Если $k < n$, то $\mathrm{F_k, F_n} = 1$).

Доказательство.

Несложно видеть, что первое утверждение достаточно обосновать только для простых делителей F_n.

Пусть простое число p делит F_n. Тогда $2^{2^n} \equiv 1(mod\, p)$. Отсюда следует, что $ord\ 2 = 2^{n+1}$ в группе Z_p . Следовательно, $2^{n+1} | (p-1)$, или $p \equiv 1(mod\, 2n+1)$. Отсюда следует, что при $n > 1$ $p \equiv 1(mod\, 8)$

$$\frac{2}{\mathrm{p}} = (-1)^{\frac{\mathrm{p}^2-1}{8}} = 1.$$

Значит, по критерию Эйлера $2^{\frac{\mathrm{p}-1}{2}} \equiv 1 \mod \mathrm{p}$.

Отсюда следует, что $\mathrm{ord}\ 2 \mid \frac{\mathrm{p}-1}{2}$, т. е. $2^{n+1} | (p-1)$. Последнее условие обозначает, что $p = k2^{n+2} + 1$.

Легко проверит, что

$$\mathrm{F_k} = 2^{2^n} - 1 = \mathrm{F_n} - 2.$$

$$\mathrm{k=0}^{\mathrm{n-1}}$$

Следовательно, всякий общий делитель чисел $F_k, F_n, k < n$ деляет число 2. F_k, F_n—нечетны, то $(F_k, F_n) = 1$.

Простой критерий простоты чисел Ферма можно получит из критерия Лукаса.

Теорема 1.2 (Пепин).

$F_n, n \geq 1$ - простое число тогда и только тогда, когда

$$3^{\frac{\mathrm{F_n}-1}{2}} \equiv -1 \mod \mathrm{F_n} . \qquad (1.5)$$

Доказательство.

Достаточно проверить условие теоремы Лукаса, так как 2 имеется

единственным простым делителем числа $F_n - 1$, при $q = 2$. Из равенства $\prod_{k=0}^{n-1} F_k = 2^{2^n} - 1$ вытекает, что $F_0 = 3$ делит $F_n - 2$. Следовательно, $3, F_n = 1$ и $F_n \equiv 2 (\mathrm{mod}\, 3)$.

После этого согласно теореме Лукаса из условия (1.5) вытекает простота числа F_n.

Обратно, считаем что число F_n – простое, если использовать критерий Эйлера, квадратичный закон взаимности, то имеем (12):

$$3^{\frac{F_n - 1}{2}} \equiv \left(\frac{3}{F_n}\right) = \left(\frac{F_n}{3}\right) (-1)^{\frac{F_n - 1}{2}} = \left(\frac{2}{3}\right) = -1 (\mathrm{mod}\, F_n).$$

Далее рассмотрим числа вида $M_n = 2^{n-1}, n \geq 1$. Так как

$$2^{kn} - 1 = \left(2^k - 1\right)\left(2^{k(n-1)} + 2^{k(n-2)} + \cdots + 2^k + 1\right),$$

то число M_n может быть простым только при простом n. Числа Мерсенна - простые числа M_n. В настоящее время известно около 40 чисел Мерсенна, большинство из которых имеет 6 320 430 десятичных цифр.

Утверждение.

Если $n > 2$ – простое число, то любой делитель числа M_n имеет вид $2kn + 1,\ k \geq 0$.

Доказательство.

Достаточно доказать только для простых делителей числа M_n. Пуст простое число p делит M_n. Тогда по малой теореме Ферма $p | (2^{p-1} - 1)$, значит, p делит $(2^{p-1} - 1, 2^n - 1) = 2^{(p-1,n)} - 1$ [5].

1.2 Критерии простоты

Все употребляемые сегодня криптосистемы основываются на алгоритмах разложения больших чисел на простые множители. В асимметричных криптосистемах используются простые числа. Для стойкости к вскрытию в криптосистемах используются простые числа большой длины.

Для проверки на простоту натуральных чисел разработаны различные критерии простоты.

Теорема 1.3. (критерий Вильсона).

Натуральное число $N > 1$ считается простым лишь тогда, когда $N|(\ N-1\ !+1)$.

Доказательство.

Для числа N— простого теорема справедлива.

Если же N— составное, то имеется $1 < d < N-1$, $d|N$. Тогда $d|(N-1)!$ и $d|(\ N-1\ !+1)$.

Для формулировки критерия простоты используем сравнения.

Теорема 1.4.

Нечетное число N считается простым тогда и только тогда, когда сравнения $x^2 \equiv 1(mod\, N)$, $x^2 \equiv 0(mod\, N)$ обладают согласно 2 и 1 решение по модулю N.

Доказательство.

Для простых чисел N теорема верно.

Пусть нечетное составное число $N = \prod_{i=1}^{r} p_i^{k_i}$.

Сравнение $x^2 \equiv 1(mod\, N)$ равносильно системе $x^2 \equiv 1\ (mod\, p_i^{k_i}), i \in \{1,\dots,r\}$ при $r \geq 2$. Следовательно, сравнение $x^2 \equiv 1(mod\, N)$ имеет $2^r > 2$ решений по модулю N.

В случае $r = 1$, то $N = p_i^{k_i}, k1 > 1$. сравнение $x^2 \equiv 0(mod\, N)$ содержит не менее 2-ух решений по модулю N: $x_0 = 0, x_1 = p_1^{k-1}$.

Далее приводится критерии, которые следует из общих теорем, в которых рассматривается представление целых чисел квадратичными формами.

Теорема 1.5.

Нечетное число N - простое тогда и только тогда, когда оно единственным образом представляется в виде разности квадратов целых неотрицательных чисел.

Доказательство.

Ясно, что всякому разложению вида

$$N = ab, a \geq b > 0$$

соответствует представление числа N в виде разности квадратов

$$N = (\frac{a+b}{2})^2 - (\frac{a-b}{2})^2. \qquad (1.6)$$

Теперь можно увидеть, что условие простоты числа N равнозначно условию однозначности его представления в виде: $N = N - 1$.

Поиск делителей числа или доказательство его простоты иногда упрощается, если имеют специальный вид все его делители. Например, как в теореме Ш. Эрмита.

Теорема 1.6.

Если натуральное число N представляется в виде

$$\mathrm{N} = \mathrm{x}^2 + \mathrm{ky}^2 \qquad (1.7)$$

где $(x, y) = 1, k \in \{1,2,3\}$, то любой делитель d числа N имеет вид

$$\mathrm{d} = \mathrm{u}^2 + \mathrm{vx}^2.$$

Доказательство.

С помощью непосредственно проверяемого равенства

$$(\mathrm{a}^2 + \mathrm{kb}^2)\ (\mathrm{c}^2 + \mathrm{kh}^2\ = (\mathrm{ac} + \mathrm{kbh})^2 + \mathrm{k}(\mathrm{ah} - \mathrm{bc})^2$$

устанавливается, что произведение чисел вида $\mathrm{N} = \mathrm{x}^2 + \mathrm{ky}^2$- число того же вида. Следовательно, нужно доказать теорему для простых делителей числа N.

Предположим, что d— простой делитель числа N. Поэтому $y, d\ = 1$ следует из условия $x, y\ = 1$. Следовательно, для некоторого z выполняется сравнение $yz \equiv 1(mod\ d)$. Следовательно, условия $\mathrm{d}|(\mathrm{x}^2+\mathrm{ky}^2)$, $\mathrm{x}^2+\mathrm{ky}^2 \equiv 0(mod\ d)$, $(\mathrm{xz})^2+\mathrm{k} \equiv 0(mod\ d)$ равносильны, и нужно будет доказать, что всякий простой делитель d числа $t^2 + k$ обозначается как $u^2 + kv^2$.

Число $\frac{\mathrm{t}}{\mathrm{d}}$ представим в виде конечную цепную дробь над Z. Для подходящих дробей $\frac{\mathrm{P_n}}{\mathrm{Q_n}}$ числа $\frac{\mathrm{t}}{\mathrm{d}}$ исполняется неравенство

$$\frac{\mathrm{t}}{\mathrm{d}} - \frac{\mathrm{P_n}}{\mathrm{Q_n}} < \frac{1}{\mathrm{Q_n Q_{n+1}}}$$

или

$$(\frac{t}{d}-\frac{P_n}{Q_n})^2<\frac{1}{Q_n^2Q_{n+1}^2}. \qquad (1.8)$$

Если знаменатели подходящих дробей изменяются от 1 до d, то появится такое n, что $Q_n^2<d<Q_{n+1}^2$. Из этой формуле получаем

$$0<(tQ_n-P_nd)^2<\frac{d^2}{Q_{n+1}^2}<\frac{dQ_{n+1}^2}{Q_{n+1}^2}=d$$

и

$$(tQ_n-P_nd)^2+kQ_n^2<d+kQ_n^2<d(k+1).$$

Левая часть неравенства равна

$$t^2+k\ Q_n^2-2tdP_nQ_n+d^2P_n^2$$

и кратна d, получаем равенство

$$(tQ_n-P_nd)^2+kQ_n^2=dm, m\le k. \qquad (1.9)$$

Из формулы (10) следует, что если $m=1$ число d имеет нужный вид. Из (10) вытекает, $(tQ_n-P_nd)^2$ если $m=k>1$. Так как по условию теоремы $k>1$ считается простым числом, то k делит tQ_n-P_nd. Тогда из (1.9) находим $d=k(\frac{tQ_n-P_nd}{k})^2+Q_n^2$. Следовательно, число d имеет требуемый вид.

