Компьютерная алгебра

(курс лекций)

Игорь Алексеевич Малышев

Computer.Algebra@yandex.ru

Лекция 12

Факторизация целых чисел

Содержание лекции

- Алгоритм Евклида и цепные дроби
- Простые числа. Решето Эратосфена. Тесты простоты
- Разложение целых чисел на множители

План лекции: тема подраздела

- **Алгоритм Евклида и цепные дроби**
- Простые числа. Решето Эратосфена. Тесты простоты
- Разложение целых чисел на множители

Алгоритм Евклида и цепные дроби

Порождение цепной дроби – 1

Рассмотрим произвольную рациональную дробь a_0 / a_1 ,
записанную в несократимом виде, т.е. $(a_0, a_1) = 1$ и $a_1 > 0$

Применив к паре a_0 , a_1 алгоритм Евклида, получим :

$$
\begin{array}{ll}
a_0 = a_1 * c_0 + a_2 & 0 < a_2 < a_1 \\
a_1 = a_2 * c_1 + a_3 & 0 < a_3 < a_2 \\
a_2 = a_3 * c_2 + a_4 & 0 < a_4 < a_3 \\
. \quad . \quad . \quad . \quad . \quad . & \\
a_{k-2} = a_{k-1} * c_{k-2} + a_k & 0 < a_k < a_{k-1} \\
a_{k-1} = a_k * c_{k-1} &
\end{array}
$$

Замечание. Обозначения в формулах отличаются от ранее использованных (см. лекция 6) : вместо q_1 , .. q_k указаны c_0 , … c_{k-1}

Положим $\xi_i = a_i / a_{i+1}$ для всех i в пределах $0 \le i \le k - 1$.
Тогда приведённые выше равенства примут вид :

$$\xi_i = c_i + 1 / \xi_{i+1} \qquad 0 \le i \le k - 2 \qquad \xi_{k-1} = c_{k-1}$$

Алгоритм Евклида и цепные дроби

Порождение цепной дроби – 2

Далее, если в выражении : $\xi_0 = c_0 + 1 / \xi_1$
заменить ξ_1 на $c_1 + 1 / \xi_2$, то получим : $\xi_0 = c_0 + 1 / (c_0 + 1 / \xi_1)$

Продолжая этот процесс переобозначений,
получим представление a_0 / a_1 в виде цепной дроби :

$$\frac{a_0}{a_1} = \xi_0 = c_0 + \cfrac{1}{c_1 + \cfrac{1}{\ddots \; \overline{\qquad} \; + \cfrac{1}{c_{k-2} + \cfrac{1}{c_{k-1}}}}}$$

Алгоритм Евклида и цепные дроби

Порождение цепной дроби – 3

Как и ранее (см. лекция 6), целые числа c_i
называются неполными частными.

Т.к. в общем случае a_0 может не быть положительным
(предполагается, что $a_1 > 0$),
то c_0 может быть положительным,
отрицательным или нулевым.

Однако, т.к. в алгоритме Евклида $0 < a_2 < | a_1 |$,
то все частные $c_1 , c_2 , \ldots, c_{k-1}$ будут положительны.

Замечание. Для цепной дроби, приведённой выше,
мы будем использовать следующее обозначение :

$$(c_0 ; c_1 , c_2 , \ldots, c_{k-1})$$

Алгоритм Евклида и цепные дроби

Два представления рациональных чисел

Всякое рациональное число
имеет только два представления в виде цепной дроби :

$a_0 / a_1 = (c_0 ; c_1 , c_2 , \ldots, c_{k-2} , c_{k-1}) = (c_0 ; c_1 , c_2 , \ldots, c_{k-2} , c_{k-1} - 1 , 1)$

Пример. Рассмотрим рациональную дробь 8 / 5.
Представим его в виде цепной дроби двумя способами :
8 / 5 = (1 ; 1 , 1, 2)
8 / 5 = (1 ; 1 , 1 , 1 , 1)

Замечание. Условие, что $c_1 , c_2 , \ldots c_{k-1}$ положительны,
не является общепринятым.
Если отказаться от него, то дробь (– 8 / 5)
также может быть представлена в виде цепных дробей :
(– 2 ; 2 , 2) и (– 1 ; – 1 , – 1 , – 2)

Алгоритм Евклида и цепные дроби

Свойства конечных цепных дробей

Определение. Целой частью $[\,c\,]$ числа c называется :

$$[\,c\,] = \{ \lfloor c \rfloor, \text{ если } c \geq 0 \;;\; \lceil c \rceil, \text{ если } c < 0 \}$$

Теорема (единственность).
Если $(c_0 ; c_1 , c_2 , \ldots, c_m) = (d_0 ; d_1 , d_2 , \ldots, d_n)$
и если $c_m > 1$ и $d_n > 1$,
то $m = n$ и $c_i = d_i$ для $i = 0 , 1 , \ldots , n$

Теорема. Любая конечная цепная дробь
представляет рациональное число,
и, обратно, всякое рациональное число может быть представлено
в виде конечной цепной дроби, причём ровно двумя способами.