Равенство (1.9) противоречиво для всех нечетных d, если $k=3, m=2$, так как если оно выполняется, тогда делится на 4 его левая часть.

Потребуется значительные вычисление если использовать сформулированные критерии. На практике ими пользуются только в тех случаях, когда известны дополнительные свойство числе N или его делителях.

Приведем три критерия простоты, используемые для построение эффективных тестов проверки простоты целых чисел.

В критерии Лукаса использовано каноническое разложение числа $N-1$.

Теорема 1.7. (критерий Лукаса)

Натуральное число N считается простым тогда и только тогда, когда имеется такое $a, (a,N)=1$, что:

1) $a^{N-1} \equiv 1 (\mathrm{mod}\, N)$;

2) Для всякого простого делителя q числа $N-1$ имеет место $a^{\frac{N-1}{q}} \not\equiv 1 (\mathrm{mod}\, N)$.

Доказательство.

Если число N считается простым, то Z_n—конечное поле, и в качестве искомого a можно выбрать всякой простой элемент данного поля.

Пусть теперь для некоторого a выполняется следующие условии 1), 2) теоремы. Это обозначает, что в группе Z_n $ord(a) = N-1$. Следовательно, по теореме Лагранжа будет равно $N-1 \mid \varphi(N)$. Если $\varphi\ N \leq N-1$, то $\varphi\ N = N-1$ означает простоту числа N.

Следствие.

Натуральное число N - простое тогда и только тогда, когда для всякого простого делителя q числа $N-1$ имеется такое a_q, что $a_q, N = 1, a_q^{N-1} \equiv 1 (\mathrm{mod}\, N)$ и $a_q^{\frac{N-1}{q}} \neq 1 (\mathrm{mod}\, N)$.

Доказательство.

Из теоремы следует необходимость условия для простоты числа N. Докажем достаточность. Считаем $N-1 = \prod_{j=1}^{r} q_j^{l_i}$ -каноническое разложение $N-1$. Следует, что для всякого $i \in 1, \ldots, r$ имеется такой $a_j \in Z_n$, что $\mathrm{ord}\ a_j \mid N-1, \mathrm{ord}\ a_j \mid \frac{N-1}{q_i}$. Следовательно, $q_j^{l_i} \mid \mathrm{ord}(a_j)$ и $\mathrm{ord}\ a_1, \ldots, \mathrm{ord}\ a_r = \prod_{j=1}^{r} q_j^{l_i} = N-1$.

В абелевой группе Z_n существует элемента a.

В следующем критерии используется свойства квадратичных вычетов для отбраковки составных чисел.

Теорема 1.8. (критерий Эйлера).

Натуральное нечетное число $N > 1$ считается простым тогда и только тогда, когда для всякого $a, (a, N) = 1$ реализовывается соотнесение

$$a^{\frac{N-1}{2}} \equiv \frac{a}{N} \mod N . \qquad (1.10)$$

Доказательство.

Если N является простым, то сравнение (1.10) выполняется. Пусть теперь условие (1.10) имеет место, но $N = \prod_{i=1}^{r} p_i^{k_i}$- составное число. Тогда для всякого $a \in Z_n$ $a^{N-1} \equiv \left(\frac{a}{N}\right)^2 \equiv 1 (\mod N)$, т.е. $ord\ a\ |N-1$. С другой стороны, по теореме Лагранжа $ord\ a\ |\varphi(N)$.

Если $k_i > 1$ для некоторого $i \in \{1, \dots, r\}$, то $p_i|\varphi(N)$, в группе Z_n существует элемент b порядка p_i. Получили противоречие: $p_i|N-1$ и $p_i|N$.

Осталось рассмотреть случай $N = \prod_{i=1}^{r} p_i, r \geq 2$. Выберем элемент b, являющийся квадратичным невычетом по модулю p_1. Найдется число a (по китайской теореме об остатках), удовлетворяющее системе сравений

$$a \equiv b \ mod\, p_1 \ ;$$
$$a \equiv 1 \ mod\, p_2 \ ;$$
$$\dots\dots\dots\dots\dots$$
$$a \equiv 1 \ mod\, p_r \ .$$

Тогда

$$\frac{a}{N} = \prod_{i=1}^{r} \frac{a}{p_i} = \left(\frac{b}{p_1}\right) = -1.$$

Значит, по условию $a^{\frac{N-1}{2}} = -1 (mod\, N)$. Итак с одной стороны, имеем $a^{\frac{N-1}{2}} = -1 (mod\, p_2)$, а с другой – по выбору a имеем $a^{\frac{N-1}{2}} = 1 (mod\, p_2)$. Отсюда следует, что $2 \equiv 0 mod\, p_2$, т. е. $p_2 = 2$ и N- четное число. Получили противоречие.

Теорема 1.9. (критерий Миллера).

Для нечетного натурального числа N, $N-1 = 2^t u,\ u, 2 = 1$ равносильны предложение:

1) число N считается простым;

2) для всякого a подобного, что $(a, N) = 1\ a^u \neq 1 (mod\, N)$, найдется $k \in \{0, 1, \dots, t-1\}$ со свойством $a^{2^k u} \equiv -1 (\mod N)$.

Глава 2. Тесты простоты целых чисел

2.1 Вероятностные тесты простоты

В настоящие время известно достаточно большое количество алгоритмов проверки чисел на простоту. Не обращая внимания на то, что основная масса из таких алгоритмов имеет субэкспоненциальную оценку сложности, на практике они демонстрируют абсолютно приемлемую скорость работы.

Каноническое разложение используют для обследования простоты числа N. Для больших N потребует значительных вычислений, так как задача факторизации целых чисел - вычислительно сложной. В силу этого для проверки простоты чисел используются алгоритмы, именуемые тестами простоты. Под тестом простоты разумеется «детерминированный или вероятностный алгоритм, позволяющий для любого целого $N > 1$, не находя его канонического разложения, определять, является ли число N простым или составным».

Тест простоты используют некоторый критерии простоты числа, в котором формулируется условия простоты. Часто проверка всех условии – трудоёмко. Поэтому на практике порой ограничиваются обследованием только части условий. В результате либо нашлось не выполняющееся условие (то есть — «число N составное»), либо все условия выполнены (тогда говорят о простоте числа N с некоторой вероятностью). Вероятностными тесты простоты считают алгоритмы, проверяющие част условии простоты. Вероятностные тесты простоты, как правило достаточно просты в обосновании и реализации, их временная сложность проявляется полиномом от $log\ N$.

Современные тесты простоты могут результативно обследовать простоту чисел, обладающих в своей десятичной записи несколько сотен цифр. Большинство из современных тестов простоты в процессе своей работы приобретают свойство: дополнительные данные, которые позволяют

быстро определить простоту N.

2.1.1 Тест простоты Ферма

Вероятностные тесты, позволяют определить простоту числа с некоторой достаточно низкой вероятностью ошибки. Среди них можно выделить следующие, наиболее эффективные тесты простоты:

Тест Ферма – базируется на малой теореме Ферма и весьма эффективен в обнаружении составных чисел.

По теорема Ферма для, N — простое,

$$a^{N-1} \equiv 1 \ \mathrm{mod} N \qquad (2.1)$$

если для всех a, $(a, N) = 1$ обладает пункт сравнение

Приведённая условия – это необходимая условия числа N

Определение.

Число N - псевдопростое по основанию a, если выполняется

$$a^{N-1} \equiv 1 \ \mathrm{mod} N$$

для чисел a и N.

Ясно, что N считается псевдопростым по основанию a тогда и только тогда, когда $(a, N) = 1$ и порядок элемента a в группе Z_N делит число $N - 1$. Отметим, что псевдопростое по основанию a число не обязательно считается простым (341 псевдопростое по основанию 2, хотя $341 = 11 * 31$).

Описывает следующие утверждение свойства псевдопростых чисел.

Утверждение.

Для N нечетного имеет место.

a) множество всех $a \in Z_N$, относительно которых N считается псевдопростым, создаст подгруппу в Z_N ;

b) если N не считается псевдопростым хотя бы по одному причине a, то N не является псевдопростым относительно по крайней мере большинство чисел из Z_N.