Алгоритм Евклида и цепные дроби

Свойства бесконечных цепных дробей – 1

Теорема. Любое иррациональное число ξ
представимо единственным образом
в виде бесконечной цепной дроби $(c_0 ; c_1 , c_2 , \ldots, c_m , \ldots)$,
где значения c_i вычисляются с помощью следующего алгоритма :

Положим $\xi_0 = \xi$

Определим по индукции для $i \geq 0$:

$$c_i = \lfloor \xi_i \rfloor$$
$$\xi_{i+1} = 1 / (\xi_i - c_i)$$

Верно и обратное утверждение :
всякая бесконечная цепная дробь,
заданная числами c_i , $c_i > 0$ для любого i ,
представляет иррациональное число ξ .

Алгоритм Евклида и цепные дроби

Свойства бесконечных цепных дробей – 2

Если положить :

$p_{-2} = 0$ $\quad$ $p_{-1} = 1$ $\quad$ $p_i = c_i * p_{i-1} + p_{i-2}$ $\quad$ $i \geq 0$

$q_{-2} = 0$ $\quad$ $q_{-1} = 1$ $\quad$ $q_i = c_i * q_{i-1} + q_{i-2}$ $\quad$ $i \geq 0$

то конечная цепная дробь $(c_0 ; c_1 , c_2 , \ldots, c_n)$
будет иметь рациональное значение

$r_n = p_n / q_n$ $\quad\quad$ $(p_n , q_n) = 1$

которое называется **n-й подходящей дробью**
иррационального числа ξ .

Знаменатели q_n подходящих дробей
образуют возрастающую последовательность
положительных целых чисел для $n > 0$
и выполняются следующие соотношения (см. следующий слайд).

Алгоритм Евклида и цепные дроби

Свойства бесконечных цепных дробей – 3

(A) Если $\xi = (c_0; c_1, \dots, c_{n-1}, \xi_n)$,
где $\xi_n = (c_n; c_{n+1}, \dots)$, $n \geq 0$, то

$$\xi = \frac{p_{n-1} * \xi_n + p_{n-2}}{q_{n-1} * \xi_n + q_{n-2}}$$

(B) $$\xi = c_0 + \cfrac{1}{c_1 + \cfrac{1}{\cdots\cdots \cfrac{}{+ \cfrac{1}{c_{n-1} + \cfrac{1}{\xi_n}}}}}, \quad n \geq 1$$

(C) $$\frac{p_n}{q_n} = c_0 + \cfrac{1}{c_1 + \cfrac{1}{\cdots \cfrac{}{+ \cfrac{1}{c_{n-1} + \cfrac{1}{c_n}}}}}, \quad n \geq 0$$

Алгоритм Евклида и цепные дроби

Свойства бесконечных цепных дробей – 4

Всякая периодическая цепная дробь
является **квадратичным** иррациональным числом, и обратно.

Замечание. Квадратичным иррациональным числом называется число вида :

$$(p \pm \sqrt{d}) / q$$

являющееся корнем квадратного полиномиального уравнения :

$$q^2 x^2 - 2 p q x + (p^2 - d)$$

где d – целое положительное число, не являющееся точным квадратом.

Замечание. Алгоритм Евклида может быть использован
только для разложения в цепную дробь рационального числа.
Однако имеется более общая процедура, которая может быть использована
как для рационального, так и для иррационального числа.

Алгоритм Евклида и цепные дроби

Обобщение алгоритма Евклида для цепных дробей

Дано число x (рациональное или иррациональное).

Вычислим c_0 – наибольшее целое число, не превосходящее x
и представим x в форме :

$$x = c_0 + 1 / x_1 \quad , \quad 0 < 1 / x_1 < 1$$

где число $x_1 = 1 (x – c_0) > 1$ иррационально,
если число x иррационально.

После этого вычислим c_1 – наибольшее целое число,
не превосходящее x_1 и представим x_1 в форме :

$$x_1 = c_1 + 1 / x_2 \quad , \quad 0 < 1 / x_2 < 1 \quad , \quad c_1 \geq 1$$

где число $x_2 = 1 (x_1 – c_0) > 1$ тоже может быть иррациональным.

Продолжая этот процесс, мы получим представление числа x в виде цепной дроби :

$$x = (c_0 ; c_1 , c_2 , \ldots)$$

которая может быть конечной или бесконечной в зависимости от того,
является ли x рациональным числом или нет.

Алгоритм Евклида и цепные дроби

Примеры цепных дробей

Замечание. Для подходящих дробей при разложении иррациональных чисел, а также для рациональных чисел выполняется следующее неравенство :

$$| x - p_n / q_n | < 1 / (q_n)^2$$

Пример. Вычисление первых членов разложения числа π в цепную дробь.

$$\pi = (3 ; 7 , 15 , 1 , 292 , 1 , 1 , 1 , 2 , 1 , 3 , 1 , 14 , 2 , \ldots)$$

Замечание. Для иррационального числа не всегда можно получить его полное разложение в цепную дробь (т.к. алгоритм Евклида в этом случае не применим). Однако, если известно десятичное приближение такого числа, то можно вычислить соответствующую часть цепной дроби, представляющей это число.

Пример. Разложение в цепную дробь числа e обладает (в отличие от числа π) замечательной регулярностью, которую открыл Эйлер :

$$e = (2 ; 1 , 2 , 1 , 1 , 4 , 1 , 1 , 6 , 1 , 1 , 8 , 1 , 1 , \ldots)$$

План лекции: тема подраздела

- Алгоритм Евклида и цепные дроби
- **Простые числа. Решето Эратосфена. Тесты простоты**
- Разложение целых чисел на множители

Задачи факторизации целых чисел

Замечание. Задача факторизации (разложения на множители) целых чисел является одной из основных в теории чисел
и благодаря своей простой формулировке известна с античности.