Доказательство.

a) Из критерии быть подгруппой в группе вытекает a).

b) Следует из пункта а) и теоремы Лагранжа о порядке подгруппы конечной группы. Действительно, если

$$H_N = \{a \in Z_N \mid a^{N-1} \equiv 1 \bmod N\},$$

то по а) имеет место $H_N < Z_N$. Из б) следует $H_N \neq Z_N$. По теореме Лагранжа $|H_N|$ делит $|Z_N|$. Отсюда, получаем $\frac{|H_N|}{|Z_N|} \leq \frac{1}{2}$.

Вероятностный тест простоты:

1) Случайно выбрать число $a \in \{1, \ldots, N-1\}$ и вычислить $(a, N) = d$. Если $d > 1$, то N — составное.
2) Если $d = 1$, то проверить выполнимость сравнения

$$a^{N-1} \equiv 1 \bmod N.$$

Если не выполняется, то N — составное. Иначе – неизвестно, то есть или простое или составное.

В алгоритме включим понятие вероятности. Пусть N- составное число. Вероятность успеха - вероятность события, заключающегося в том, что алгоритм выдаст N— составное. Ясно, что $P_0 = 1 - \frac{|H_N|}{N-1}$.

В результате выполнении теста возможно ситуации:

1) N— простое число, тест дает «неизвестно»;

2) N— число составное, и N не считается псевдопростым хотя бы по одному основанию a, тест дает - «N— составное» с вероятностью успеха P_0. Из б) вытекает, что

$$P_0 = 1 - \frac{H_N}{N-1} \geq 1 - \frac{H_N}{Z_n} \geq \frac{1}{2};$$

3) N— составное число и N является псевдопростым по всем основанием $a \in Z_n$, тогда

$$P_0 = 1 - \frac{Z_N}{N-1} = 1 - \frac{\varphi(N)}{N-1}.$$

Если $N = \prod_{i=1}^{s} p_i^{k_i}$- каноническое разложение числа N, то

$$P_0 = 1 - \frac{1}{N-1} \prod_{j=1}^{s} p_j^{k_j - 1} \left(p_j - 1 \right).$$

Отметим, если N — составное и не является псевдопростым хотя бы по одному основанию a, то при использовании алгоритма для $s > 1$ различных значений a вероятность успеха $P_0^{(s)}$ оценивается таким образом:

$$P_0^{(s)} = 1 - \left(1 - P_0\right)^s \geq 1 - \frac{1}{2^s}.$$

Последнее неравенство обозначает, что с увеличением s вероятность обосновать непростоту числа N с помощью алгоритма стремится к единице.

Отметим, третья ситуация возможна: существуют составные числа, являющиеся псевдопростыми по всем основаниям $a \in Z_n$.

Определение.

Составные числа N, для сравнение которых (1) осуществляется для всех $a \in Z_n$, именуются числами Кармайкла.

Числа Кармайкла встречаются очень редко. Встречается 2163 чисел Кармайкла, не превосходящих $25 * 10\text{^}9$. Из чисел меньших 10000 числами Кармайкла являются следующие 16 чисел: 561, 1105, 1729, 2465, 2821, 6601, 8911, 10 585, 15 841, 29 341, 41 041, 46 657, 52 633, 62 745,973, 75 361. Недавно было обосновано, что чисел Кармайкла бесконечно множество. Для проверки принадлежности числа N к числам Кармайкла требуется нахождение разложения числа на простые сомножители (факторизации числа N). Предварительная отбраковка чисел Кармайкла не является возможной, так как задача факторизации чисел - более сложная, чем задача проверки простоты.

Опираясь на теорема Лукаса сформулируем детерминированный тест простоты.

Пусть $N - 1 = \prod_{i=1}^{r} q_i^{l_i}$.

Последовательно перебираются все числа $a \in \{1, \ldots, N-1\}$ и для всех a реализовываются следующие действия:

1. Вычисляется $(a,N)=d$. Если $d>1$, то N- составное.

2. Для $d=1$, проверяется выполнимость сравнения(1). Если (1) не выполнено, то N- составное.

3. Иначе для всех $q_i, i \in \{1,\dots,r\}$ проверяется условие $a^{\frac{N-1}{q_i}} \not\equiv 1 (\mathrm{mod}\ N)$. Если это условие выполнено для некоторых q_i, то простое, иначе следующее значение a выбирается.

4. Если не найдётся числа a, удовлетворявшего условиям критерия Лукаса, то N- составное.

Недостатки детерминированного теста простоты: он экспоненциальное по сложности, требует знать каноническое разложение числа $N-1$.

2.1.2 Тест простоты Соловея и Штрассена

Между различии символами Якоби и Лежандра основан тест Соловея и Штрассена. Тест может всегда определят корректно, что число является простым, но для составных чисел с некоторой вероятностью он имеет возможность предоставить неправильный ответ.

Критерии Эйлера.

Натуральное нечетное число N>1 является простым тогда и только тогда, когда для любого $a, (a,N)=1$ выполняется сравнение

$$a^{\frac{N-1}{2}} = (\frac{a}{N})(mod\ N).$$

Используя этот критерий простоты Эйлира, Р. Соловей и В. Штрассен сформулировали вероятностный тест проверки простоты чисел.

1. Случайно выбрать число $a \in \{1,\dots,N-1\}$ и вычислить $(a,N) = d$. Если $d>1$, то N— составное.
2. При $d=1$, проверяется

$$a^{\frac{N-1}{2}} \equiv \frac{a}{N} \mod N . \qquad (2.2)$$

Если предоставленное сравнение не выполняется, то N— составное, иначе неизвестно.

Трудоемкость проведя данного теста оценивается величиной $O(Log^3 N)$. Так как вычисление $a^{\frac{N-1}{2}} \bmod N$ требует $O(log_2 N)$ умножений в кольце Z_n,, то следует оценка $O(\log_3 N)$ для сложности вычисления $a^{\frac{N-1}{2}} \bmod N$. Для вычисление (a, N) и $\frac{a}{N}$ необходимо $O(\log_2 N)$ операций.

Приведенный тест во многом подобен тесту, в основе которого малая теорема Ферма, но при его использовании, возникают две ситуации:

1. число N — простое и тест постоянно выдает ответ «неизвестно»;
2. число N— составное, тест дает «N— составное» с вероятностью успеха не менее 1/2.

Докажем оценку вероятности успеха теста Соловея и Штрассена.

Определение.

Число N называется Эйлеровым псевдопростым по основанию a, если для чисел a, N выполняется

$$a^{\frac{N-1}{2}} \equiv \frac{a}{N} \bmod N .$$

Отметим, что не обязательно является простым, Эйлерово псевдопростое число N по основанию a. Возведя в квадрат сравнение $a^{\frac{N-1}{2}} \equiv \frac{a}{N} \bmod N$, получим сравнение $a^{N-1} \equiv 1 \bmod N$. Значит, если N — Эйлерово псевдопростое по основанию a, то N- псевдопростое по основанию a. Аналога чисел Кармайкла, не существует.

Подтверждение оценки вероятности успеха вытекает из утверждения.

Утверждение.

Для нечетное число N, имеет место:

a) множество всех $a \in Z_n$, относительно которых N является Эйлеровым псевдопростым, образуют подгруппу в Z_n;

b) если N— составное число, то N не является эйлеровым псевдопростым относительно, по крайней мере, половины чисел из Z_n.

Доказательство.

а) доказывается с помощью критерия быть подгруппой в конечной группе и свойств символа Якоби.

б) следует из а) и теоремы Лагранжа о порядке подгруппы конечной группы. Действительно, если

$$K_N = \{a \in Z_N : a^{\frac{N-1}{2}} \equiv \left(\frac{a}{N}\right) \bmod N\},$$

то по а) $K_N < Z_N$. Из б) и критерии Эйлера следует $K_N \neq Z_N$, а из теоремы Лагранжа следует,что $|K_N|$ делит $|Z_N|$. Тогда имеем $\frac{|K_N|}{|Z_N|} \leq \frac{1}{2}$.

Для составного числа N из б) утверждения получим оценку вероятности успеха в тесте Соловея–Штрассена:

$$P_0 = 1 - \frac{K_N}{N-1} \geq 1 - \frac{K_N}{Z_N} \geq \frac{1}{2}.$$

Вероятность успешного выполнение теста не менее 1/2. Если N— составное, то применяя алгоритм для $s > 1$ разных значений a вероятность успеха $P_0^{(s)}$ оценивается так:

$$P_0^{(s)} = 1-(1-P_0)^s \geq 1 - \frac{1}{2^s}.$$

Существует детерминированный вариант теста Соловея–Штрассена. Перебираются последовательно числа $a \in \{1,\ldots,N-1\}$ и для любого a:

1. Вычисляется $(a,N) = d$. Если $d > 1$, то N — составное.