Простейшая формулировка задачи факторизации целых чисел

Дано : целое число n > 0
Найти : (если это возможно) два целых числа a и b , таких, что a * b = n

Замечание. Фактически имеем не одну, а две различные задачи :

(1) Тест на простоту. Проверка существования целых чисел a и b .

(2) Разложение на множители. Вычисление таких целых чисел a и b.

Простые числа

Основные определения

Определение 1. Ненулевое целое число $a \neq \pm 1$ называется **неразложимым**, если все его делители тривиальны, т.е. его делителями являются только следующие числа: ± 1 и $\pm a$.

Пример. Числа 13 и -7 являются неразложимыми.

Определение 2. Ненулевое целое число $a \neq \pm 1$ называется **разложимым** или **составным**, если у него есть нетривиальные делители, т.е. оно может быть представлено в виде : $a = b * c$, где b и c не равны ± 1 и $\pm a$.

Пример. Число $276 = 12 * 23$ – является составным.

Замечание. Делители также называют **сомножителями**.

Замечание. Составное число может быть представлено в виде произведения нетривиальных сомножителей. Если, в свою очередь, сомножители тоже разложимы, то они также могут быть представлены в виде произведения нетривиальных сомножителей и т.д.

Простые числа

Неприводимое разложение

Теорема 1. (существование неприводимого разложения). Всякое ненулевое целое число $a \neq \pm 1$ может быть представлено как $\pm$ произведение конечного числа положительных неразложимых целых чисел, т.е. $a = u_1 * u_2 * \ldots * u_r$, где $u_i > 1$ для $i = 1, \ldots, r$ и неразложимы.

Пример. Число 1008 = 2 * 2 * 2 * 2 * 3 * 3 * 7

Является ли неприводимое разложение единственным ?

Определение. Целое число $p > 1$ называется простым, если для любых a и b из того, что $p \mid (a * b)$, следует, что $p \mid a$ или $p \mid b$.

Утверждение. Целое число $p > 1$ является простым тогда и только тогда, когда оно неразложимо.

Теорема 2. (единственность неприводимого разложения). Всякое ненулевое целое число $a \neq \pm 1$ может быть представлено как $\pm$ произведение простых чисел только одним способом, с точностью до порядка сомножителей.

Замечание. Теорему 2 называют **основной теоремой арифметики**.

Простые числа

Проблема единственности разложения

Множество целых чисел **Z**
является **областью с однозначным разложением на множители**.
Однако существуют математические структуры,
в которых утверждение о единственности неприводимого разложения
не выполняется.

Пример. Рассмотрим множество E,
элементами которого являются положительные чётные числа 2, 4, 6, 8, …
Очевидно, что множество E замкнуто относительно умножения.
Пусть все числа, которые мы знаем, являются элементами множества E.
Тогда число 8 = 2 * 4 является «составным»,
а число 14 – «простым» (т.к. оно не является произведением «чисел»).
Более того, число 840 имеет два разложения на «простые» числа :

840 = 2 * 14 * 30 = 6 * 10 * 14

Таким образом, теорема о единственности разложения не выполняется.

Простые числа

Произведение степеней простых чисел – 1

Основная теорема арифметики определяет форму представления целого числа – **разложение числа** a **в произведение степеней простых чисел** :

$$a = \pm (p_1)^{e_1} * (p_2)^{e_2} * \ldots * (p_k)^{e_k}$$

где p_i – различные простые числа;
e_i – положительные целые числа.

Замечание. Иногда удобнее считать, что показатели степеней могут быть равными нулю.

Выразим НОД и НОК целых чисел a и b с помощью такого представления.

Пусть $a = (p_1)^{e_1} * (p_2)^{e_2} * \ldots * (p_k)^{e_k}$ и $b = (p_1)^{f_1} * (p_2)^{f_2} * \ldots * (p_k)^{f_k}$
где $e_i \geq 0$ и $f_i \geq 0$.

Тогда, т.к. $p^m \mid p^n$ если и только если $m \leq n$, то имеем :

$$\text{НОД}(a, b) = (a, b) = (p_1)^{\min(e_1, f_1)} * (p_2)^{\min(e_2, f_2)} * \ldots * (p_k)^{\min(e_k, f_k)}$$
$$\text{НОК}(a, b) = [a, b] = (p_1)^{\max(e_1, f_1)} * (p_2)^{\max(e_2, f_2)} * \ldots * (p_k)^{\max(e_k, f_k)}$$

Простые числа

Произведение степеней простых чисел – 2

Далее, используя тот факт, что :

$$| a * b | = (p_1)^{(e_1 + f_1)} * (p_2)^{(e_2 + f_2)} * \ldots * (p_k)^{(e_k + f_k)}$$

а также следующее равенство :

$$\min (a , b) + \max (a , b) = a + b$$

получим следующую теорему.
Теорема (существование наименьшего общего кратного).
Если a и b – ненулевые числа, то для них существует наименьшее общее кратное,
и справедливо следующее равенство :

$$[a , b] = | a , b | / (a , b)$$

Замечание. Целое число $m > 0$ называется НОК ненулевых целых чисел a и b, если :

(1) $a \mid m$ и $b \mid m$

(2) если $a \mid c$ и $b \mid c$, то $m \mid c$

Единственность НОК следует из (2) и из того, что $m > 0$.

Простые числа

Формула вычисления функции Эйлера

Замечание. В лекции 6 указано, что функция Эйлера определяет для заданного целого числа $n > 0$ количество положительных целых чисел m, таких, что $m \le n$ и $(m, n) = 1$ (т.е. m и n – взаимно простые числа).

Замечание. Для всякого простого числа p:
$\phi(p) = p - 1$, где $\phi(n)$ – функция Эйлера.

Функцию Эйлера можно выразить с помощью разложения числа в произведение степеней простых чисел в виде формулы :

$$\phi(n) = n - (n / p_1) - (n / p_2) - \ldots - (n / p_k) + \\ + (n / (p_1 * p_2)) + (n / (p_1 * p_3)) + \ldots + (n / (p_{k-1} * p_k)) + \ldots + \\ + (-1)^k (n / (p_1 * p_2 * \ldots * p_k))$$

где $n = (p_1)^{e_1} * (p_2)^{e_2} * \ldots * (p_k)^{e_k}$

Выполнив факторизацию указанной формулы, получим :
$\phi(n) = n * (1 - 1 / p_1) * (1 - 1 / p_2) * \ldots * (1 - 1 / p_k)$

Простые числа

Решето Эратосфена

Замечание. Поиск простых чисел обычно производят методом решета. Впервые такой метод предложил древнегреческий математик Эратосфен (III век до н.э.)

Для того, чтобы найти все простые числа $\leq n$, запишем по порядку все целые числа от 2 до n .

Затем вычеркнем все чётные числа, кроме числа 2, т.к. они делятся на 2 и потому не являются простыми. Потом вычеркнем все числа, кратные 3, и так далее.

После i-го прохода будут вычеркнуты все числа, которые делятся на первые i простых чисел $p_1, p_2, \ldots, p_i$
Первое число $x > p_i$, которое останется невычеркнутым, будет $(i + 1)$-м простым числом.
Затем будут вычеркнуты все числа, кратные p_{i+1} .

Процесс остановится, когда в списке не останется невычеркнутых чисел, которые больше последнего найденного простого числа.
Целые числа, которые остались в списке, прошли сквозь решето и являются простыми $\leq n$.

Простые числа

Эффективность метода простого решета

Метод решета Эратосфена можно использовать для проверки простоты заданного числа n (результат проверки зависит от того, будет ли оно вычеркнуто).

Однако такой метод не эффективен для проверки наибольших из известных простых чисел.

Замечание. Учитывая, что у любого составного числа n
обязательно есть простой делитель $\leq \sqrt{n}$

(что обусловлено свойством парности делителей: если число имеет делитель больший, чем квадратный корень из него, оно также должно иметь делитель, меньший чем этот корень)

при вычёркивании чисел, кратных p_i, мы можем рассматривать только простые числа $\leq \sqrt{n}$
и процесс остановится, когда $(p_i)^2 \geq n$ для некоторого i.
Таким образом можно получить улучшенный метод решета Эратосфена.

Пример. Для проверки 13395-значного числа $(2^{44497} - 1)$,
простота которого была доказана в 1979 году,
компьютеру, выполняющему 1 миллион операций в секунду,
потребовалось бы 10^{6684} лет для решения задачи по улучшенному методу решета Эратосфена.

Простые числа

Генерация больших простых чисел

В ряде приложений (в частности, в теории кодирования)
требуется уметь находить n наибольших простых чисел $\leq M$,
где M – максимальное целое число,
представимое аппаратными средствами компьютера.

Для нахождения таких простых чисел,
сначала методом решета строится таблица простых чисел $\leq \sqrt{M}$.
Затем кратные каждого из этих простых чисел вычёркиваются из списка
L , L+1 , … , M
Целые числа, оставшиеся невычеркнутыми, будут простыми.

Чтобы вышеописанный метод был практически реализуем, необходимо :

(1) Список простых чисел $\leq M$ должен быть не слишком велик.
(2) Интервал (L , M) должен быть достаточно велик,
чтобы содержать как минимум n простых чисел.
(3) Интервал (L , M) должен быть не слишком большим для небольшого n .

Для достижения требований (1) (2) (3) может быть использована следующая теорема.

Простые числа

Теорема о количестве простых чисел

Теорема. Обозначим PRIMES (x) – количество простых чисел ≤ x . Тогда верно равенство :

$$\lim_{x \to \infty} (\text{PRIMES} (x) / (x / \ln (x))) = 1$$

По сути, теорема утверждает, что из каждых ln (M) чисел одно является простым.

Таким образом, интервал, достаточно превышающий по длине (n * ln (M)) ,
должен содержать n простых чисел.

При этом размер таблицы простых чисел примерно равен $(\sqrt{M} / \ln (\sqrt{M}))$.