2. Для $d = 1$, проверяют (2). Если (2) не выполнено, то N — составное, иначе выбирается следующее a.

3. Если (2) выполняется для всех $a \in \{1,\ldots,N-1\}$, то N — простое.

Детерминированный тест простоты экспоненциален по сложности, но при дополнительных условиях можно снизить его трудоемкость.

Основное превосходство теста является то, что он, в отличие от теста Ферма, определяет числа Кармайкла как составное. Не смотря на это, использование теста Миллера и Рабина является достаточно достоверно, чем теста Соловея и Штрассена.

2.1.3 Тест простоты Соловея и Штрассена

Тест Миллера-Рабина также, как и провидение выше тесты, является вероятностным тестом, однако реализовывается на ЭВМ более эффективно чем тест Соловея-Штрассена. Вероятность ошибки у теста Миллера-Рабина гораздо ниже чем у провиденных двух тестов. Обычно для нахождение простого числа в данном тесте достаточно одной итерации.

N— нечетное число, $N-1=2^t u, (u,2)=1$.

Тест Миллера–Рабина – один из лучших вероятностных тестов проверки простоты чисел.

1. Случайно выбрать число $a \in \{1,\dots,N-1\}$ и вычислить $(a,N)=d$. Если $d>1$, то N— составное.
2. При $d=1$, то вычислить

$$r_k \equiv a^{2^k u} (\mathrm{mod}\ N) \text{ для } k \in \{0,..,t-1\}.$$

Если $r_0 \equiv 1(mod\ N)$ или $r^k \equiv -1(mod\ N)$ при некотором $k \in \{0,\dots,t-1\}$, то неизвестно, иначе N— составное.

Если в алгоритме неизвестно, то можно повторить тест для следующего числа a. На практике обычно применяют алгоритм для концентрированного числа $s>1$ разных значений a.

Определение.

Число N псевдопростое по основанию a называется сильно псевдопростым по основанию a, если выполняется одно из условий:

1. либо $a^u \equiv 1(mod\ N)$;
2. либо найдется $k \in \{0,\dots,t-1\}$ такое, что $a^{2^k u} \equiv -1(\mathrm{mod}\ N)$.

Из критерии Миллера следует, что:

1. число N — простое, тест Миллера–Рабина дает - неизвестно;

или

2. число N — составное, тест дает N—составное с вероятностью успеха, большей нуля.

$O(log^3 N)$ - трудоемкость теста Миллера–Рабина для одного числа a.

Для составного N вероятность правильного ответа в тесте не менее ¾, как показал М. Рабин. Результат будет доказан ниже.

Пусть A_N— множество всех $a \in \mathbb{Z}_N$, относительно которых N является сильно псевдопростым, тогда вероятность P_0успеха в тесте Миллера–Рабина может быть оценена следующим образом:

$$P_0 = 1 - \frac{A_N}{N-1} \geq 1 - \frac{A_N}{\mathbb{Z}_N} = 1 - \frac{A_N}{\varphi\ N}.$$

Поэтому ниже будет вычисляться величина $\frac{A_N}{\varphi\ N}$. Сначала введем необходимые обозначения. Пусть N — нечетное число,

1) $N = \prod_{i=1}^{s} p_i^{k_i}$-каноническое разложение числа N;
2) $N - 1 = 2^t u,\ u, 2\ = 1, t \geq 1;$
3) $p_i - 1 = 2^{v_i} u_j,\ u_j, 2\ = 1, v_j \geq 1, i \in 1, \ldots, s\ ;$
4) $t_j = \ i \in 1, \ldots, s\ v_i = j\ , t_j \geq 0, j \geq 1;$
5) $m = \min_{i \in 1,\ldots,s} v_i, M = \max_{i \in 1,\ldots,s} v_i.$

Нетрудно заметить, что

1) $t_1 = \cdots = t_{m-1} = 0, t_m \neq 0, t_M \neq 0, t_{M+1} = t_{M+2} = \cdots = 0;$
2) $\sum_{i=m}^{M} t_i = s;$
3) $\sum_{i=m}^{M} i t_i = r, то\ \varphi\ N\ = 2^r\ \prod_{j=1}^{s} u_j p_j^{k_i - 1}.$

Также легко заметить, что $m \leq t$. Действительно, из очевидного выражения $N - 1 = p_s^{k_s} - 1 + \prod_{i=1}^{s-1} p_i^{k_i} - 1\ p_{i+1}^{k_{i+1}} \ldots p_s^{k_s}$ и разложения $p^k - 1 = (p-1)(p^{k-1} + \cdots + p + 1)$ вытекает, что $2^m | (N-1)$.

Теорема 2.1.

Если $N > 9$ –нечетное составное число, то $\frac{|A_N|}{\varphi(N)} \leq \frac{1}{4}$,

$$A_N = (1 + \frac{2^{sm}-1}{2^s-1}) \prod_{i=1}^{s} (u, u_i).$$

Доказательство.

Пусть сначала $s = 1, N = p^k, k \geq 2$. В этом случае $m = v_j$ и

$$\frac{|A_N|}{\varphi(N)} = \frac{2^m(u,u_j)}{2^m u_j p^{k-1}} = \frac{(u,u_j)}{u_j p^{k-1}}$$

Так как $p-1 \quad N-1$,то u_j u и $\frac{|A_N|}{\varphi(N)} = \frac{1}{p^{k-1}} \le \frac{1}{4}$ при $N > 9$.

1) Пусть теперь $s \ge 2$.По доказанной теореме

$$\frac{|A_N|}{\varphi(N)} = 1 + \frac{2^{sm}-1}{2^s-1} \frac{\sum_{j=1}^{s} u,u_j}{2^r \sum_{j=1}^{s} u_j p_j^{k_j-1}} = C \frac{\sum_{j=1}^{s}(u,u_j)}{\sum_{j=1}^{s} u_j p_j^{k_j-1}}, \quad (2.3)$$

где

$$C = \frac{2^{sm}+2^s-2}{2^r(2^s-1)}.$$

Из (2.3) следует, что $\frac{|A_N|}{\varphi(N)} \le C$. Рассмотрим два возможных сучая.

А) Пусть сначала $m = M$. Тогда $r = ms$ и

$$C = \frac{1}{2^s-1} + \frac{1}{2^{sm}} - \frac{1}{2^{sm}\ 2^s-1}. \quad (2.4)$$

Если $s > 3$, то из 2.4 следует, что $C < \frac{1}{15} + \frac{1}{16} < \frac{1}{4}$.

Если $s = 3,$ и $m > 1$, то $C < \frac{1}{7} + \frac{1}{64} < \frac{1}{4}$.

Если $s = 3,$ и $m = 1$, то $C = \frac{1}{7} + \frac{1}{8} - \frac{1}{56} < \frac{1}{4}$.

Если $s = 2,$ и существует$k \ge 2$, то $C < \frac{1}{3} + \frac{1}{4} = \frac{7}{12}$. Кроме того, в этом случае

$$\frac{\sum_{j=1}^{s}(u,u_j)}{\sum_{j=1}^{s} u_j p_j^{k_j-1}} \le \frac{1}{p_j}.$$

Значит, $\frac{|A_N|}{\varphi(N)} \le \frac{7}{12} * \frac{1}{p_j} \le \frac{7}{12} * \frac{1}{3} < \frac{1}{4}$.

Если $s = 2, k_1 = k_2 = 1$ имеется i, для которого $(u,u_i) \ne u_i$,то в силу нечетности u,u_i верно неравенство $3(u,u_i) \le u_i$. В этом случае получаем аналогичную оценку:

$$\frac{|A_N|}{\varphi\ N} \le \frac{7}{12} * \frac{u,u_i}{u_i} \le \frac{7}{12} * \frac{1}{3} < 1/4.$$

Пусть теперь $s = 2, k_1 = k_2 = 1$ и $u, u_i = u_i, i \in \{1,2\}$. В этом случае из условий $u_i | u, i \in \{1,2\}$ и $m \leq t$ получаем, что $p_i - 1 \mid (N-1)$.

Б) Пусть теперь $m < M$. Тогда

$$r = \sum_{i=m}^{M} i\, t_i = ms + \sum_{i=m+1}^{M} (i-m)t_i \geq ms + M - m .$$

Значит, для величины C в (4) верна оценка

$$C \leq \frac{1}{2^{M-m}} \cdot \frac{1}{2^s - 1} + \frac{1}{2^{sm}} - \frac{1}{2^{sm} \cdot 2^s - 1} . \qquad (2.5)$$

Если $M - m \geq 2$, то $C < \frac{1}{4} * \frac{1}{3} + \frac{1}{4} = \frac{7}{48} < \frac{1}{4}$.

Если $M - m = 1, s \geq 3$ то $C < \frac{1}{2} * \frac{1}{7} + \frac{1}{8} = \frac{15}{112} < \frac{1}{4}$.