Пример. Пусть n = 10 . Тогда для компьютера, имеющего разрядность 32 бита,
получим следующие характеристики генератора больших простых чисел :

Оценка длины интервала :

$n * \ln (M) = 10 * \ln (2^{31}) \approx 214$ (выберем интервал длины 500)

Оценка размера таблицы простых чисел :

$\sqrt{M} / \ln (\sqrt{M}) = \sqrt{(2^{31})} / \ln (\sqrt{(2^{31})}) \approx 4314$

Фактическое количество простых чисел :

$\text{PRIMES} (\sqrt{(2^{31})}) = 4691$

Простые числа

Алгоритм генерации простых чисел

Вход:
целые числа k и m одинарной точности
одномерный массив A длины k
m – нечётное целое число ≥ 3

Выход:
простые числа $p_1 < p_2 < \ldots < p_r$
из замкнутого отрезка
[m , m + 2 * k – 2]

[Инициализация]
n := m + 2 * k – 2 d := 3
Для i := 1 , 2 , … , k { A (i) := 1 }

[Условие завершения]
Если $d^2 > n$ **то** получить простые числа **Конец**
Если d > [n / d] **то** перейти к шагу **[Получение простых чисел]**

[Вычисление наименьшего положительного числа j , такого, что d | (m + 2 * j – 2) и (m + 2 * j – 2) ≥ 3]
r := m mod d j := 1
Если r > 0 и r – чётно **то** j := j + d – (r / 2)
Если r > 0 и r – нечётно **то** j := j + (d – r) / 2
Если m ≤ d **то** j := j + d

[Вычёркивание составных чисел]
Для i := j , j + d , j + 2 * d , … **Пока** i < k { A (i) := 0 }

[Изменение d]
Если (d mod 6) = 1 **то** d : = d + 4 **иначе** d := d + 2
Перейти к шагу **[Условие завершения]**

[Получение простых чисел]
Для i := k , k-1 , k-2 , … , 1 { **Если** A (i) = 1 **то** выдать простое число m + 2 * i – 2 } **Конец**

Простые числа

Сложность алгоритма генерации простых чисел

Все арифметические операции в алгоритме выполняются с короткими целыми числами, поэтому сложность каждой операции равна O (1).

Шаги **[Инициализация]** и **[Получение простых чисел]**
выполняются только один раз, каждый из них включает k операций,
Поэтому сложность обоих шагов равна O (k).

Шаги алгоритма от [Условие завершения] до [Изменение d]
образуют цикл, для которого число $\sqrt{n}$, где $n = m + 2 * k – 2$
является верхней оценкой количества повторений.

Кроме того, при каждом выполнении тела цикла
шаг **[Вычёркивание составных чисел]** включает не более k операций,
а остальные шаги цикла – одну или две операции.
Поэтому сложность выполнения всего цикла равна $O (k * \sqrt{n})$.

Объединяя результаты анализа, получим итоговую оценку сложности алгоритма : **$O (k * \sqrt{n})$**.
($n = m + 2 * k – 2$)

Простые числа

Применение алгоритма генерации простых чисел

Пример. Найдём все простые числа между 3 и 21.
(т.е. в случае $m = 3$ $k = 10$)
После шага **[Инициализация]** $d = 3$ $A(i) = 1$ для всех $i = 1, \dots, 10$
На шаге **[Условие завершения]** условие окончания не выполняется, поэтому переходим к следующему шагу.
Получаем $r = 0$ $j = 4$ (т.к. $m = d$)
На шаге **[Вычёркивание составных чисел]** изменяем массив A :

$A(4) := 0$
$A(7) := 0$
$A(10) := 0$

На следующем шаге изменяем $d := 5$

В начале второго выполнения цикла проверка условия окончания успешна, поэтому переходим к шагу **[Получение простых чисел]** :
Для $i = 10$ выдавать нечего, т.к. $A(10) = 0$
Для $i = 9$, $A(9) = 1$, поэтому выдаём простое число $19 = 3 + 2 * 9 – 2$
Аналогичным образом вычисляем и выдаём остальные простые числа ≤ 21 :
17 , 13 , 11 , 7 , 5 , 3

Простые числа

Обзор тестов простоты

Существуют две группы алгоритмов
проверки простоты положительных целых чисел :

(1) Детерминированные тесты (ответ получаем всегда)

(2) Вероятностные тесты (ответ получаем с некоторой вероятностью)

Замечание. Практически все тесты простоты
так или иначе связаны с малой теоремой **Пьера Ферма**.

Малая теорема Ферма (1640). Если m – простое число
и a – произвольное целое число, не делящееся на m ,
то $a^{m-1} \equiv 1 \pmod{m}$ (в других обозначениях: $a^{m-1} \equiv_m 1$)

Замечание. По определению при фиксированном $m > 1$ $b \equiv_m a$
(т.е. b сравнимо с a по модулю m) тогда и только тогда,
когда m делит $(b – a)$.

Замечание. Систематизация тестов проверки простоты к настоящему времени не завершена,
поэтому будут рассмотрены отдельные представители групп тестов.

Простые числа

Первый детерминированный тест

Замечание. Другое название данного теста – метод пробных делений.

Замечание. Здесь и далее будем обозначать тестируемое число – m.

Алгоритм тестирования :
Разделим число m последовательно на числа $2, 3, \ldots, \lfloor \sqrt{m} \rfloor$
Если при каком-нибудь делении, получим нулевой остаток,
то число составное, а делитель и частное являются его сомножителями.
В противном случае число m простое.

Сложность алгоритма тестирования :
Очевидно, необходимо выполнить $\sqrt{m}$ делений,
поэтому оценка сложности алгоритма равна $O(\sqrt{m})$.

Однако эта оценка не является полиномиальной, т.к. учитывая разрядность представления m ,
имеем оценку битовой сложности – $O(2^{L(m)/2})$.

Замечание. Метод пробных делений не только определяет, является ли число простым,
но и находит сомножители составного числа.

Простые числа

Второй детерминированный тест – 1

Замечание. Другое название данного теста –
тест **Эдуарда Люка** (1878) – **Дерика Генри Лемера** (1930).