Если $M - m = 1, s = 2, m \geq 2$ то $C < \frac{1}{2} * \frac{1}{3} + \frac{1}{16} = \frac{19}{96} < \frac{1}{4}$.

Пусть наконец, $M - m = 1, s = 2$б ь $= 1$. В этом случае

$p_1 - 1 = 2u_i, p_2 - 1 = 2^2 u_2, N - 1 = 2u$ и $C = \frac{2^{sm} + 2^s - 2}{2^r \cdot 2^s - 1} = \frac{6}{24} = \frac{1}{4}$.

С помощью ЭВМ и различных тестов простоты получены интересные результаты, позволяющие доказывать простоту небольших простых чисел.

Приведем нескольких примеров:

если $N < 1\,373\,653$ и N сильно псевдопростое относительно всех $a \in \{2,3\}$, то N — простое;

если $N < 25\,326\,001$ и N сильно псевдопростое относительно всех $a \in \{2,3,5\}$, то N — простое;

если $N < 3\,474\,749\,660\,383$ и N сильно псевдопростое относительно всех $a \in \{2,3,5,7,11,13\}$, то N — простое;

если $N < 341\,550\,071\,728\,321$ и N сильно псевдопростое относительно всех $a \in \{2,3,5,7,11,13,17\}$, то N — простое.

Если N— составное, то при применении алгоритма 2.3 для $k > 1$ разных значений a вероятность успеха $p_0^{(k)}$ оценивается следующим образом:

$$p_0^{(k)} = 1 - \left(1 - P_0\right)^k \geq 1 - \frac{1}{4^k}.$$

Существует, детерминированный вариант теста Миллера–Рабина, основанный на критерии Миллера и имеющий экспоненциальную сложность.

Итак, вероятностные тесты простоты могут довольно результативно устанавливать непростоту натуральных чисел. Если тест выдают результат «неизвестно», то число N скорее всего простое. Для определение простоты чисел вероятностные тесты не подходят, поэтому используют достаточно сложные детерминированные тесты простоты.

2.2 Полиномиальный тест распознавания простоты

Детерминированные тесты – дают гарантированно точный ответ простое ли исследуемое число или нет. Не смотря на это не получили широкого практического применения вследствие сложности алгоритмов их реализации.

В работе [6 и 7] приведен полиномиальный детерминированный алгоритм распознавания простоты. Алгоритм основан на следующем критерии простоты.

Теорема 2.2.

Для взаимно простых чисел a и N число N — простое в том и только в том случае, когда имеет место

$$(x-a)^N \equiv x^N - a \mod N . \qquad (2.6)$$

Доказательство.

Если $0 < i < N$, то коэффициент при x^i в выражении $(x-a)^N - x^N - a$ равен $(-1)^i \binom{N}{i} a^{N-1}$. Поэтому, для простого N все эти коэффициенты сравнимы с нулем по модулю N. Для $i = 0$ соответствующий коэффициент равен $(-1)^N a^N + a$. По малой теореме Ферма он сравним с нулем по модулю N.

Пусть N - составное, q— простой делитель числа N, причем $N = q^k u,\ q, u = 1$. Тогда q^k не делит $\binom{N}{q}$, взаимно просто с a^{N-q}, тогда, коэффициент при x^q не сравним с нулем по модулю N.

При проверке равенства (2.6) необходимо вычислить всех N коэффициентов. Поэтому в алгоритме вместо сравнения (6) используется

$$(x-a)^N \equiv \ x^N - a \mod x^r - 1 \mod N \ , \qquad (2.7)$$

где значения a и r перебираются особым образом: находят «подходящее» значение r и для него проверяется сравнение (2.7) для всех «малых» значение a.

Алгоритм

Дано: целое $N > 1$.

1. Если число $N = a^b, b > 1$, то N— составное.
2. $r = 2$.
3. Если $r < N$, то 4–8.
4. $d = (r, N)$. Если $d > 1$, то N— составное, иначе перейти к 5
5. Если r - простое, то 6–7, иначе 8.
6. q— НОД $(r-1)$.
7. Если $q > 4 \ \bar{r} \log_2 N$ и $N^{\frac{r-1}{q}} \not\equiv 1 (mod\ r)$ то перейти к 9 с тем же r.
8. Прибавить к r единицу. Если $r = N$, то N— простое, иначе 3.
9. Если $N - 1 \le [2 \ \bar{r} \log_2 N]$ то для всех $r < a \le N - 1$ проверить выполнение условия $(a, N) = 1$. Если хотя бы для одного такого a значение (a, N) больше 1, то N— составное. В противном случае перейти на 10.

Если

$$N - 1 >\le [2 \ \bar{r} \log_2 N],$$

то для всех

$$1 \le a \le\le [2 \ \bar{r} \log_2 N]$$

проверить выполнение соотношения

$$(x-a)^N \equiv \ x^N - a \mod x^r - 1 \mod N \ .$$

Если хотя бы для одного такого a соотношение

$$(x-a)^N \equiv \ x^N - a \mod x^r - 1 \mod N$$

не выполнено, то N — составное. В противном случае перейти на шаг 10.

10. Выдать ответ: «N — простое».

Замечание. Проверка условия $r = N$ на шаге 8 алгоритма внесена в связи с тем, что при малых значениях N цикл по r (шаг 3) может не найти искомого числа r. Действительно, из неравенств $\frac{r-1}{2} \geq q > 4\ \bar{r}\log_2 N$ получаем $r - 8\ \bar{r}\log_2 N - 1 > 0$. Так как положительный корень уравнения $x^2 - (8\log_2 N)x - 1 = 0$ имеет вид $x = 4\log_2 N + \sqrt{16\log_2 N^2 + 1} > 8\log_2 N$, то $r > 64\log_2 N$. При этом цикл может заканчиваться значением $r = N$ только при простых N, поэтому последующие шаги оказываются ненужными.

Из анализа алгоритма следует, для завершения первого цикла по r (3) выполнить $O((\log_2 N)^6)$ шагов, тогда во втором цикле следует выполнить a (9) надо выполнить $2\ \bar{r}\log_2 N = O((\log_2 N)^4)$ шагов, следовательно алгоритм выполняется за полиномиальное число шагов, каждый из которых полиномиально сложен.

Опишем фундаментальным результатом Е. Фоуври из аналитической теории чисел.

Теорема 2.3.

Найдутся положительные константы a_1, a_2 и натуральное N_0, такие что для всех $N > N_0$ в интервале $[a_1(\log_2 N)^6;\ a_2(\log_2 N)^6]$ существует простое число r, для которого выполняется условия:

1) либо $r|N$;
2) либо $r-1$ имеет такой простой делитель

$$q \geq 4\ \bar{r}\log_2 N,$$

что $N^{\frac{r-1}{q}} \not\equiv 1\ mod\ r$, и $q|ord(N)$ в группе $\mathbb{Z}_r$.

Доказательство.

Найдём оценку числа M простых чисел r в интервале $[a_1(\log_2 N)^6;\ a_2(\log_2 N)^6]$, удовлетворяющих условию

$$P\ r-1\ > (a_2(\log_2 N)^6)^{\frac{2}{3}} > r^{\frac{2}{3}}$$

Согласно теореме Чебышева при некоторых константах $0 < c1 < 1 < c2$ выполняются неравенства

$$c_1 \frac{x}{\log_2 x} < \pi\ x\ < c_2 \frac{x}{\log_2 x}.$$

Поэтому получаем, что при всех N, начиная с N_0,имеет место

$$M \geq \pi_i\ a_2\ \log_2 N^{\ 6}\ - \pi\ a_1\ \log_2 N^{\ 6}\ \geq$$

$$\geq \frac{c_1 a_2\ \log_2 N^{\ 6}}{\log_2\ a_2\ \log_2 N^{\ 6}} - \frac{c_2 a_1\ \log_2 N^{\ 6}}{\log_2\ a_1\ \log_2 N^{\ 6}} \geq$$

$$\geq \frac{c_1 a_2\ \log_2 N^{\ 6}}{7 \log_2 \log_2 N} - \frac{c_2 a_1\ \log_2 N^{\ 6}}{6 \log_2 \log_2 N} \geq$$

$$\geq\ \frac{c_1 a_2}{7} - \frac{c_2 a_1}{6}\ \frac{\log_2 N^{\ 6}}{7 \log_2 \log_2 N} = c_3 \frac{\log_2 N^{\ 6}}{\log_2 \log_2 N},$$

где константы a_1, a_2 выбраны так, что

$$\log_2 a_1 > 0, log a_2 < \log_2 \log_2 N\ \text{и}\ c_3 > 0,$$

всегда можно сделать при довольно больших N.