Вспомогательное утверждение. Если m – простое число,
то существует натуральное число $b < m$
порядок которого по модулю m равен $(m – 1)$, т.е. $b^{m-1} \equiv 1 \pmod{m}$
и никакая меньшая степень числа b не равна $1 \pmod{m}$.

Обратное утверждение формулируется в виде следующей теоремы.

Теорема. Пусть m – целое число ≥ 2.
Если существует $b < m$, такое, что порядок b по модулю m
равен в точности $(m – 1)$,
то m – простое число.

Простые числа

Второй детерминированный тест – 2

Тест Люка – Лемера :
Число m просто тогда и только тогда,
когда существует элемент b, порядок которого по модулю m в точности равен $(m – 1)$.

<u>Эквивалентная формулировка :</u>
Число m просто тогда и только тогда,
когда существует b, $(b, m) = 1$ такое, что
$b^{m-1} \equiv 1 \pmod{m}$ и $b^{(m-1)/p}$ не равно $1 \pmod{m}$
для каждого простого делителя p числа $(m – 1)$.

Алгоритм тестирования :
Если b^{m-1} не равно $1 \pmod{m}$ для некоторого $b < m$,
то m не является простым (негативный тест, следует из теоремы Ферма).
С другой стороны, если для некоторого b порядок b равен $(m – 1)$,
то m – простое число (позитивный тест, следует из теоремы на предыдущем слайде).

Сложность алгоритма тестирования :
Очень сложно проверить, равен ли порядок b по модулю m числу $(m – 1)$,
т.к. нужно <u>для каждого</u> простого делителя p числа $(m – 1)$ показать,
что $b^{(m-1)/p}$ не равно $1 \pmod{m}$

Простые числа

Третий детерминированный тест

Замечание 1.
Этот тест был разработан Адлеманом, Померанцом и Рюмли в 1980 году,
а затем улучшен Коэном и Ленстрой в 1982 году.

Замечание 2.
Алгоритм тестирования использует специальную технику
алгебраической теории чисел, но принципиально близок к тесту Ферма.

Замечание 3.
На сегодняшний день это наиболее эффективный тест проверки целых чисел на простоту,
оценка временной сложности которого равна :

$$O\left(L(m)^{L\{L[L(m)]\}}\right)$$

Замечание 4.
Несмотря на то, что тест обладает экспоненциальной сложностью,
показатель степени стремится к бесконечности очень медленно.
Так, первое число, для которого показатель $L\{L[L(m)]\} = 2$, равно $H = 10^{999\,999\,999}$
Иными словами, проверка на простоту чисел $< H$ возможна за полиномиальное время.

Простые числа

Тесты псевдопростоты

Замечание. Вероятностные тесты простоты называют также тестами псевдопростоты, т.к. имея полиномиальную сложность, они не всегда способны дать однозначный ответ.

Под **тестом псевдопростоты** понимается тест, применяемый к паре целых чисел (b , m) и обладающий следующими свойствами :

(1) Тест может выдавать следующие ответы :
«m – составное число» или «не удалось определить»
(2) Если тест выдал ответ : «m – составное число» , то m – составное число
(3) Время выполнения теста полиномиально зависит от L (m)

Для **хорошего теста** псевдопростоты
существует фиксированное положительное вещественное число k такое, что
для любого составного целого числа m тест выдаёт ответ «составное»
по крайней мере для k * m выборов различных оснований b , где $1 \le b \le m$

Кроме того, будем говорить, что целое число m является
простым с большой вероятностью, если оно было подвергнуто хорошему тесту простоты и в результате был получен ответ «не удалось определить» для всех этих оснований b .

Простые числа

Первый вероятностный тест

Алгоритм тестирования :

Для заданного m выберем случайным образом b, $1 < b < m$

Если $b \mid m$, то тест выдаёт ответ «m – составное число»,

в противном случае – «не удалось определить».

Вероятность того, что выдаётся ответ «m – составное число», равна вероятности того, что $b \mid m$.

Если $d(m)$ – число делителей m

и b случайно выбрано в пределах $1 < b < m$,

то вероятность этого равна $p = (d(m) – 2) / m$.

Замечание. Очевидно, что это очень слабый тест.

Простые числа

Второй вероятностный тест

Алгоритм тестирования :

Для заданного m выберем случайным образом b , $1 < b < m$

Если $(b, m) \neq 1$, то тест выдаёт ответ «m – составное число»,

в противном случае – «не удалось определить».

Если m составное, то количество чисел $b < m$,

для которых тест выдаёт ответ «m – составное число»

равно $m - \phi(m)$, где $\phi(m)$ – это функция Эйлера.

Это количество велико,

если m имеет маленькие простые делители.

Однако, если $m = p * q$, где p, q – большие простые числа,

то доля хороших оснований очень мала.

Иными словами, этот тест не лучше предыдущего.

Простые числа

Третий вероятностный тест

Алгоритм тестирования :
Если для заданных чисел b , m
степень b^{m-1} не равна $1 \pmod{m}$,
то тест выдаёт ответ «m – составное число»,
в противном случае – «не удалось определить».

Замечание. Очевидно, что этот тест гораздо лучше двух предыдущих.
Однако он также несовершенен,
т.к. для всех псевдопростых по основанию b чисел
он выдаёт ответ «не удалось определить».

Замечание. Существует бесконечно много псевдопростых по основанию 2 чисел.
Такое утверждение следует из приведённой ниже теоремы.

Теорема. Если m – псевдопростое число по основанию 2,
то то же самое верно для числа $n = 2^m - 1$

Простые числа

Числа Кармайкла

Кроме псевдопростых чисел существуют составные числа, которые называются **абсолютными псевдопростыми числами** (или числами **Роберта Кармайкла**).

Числа Кармайкла определяются следующими условиями:

$m \mid (b^{m-1} - 1)$ для всех b таких, что $(b , m) = 1$

Наименьшее число Кармайкла – это число 561 :

561 = 3 * 11 * 17

Примеры других чисел Кармайкла :

1105 = 5 * 13 * 17

1729 = 7 * 13 * 19

Замечание. Для всех чисел Кармайкла третий вероятностный тест выдаёт ответ «не удалось определить». Однако следующий тест справляется с такими числами.

Простые числа

Четвёртый вероятностный тест

Алгоритм тестирования :

Пусть заданы b , m

Пусть $m - 1 = t * 2^s$, где t – нечётное число.

Рассмотрим числа $x_r \equiv b^{t * (2)^r} \pmod{m}$ для $0 \leq r < s$

(x_r – наименьший по абсолютной величине остаток по модулю m)

Если либо $x_0 = 1$, либо найдётся индекс i , $0 \leq i < s$ такой, что $x_i = -1$

то m называется **сильно псевдопростым** по основанию b

и тест выдаёт ответ «не удалось определить»,

в противном случае ответ – «m – составное число».

Замечание. Этот тест успешно применим и к псевдопростым числам.

Например, пусть $m = 561$. Тогда $m - 1 = 560 = 35 * 2^4$.

Далее, для $r = 4$ и $r = 3$ имеем соответственно :

$$2^{35 * (2)^4} = 2^{560} \equiv 1 \pmod{561} \qquad 2^{35 * (2)^3} = 2^{280} \equiv 1 \pmod{561}$$

Вместе с тем, для $r = 2$ имеем :

$$2^{35 * (2)^2} = 2^{140} \equiv 67 \pmod{561}$$

Следовательно, 561 – составное число.

Простые числа

Сложность четвёртого вероятностного теста

Обобщая вышеприведённые рассуждения имеем следующую теорему.

Теорема. Если тест сильной псевдопростоты выдаёт ответ «m – составное число», то m – составное число

Анализ временной сложности алгоритма четвёртого вероятного теста :
Число b^t вычисляется за время $O(L^3(m))$, т.к. :

(1) в алгоритме быстрого возведения в степень выполняется $O(L(t))$ умножений, $t \le m$
(2) длины умножаемых по модулю m чисел равны $\sim L(m)$
(3) каждое умножение выполняется за время $O(L^2(m))$

После этого при вычислении последовательности $b^{t*(2)^i}$, $i = 1, 2, \dots$
производится r возведений в квадрат, где также $r \le L(m)$,
и каждое возведение в квадрат выполняется за время $O(L^2(m))$.

Таким образом, тест выполняется за время $\mathbf{O(L^3(m))}$

Простые числа

Свойства четвёртого вероятностного теста

Замечание. Доказано (**Герберт Вилф**, 1986), что четвёртый вероятностный тест обладает следующим свойством :
если m – составное число, то вероятность того,
что тест выдаст ответ «m – составное число», не меньше $1/2$.

Основная идея доказательства состоит в том,
что собственная подгруппа конечной группы
не может содержать больше половины её элементов.

Замечание. Доказано (**Майкл Рабин**, 1980), что не существует
нечётного составного числа m , которое является сильно псевдопростым
по более, чем $1/4$ части всех оснований, меньших m .

Замечание. Практически, если мы применяем тест 100 раз,
используя 100 случайно и независимо выбранных оснований b_i , $0 \le b_i \le m$,
то тест определит, что m – составное с вероятностью $\ge 1 - 2^{-100}$,
причём каждая проверка будет выполнена за полиномиальное время.

План лекции: тема подраздела

- Алгоритм Евклида и цепные дроби
- Простые числа. Решето Эратосфена. Тесты простоты
- **Разложение целых чисел на множители**

Разложение целых чисел

Вводные замечания

Замечание 1. Задача нахождения делителей больших целых чисел существенно сложнее задачи проверки целых чисел на простоту.

Замечание 2. Неизвестно, существует ли вероятностный алгоритм, который за полиномиальное время
выдаёт делитель большого составного целого числа с вероятностью, большей, чем 1/2.

Замечание 3. В настоящее время
известно три метода разложения простых чисел на множители :

(1) Метод **Адриена Мари Лежандра**
(2) Метод Ферма
(3) Метод цепных дробей

Замечание 4. Метод Лежандра (1798) является наиболее сильным из известных методов разложения на множители целых чисел общего вида.

Разложение целых чисел

Метод Лежандра – 1

Метод основан на следующей идее.

Если

$u^2 \equiv v^2 \pmod{m}$,

$0 < u , v < m$,

$u \not\equiv \pm v \pmod{m}$

то

m делит $(u - v) * (u + v)$,

но не делит ни $(u - v)$, ни $(u + v)$.

Поэтому $(u - v, m)$ т.е. НОД чисел $u - v$ и m ,
является нетривиальным делителем m
и может быть легко вычислен с помощью алгоритма Евклида.

Замечание. Поиск таких чисел u и v производится в два этапа (см. ниже).