Пусть $x = a_2(\log_2 N)^6$ проанализируем произведение

$$L = \ N-1\ \ N^2-1\ \ldots\ N^{x^{\frac{1}{3}}}-1\ .$$

В данном произведении $[x\text{^}1/3]$ сомножителей, каждый из которых хранит не более $\log_2\ N^{x^{\frac{1}{3}}} - 1\ \leq [x^{1/3}] \log_2 N$ простых делителей. Следовательно, L имеет не более $x^{\frac{2}{3}} \log_2 N$ простых делителей. С другой стороны,

$$x^{\frac{2}{3}} \log_2 N < c_3 \frac{\log_2 N^{\ 6}}{\log_2 \log_2 N} < M.$$

Следовательно, существует простое число r, не являющееся делителем числа L, для которого существует простой делитель q числа $r-1$ со свойствами:

1) $q = P\ r-1\ > r^{2/3} > 4\ \bar{r}\log_2 N$;

2) $N^{\frac{r-1}{q}} \not\equiv 1\ mod\, r$;

3) $q|ord(N)$ в группе $\mathbb{Z}_r$.

В самом деле, $\frac{r-1}{q} \leq \frac{r-1}{r^{\frac{2}{3}}} < r^{\frac{1}{2}} < x^{\frac{1}{2}}$, и по выбору числа L будет выполнено свойство 2). С другой стороны,

$N^{r-1} \equiv 1(mod\, r)$ и, следовательно, $ord(N)$ не делит $\frac{r-1}{q}$ и

$ord(N)|r-1$. Теорема доказана.

Теорема 2.4.

Алгоритм имеет асимптотическую сложность $O((\log_2 N)^{12} pol(\log_2 \log_2 N))$, где $pol(x)$ — некоторый многочлен.

Теорема 2.5.

Для нечетное число $N > 1$ алгоритм дает результат «N— простое» тогда и только тогда, когда N— простое.

Итак, алгоритм показывает полиномиальность задачи проверки простоты чисел, однако реальная сложность алгоритма высокая.

Заметим, что оценка сложности алгоритма может быть снижена с

$$O(\ \log_2 N\ ^{12} pol(\log_2 \log_2 N)) \text{ до}$$

$$O(\ \log_2 N\ ^{3} pol(\log_2 \log_2 N)),$$

если доказать следующую гипотезу:

Если $r|N$ и $(x-1)^N \equiv (x^N-1)(mod\, x^r-1)(mod\, N)$, то либо N— простое число, либо $N^2 \equiv 1(mod\, r)$.

Глава 3. Алгоритмы тестов простоты целых чисел

3.1 Алгоритм теста Ферма

Реализация теста Ферма была выполнена на основе алгоритма, изображенного в виде блок схемы на рис.3.1.

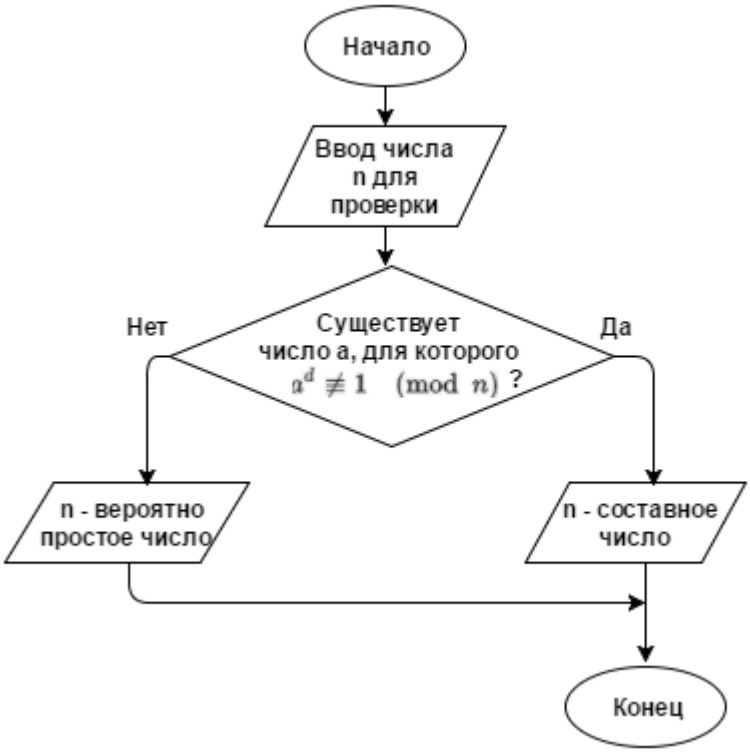

Рисунок 3.1 – Блок схема

Пошаговое выполнение алгоритма теста Ферма:

Алгоритм 3.1.

1) число $a \in \{1, \ldots, N-1\}$ выбирается случайно. Вычислить $(a, N) = d$. Если $d > 1$, то N — составное.

2) При $\mathrm{d} = 1$ проверить

$$\mathrm{a}^{\mathrm{N}-1} \equiv 1 \ \mathrm{mod N} .$$

Если не выполняется, то N — составное. Иначе – неизвестно, то есть или простое или составное.

Если алгоритм выдал ответ «неизвестно», то можно повторять тест для следующего числа a. Алгоритм 3.1 для фиксированного числа $s > 1$ различных значений a применяется на практике.

Вычисление $a^{N-1} mod\, N$ требует $O(\log_2 N)$ умножений в кольце Z_n. Оценка $O(Log^3 N)$ для сложности проверки условия $a^{N-1} 1(mod\, N)$. На вычисление (a, N) необходимо $O(\log_2 N)$ операций. Следовательно, трудоемкость алгоритма 3.1 оценивается величиной $O(Log^3 N)$.

Результат выполнения теста Ферма реализованной на языке C++ изображена на рис. 3.2.

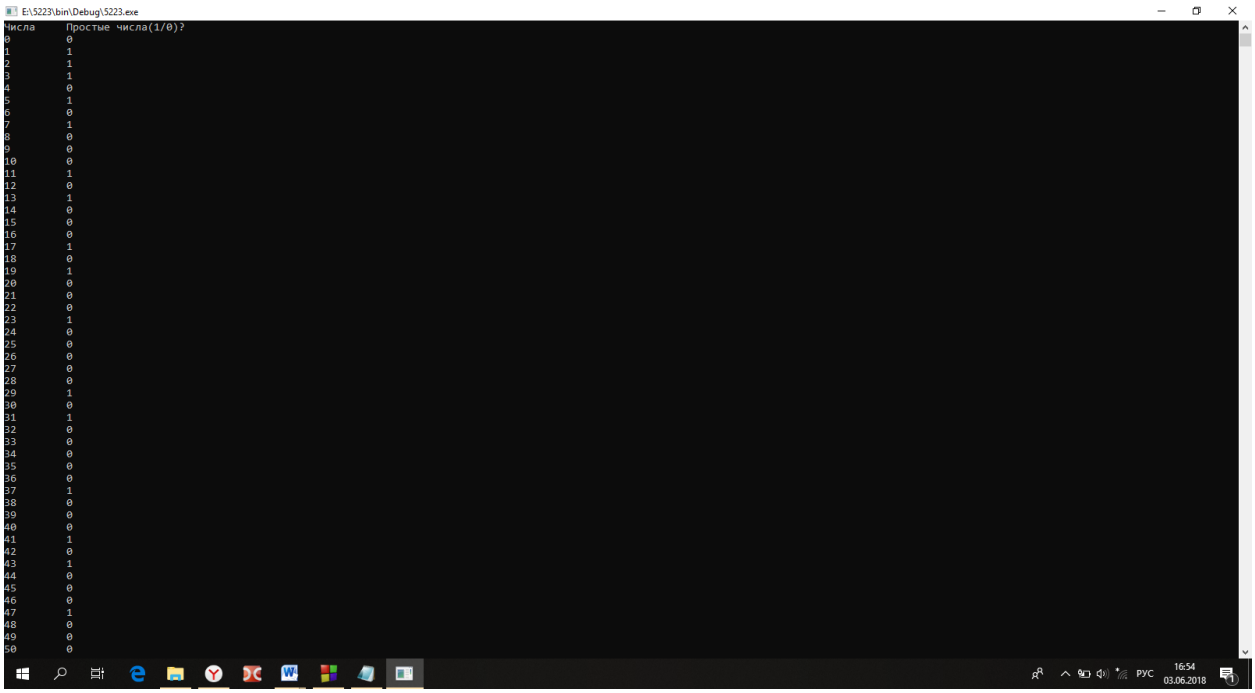

Рисунок - 3.2. Результат выполнения теста Ферма

3.2 Алгоритм теста Соловея и Штрассена

Реализация теста Соловея и Штрассена была выполнена на основе алгоритма, представленного в виде блок схемы на рис.3.3.

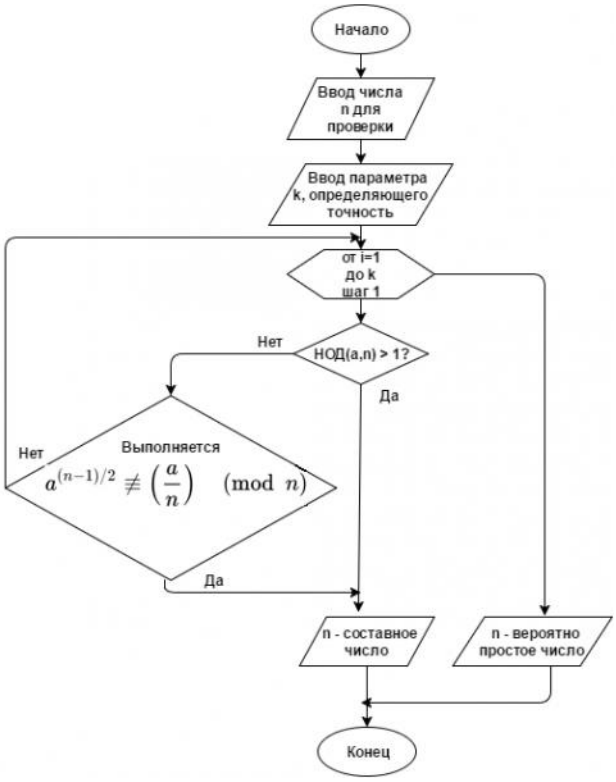

Рисунок 3.3 – Блок схема.

Пошаговое выполнение алгоритма теста Соловея и Штрассена:

Алгоритм 3.2.

3. Случайно выбрать число $a \in \{1,\ldots,N-1\}$ и вычислить $(a,N) = d$. Если $d > 1$, то N— составное.

4. При $d = 1$, проверяется

$$a^{\frac{N-1}{2}} \equiv \frac{a}{N} \mod N . \qquad (2)$$

Если предоставленное сравнение не выполняется, то N— составное, иначе неизвестно.

Для случае неизвестно тест повторяется. Алгоритм 3.2 для фиксированного числа $s > 1$ различных значений а применяется на практике.

Результат выполнения теста Соловея-Штрассена реализованной на языке C++ показан на рис. 3.4.

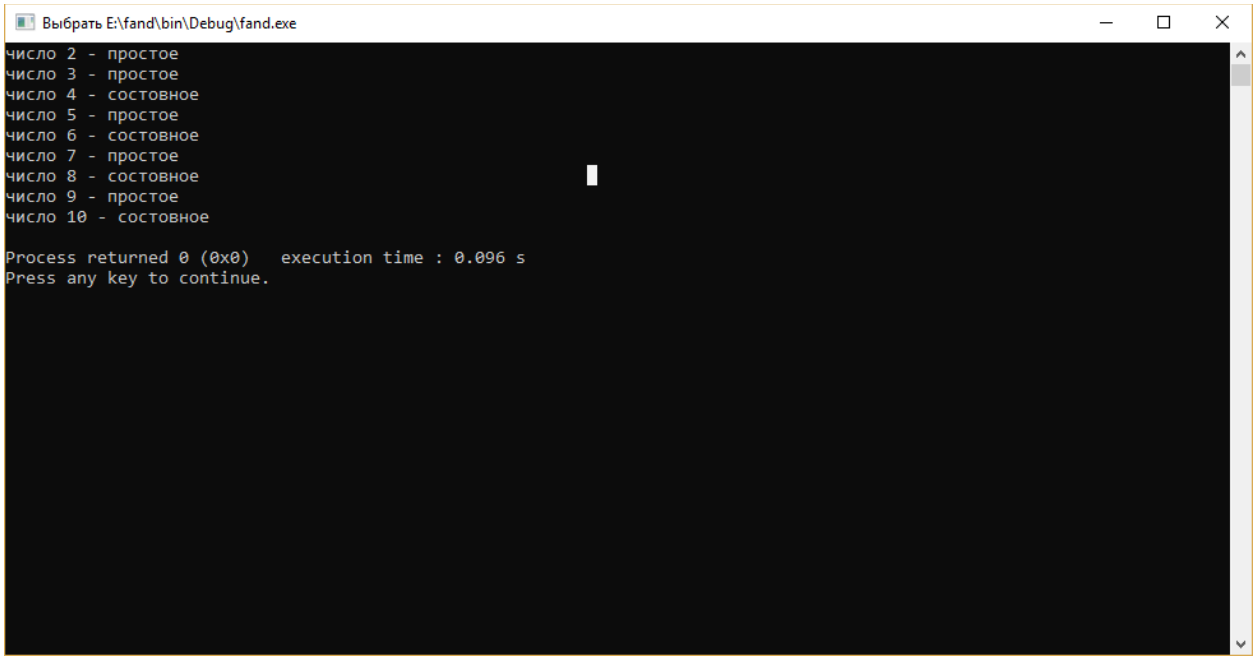

Рисунок 3.4 - Результат выполнения теста Соловея и Штрассена

3.3 Алгоритм теста Миллера и Рабина

Реализация теста Миллера и Рабина была выполнена на основе алгоритма, изображённого в виде блок схемы на рис.3.5.

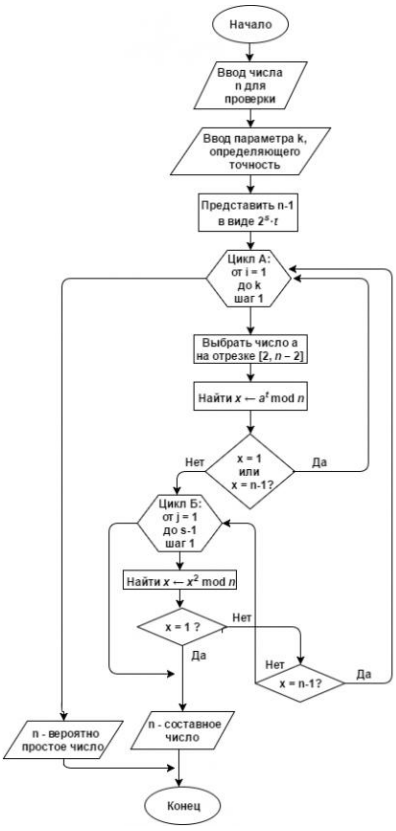

Рисунок 3.5 – Блок схема

Пошаговое выполнение алгоритма теста Миллера-Рабина:

Алгоритм 3.3.

3. Случайно выбрать число $a \in \{1,\ldots,N-1\}$ и вычислить $(a,N)=d$. Если $d>1$, то N— составное.

4. При $d=1$, то вычислить

$r_k \equiv a^{2^k u} (\mathrm{mod}\ N)$ для $k \in \{0,..,t-1\}$.

Если $r_0 \equiv 1(mod\ N)$ или $r^k \equiv -1(mod\ N)$ при некотором $k \in \{0,\ldots,t-1\}$, то неизвестно, иначе N— составное.

Если алгоритме 3.3 неизвестно, то тест повторяется. На практике обычно применяют алгоритм 3.3 для концентрированного числа $s>1$ разных значений a.

Результат выполнения теста Миллера и Рабина реализованной на языке C++ показана на рис. 3.5.

Рисунок 3.5 - Результат выполнения теста Миллера и Рабина

Заключение

Бакалаврская работа посвящена алгоритмам проверки простоты целых чисел. Простые числа используется в криптосистемах с открытым ключом.

Целью данной работы были алгоритмы простоты целых чисел. Для этого в работе были рассмотрены следующие вопросы: распределение простоты чисел в натуральном ряду, критерии простоты. Тесты основаны на некоторых критериях простоты. Описаны следующие тесты простоты целых чисел: вероятностные тесты простоты, тест простоты Ферма, тест простоты Соловея и Штрассена, тест простоты Миллера и Рабина, полиномиальный тест распознавания простоты.

Результатом работы являются реализации алгоритмов тестов Ферма, Соловея и Штрасена, Миллера и Рабина на языке C++.

Задачи, выполненные в ходе данной работы, позволили в итоге реализовать критерии проверки простоты целых чисел и понять в каких аспектах и какой из, них имеет преимущество.

Подводя итог, можно констатировать, что тест Миллера-Рабина считается более результативным и универсальным из тех, которые рассматриваются в этой бакалаврской работе.

Список используемых источников

Учебники и учебные пособия

1. Коблиц Н. Курс теории чисел и криптографии / Н. Коблиц, - М.: научное изд-во ТВП, 2001
2. Андерсон Дж. Дискретная математика и комбинаторика / Дж. Андерсон, - Вильямс 2003.
3. Додонова Н.Л. Конспект лекций по дисциплине алгебраические структуры и теория чисел / Л.Н. Додонова, - Самара, 2016
4. Найер Б.М. Прикладная криптография / М.Б. Найер, - ТРИУМФ 2002. – 816 с
5. Глухов М.М. Введение в теоретико-числовые методы / М.М. Глухов, И.А. Круглов, А.Б. Пичкур, А.В. Черемушкин, - СПБ.: Лань, 2016. – 400 с.
6. Василенко О.Н. Теоретико-числовые алгоритмы в криптографии / О.Н. Василенко. – 2-е изд. – М.: МЦНМО, 2007.
7. Черемушкин А.В. Лекции по арифметическим алгоритмам в криптографии / А.В. Черемушкин. – М.: МЦНМО, 2002. – 104 с.
8. Ященко В. В. Основные понятия криптографии / Математическое просвещение. Сер. 3. №2. 1998. С. 53-70.
9. Виноградов И. М. Основы теории чисел / М.И. Виноградов, - М.: Наука. 1972.
10. Карацуба А. А. Основы аналитической теории чисел / А.А. Карацуба, - М.: Наука. 1983 г.
11. Василенко О. Н. Современные способы проверки простоты чисел / Н.О. Василенко, - 1988. С. 162-188.
12. Прахар К. Распределение простых чисел / К. Прахар, - М.: Мир. 1967.
13. Боревич З.И. Шафаревич И.Р. Теория чисел / И.З. Боревич, - М.: Наука. 1964.

14. Бухштаб А.А. Теория чисел / А. А. Бухштаб. — М.: Просвещение, 1966.

15. Болотов *А.А.* Элементарное введение в эллиптическую криптографию: Алгебраические и алгоритмические основы / А. А. Болотов — М.: Ком Книга, 2006. — 328 с.

16. Соловьев *Ю.П.* Эллиптические кривые и современные алгоритмы теории чисел / Ю. П. Соловьев, В. А. Садовничий, Е. Т. Шавгулидзе. — Ижевск: ИКИ, 2003.

Литература на иностранном языке

17. Maurer U. M. Fast generation of prime numbers and secure public key cryptographic parameters / U. M. Maurer / J. cryptology. — 1995. — 8. — P. 123–155

18. Menezes A. J. Reducing elliptic curve logarithms to loga rithms in a finite field / A. J. Menezes, P. C. van Oorschot, S. A. Vanstone / IEEE Trans. on Information Theory. — 1993. — 39. — P. 1639–1646.

19. Ngugen P. Lattice reduction in cryptology: an update / P. Ngugen, J. Stern / Algorithmic number theory. Proceed ings of ANTS IV, Lecture notes in computer science. — 2000. — N 1838. P. 85–112.

20. Fouvry E. Theoreme de Brun–Titchmarsh; application au theoreme de Fermat / E. Fouvry / Invent. Math. — 1985. — 79. — P. 383–407.

21. Dennij T. On the reduction of composed relations from the number field sieve / T. Dennij, V. Muller / Proceedings of ANTS II. — 1996. — Lect. Notes in comp. sci. vol. 1122. — P. 75–90.

22. Coppersmith D. Fast evaluation discrete logarithms in field characteristic two / D. Coppersmith / IEEE Trans. On inform. Theory. — 1984. — 30. — P. 587–594.

Алгоритм проверки числа на простоту при помощи теста Ферма

```
#includ<iostream>
#includ<cmath>
#includ<cstdlib>
#includ<iomanip>
#includ<ctime>
usingnamespace std;
boolprime(long long n){
 for(longlongi=2;i<=sqrt(n)i++)
        if(n%i==0)
                returnfalse;
 returntrue;
}

longlonggcd(longlonga,longlongb){
 if(b==0)
        return a;
 returngcd(b,a%b);
}

longlongmul(longlonga,longlongb,longlongm){
 if(b==1)
        returna;
 if(b%2==0){
        longlongt=mul(a,  b/2, m);
        return(2 * t)% m;
 }
 return(mul(a,  b-1,m)+a) % m;
```

```
}

longlongpows(longlong a,long longb, longlong m){
 if(b==0)
        return 1;
 if(b%2==0){
        longlongt =pows(a, b/2, m);
        return mul(t , t, m) % m;
 }
 return(mul(pows(a, b-1,m) , a, m)) % m;
}

boolferma(longlong x){
 if(x==2)
        returntrue;
 srand(time(NULL))
 for(inti=0;i<100;i++){
        longlong a = (rand() % (x - 2)) + 2;
        if(gcd(a, x) != 1)
                returnfalse;
        if( pows(a,x-1, x) != 1)
                returnfalse;
 }
 returntrue;
}

int main()
{
   setlocale(LC_ALL, "rus");
```

```
    cout << setiosflags(ios::left) << setw(10) << "Числа" << setw(10) << "Простые числа(1/0)?" <<endl;
    for(int i = 0; i <10000; i++)
    {
       cout<<setiosflags(ios::left) << setw(10)<< i <<setw(10) << ferma(i) << endl;
    }
    return 0;
}
```

Алгоритм проверки числа на простоту при помощи теста Соловея — Штрассена

```
#define N 11

using namespace std;

void formulaSravneniya();
int NOD(int x);

int main()
{
    setlocale(0, "rus");
    srand(time(0));
    formulaSravneniya();
    return 0;
}

int NOD(int x)
{
```

```
    long nod;
    for (int i = x; i > 0; i--) {
        if (x % i == 0 && N % i == 0) {
            cout << "nod = " << i << endl;
            nod = i;
            break;
        }
    }
    return nod;
}


bool res(int r, int s)
{
    if(r == s)
        return true;
}

void formulaSravneniya()
{
    int x = rand()%N;
  //  cout << " x = " << x << endl;
    if((NOD(x) != 1) && (NOD(x) != (N-1))){
        cout << N << " - составное";
    }
    else if(x == 1){
        int temp, r, s;
        temp = pow(x, (N-1)/2);
        r = (x/N);
```

```
        cout << r;
        s = temp % N;
        if(res(r, s) == true){
            int y = rand()%N + 1;
            cout << "y=" << y << endl;
            if(NOD(y) > 1){
                cout << N << " - составное";
            }
            else if(y == 1){
                int temp2, r2, s2;
                temp2 = pow(y, (N-1)/2);
                r2 = (y/N);
                s2 = temp % N;
                if(res(r2, s2) == true){
                    cout << N << " - простое";
                }
            }
        }
    }
}
```

Алгоритм проверки числа на простоту при помощи теста Миллера и Рабина

```
#include<iostream>
#include<vector>
#include<cstdlib>
#include<ctime>
#include<math.h>
usingnamespacestd;
```

```
intgcd(int a, int b) {
return b?gcd(b, a % b) : a;
}
voidalgo () {
intn, p = 1;
cin>>n;
srand(timeNULL));
while(true)  {
p=1;
srand(time(NULL));
int a=rand() % (n - 1) + 2;
if(gcd(a, n) > 1) {
p= 0;
break;
}
else{
int ch=pow((double)a, n - 1);
if (ch %n!= 1) {
p= 0;
break;
}
}
}
if (p== 1)
cout<< "prime";
else
cout <<"not prime";
}
intmain  () {
```

```
algo();
return0;
}
```

Алгоритм проверки числа на простоту при помощи теста Миллера и Рабина

```
#include<bits/stdc++h>
usingnamespace std;
typedefunsigned long long ll;
llpows[70];
const ll INF= 1e10 + 89;
void fil(){
    pows[0] =1ll;
    for (int i =1; i <= 63; ++i)
        pows[i]= pows[i - 1] * 2ll;
}
ll mul_mod(lla,  ll b, ll mod) {
    ll res= 0;
    while(b)  {
        if (b& 1)
            res= (res + a) % mod;
        a = (a* 2ll) % mod;
        b>>=1;
  }
  return res;
}
ll powm(ll p,ll n, ll mod) {
    ll res= 1;
    while(n)  {
```

```
        if (n& 1)
            res=mul_mod(res, p, mod);
    p = mul_mod(p, p, mod);
     n>>= 1;
 }
    returnres;
}
bool mrt(llp) {
    if (p ==1)
        returnfalse;
    ll s =0, d = p - 1;
    while(!(d & 1)) {
     d= d / 2;
     s++;
  }
  srand(time(NULL));/// Генерируем действительно случайные и разные a
    for(int i = 1; i <= 5; ++i) { /// Те самые 5 проверок
        lla = rand() % p;
        if(a == 0)
          a++;
        boolflag1 = false, flag2 = false;
        if(powm(a % p, d, p) == 1) /// Первая проверка
            flag1= true;
        if(!flag1)
            for(int r = 0; r < s; ++r) /// Вторая проверка
                if(powm(a % p, pows[r] * d, p) == p - 1)
                flag2= true;
        if (!flag1&& !flag2)
            returnfalse;
```

```
  }
  return true;
}
intmain()  {
   llm, n;
   fil();
    cin>> m;
    if(mrt(m))
        cout<<"YES" << '\n';
    else
        cout<<"NO";
}
```