Разложение целых чисел

Метод Лежандра – 2

Пусть требуется разложить на множители число m .
Пусть $n = [\sqrt{m}]$ – максимальное число, не превосходящее $\sqrt{m}$.
Вычислим следующие числа :

$$a_k = (n + k)^2 - m$$

для небольших значений k (при этом числа k могут быть и отрицательными).

Пусть $\{ q_i , i = 1, 2, \dots, j \}$ – множество небольших простых чисел,
которые могут делить выражения вида $x^2 - m$
(т.е. число m является квадратом по модулю q_i)
Такое множество обычно называют **мультипликативной базой** B.

Запомним все числа a_k ,
которые могут быть разложены по мультипликативной базе B
(такие числа a_k называются B – числами), т.е. записаны в следующем виде :

$$a_k = (-1)^{\omega_{k0}} \prod_{1 \le i \le j} q_i^{\omega_{ki}}$$

Разложение целых чисел

Метод Лежандра – 3

С каждым B – числом a_k связывается следующий вектор показателей :

$e_k = (w_{k\,0} , w_{k\,1} , \dots , w_{k\,j})$, $w_{k\,i} \equiv \omega_{k\,i}$ (mod 2) , $i = 0 , 1 , \dots j$

Если мы найдём достаточно B – чисел,
чтобы множество соответствующих векторов показателей
было линейно зависимо по модулю m
(любое множество из ($j + 2$) B – чисел обладает этим свойством),
то можно будет представить нулевой вектор
в виде суммы векторов показателей некоторого множества S
(например, в виде $\sum_{k\,:\,a_k \in S} e\, k \equiv (0 , 0 , \dots , 0)$ (mod 2))

Определим теперь целые числа u и v (см. следующий слайд).

Замечание. Из сказанного выше следует, что $u^2 \equiv v^2$ (mod m) и ($u - v , m$) может быть нетривиальным делителем m .

Разложение целых чисел

Метод Лежандра – 4

$$e'_i = \frac{1}{2} \sum_{k : a_k \in S} \omega_{k\,i} \quad , \quad i = 0, 1, \dots, j$$

$$u = \prod_{k \in S} (n + k) \pmod m$$

$$v = \prod_{1 \le i \le j} q_i^{e'_i} \pmod m$$

Разложение целых чисел

Пример применения метода Лежандра – 1

Пример. Разложим на множители число 1729 (это третье число Кармайкла).

В этом случае $m = 1729$, $n = [\sqrt{1729}] = 41$

Вычислим числа :

$a_k = (n + k)^2 - m$ для небольших значений k .

Имеем :

$a_1 = 35$, $a_2 = 120$, $a_3 = 207$, $a_4 = 296$, $a_5 = 387$,

$a_6 = 480$, $a_7 = 575$, $a_8 = 672$, $a_9 = 771$ и т.д.

Зафиксируем множество небольших простых чисел $\{ 2 , 3 , 5 , 7 \}$

Очевидно, что только числа : 35 , 120 , 480 , 672 являются B – числами :

$35 = (-1)^0 * 2^0 * 3^0 * 5^1 * 7^1$

$120 = (-1)^0 * 2^3 * 3^1 * 5^1 * 7^0$

$480 = (-1)^0 * 2^5 * 3^1 * 5^1 * 7^1$

$672 = (-1)^0 * 2^5 * 3^1 * 5^0 * 7^1$

Все указанные B – числа, кроме числа 480, включаются в множество S, т.к. сумма векторов показателей этих чисел равна (0 , 0 , 0 , 0 , 0) .

Разложение целых чисел

Пример применения метода Лежандра – 2

Далее вычисляем показатели e'_i и числа u и v :

$e'_0 = 0$
$e'_1 = 8 / 2 = 4$
$e'_2 = 2 / 2 = 1$
$e'_3 = 2 / 2 = 1$
$e'_4 = 2 / 2 = 1$

$u = (41 + 1) * (41 + 2) * (41 + 8) \equiv 315 \pmod{1729}$

где $(41 + 1)^2 \equiv (a_1 = 35) \pmod{1729}$
$(41 + 2)^2 \equiv (a_2 = 120) \pmod{1729}$
$(41 + 8)^2 \equiv (a_8 = 672) \pmod{1729}$

$v = (-1)^0 * 2^4 * 3^1 * 5^1 * 7^1 = 1680$

Квадраты чисел u и v совпадают по модулю 1729 :
$u^2 = 99225 \equiv 672 \pmod{1729}$ $v^2 = 2822400 \equiv 672 \pmod{1729}$

Вычисляем НОД $(u - v, m) = (315 - 1680 , 1729) = (-1365 , 1729) = 91$
Таким образом, число 91 является делителем числа 1729.

Разложение целых чисел

Сложность метода Лежандра

Замечание. Доказано (**Джон Диксон**, 1981), что число m разложимо на множители методом Лежандра за время равное :

$$O\left(e^{(\alpha + o(1)) * \sqrt{(\ln(m) * \ln(\ln(m)))}}\right)$$

где α – некоторая константа, $o(1) \to 0$ при $m \to \infty$

Эта величина растёт медленнее, чем экспонента, но быстрее, чем любая степень числа $L(m)$.

Разложение целых чисел

Числа Ферма и Мерсенна

Числа следующего вида :

$F_m = 2^{2^m} + 1$

$M_m = 2^m - 1$

называются соответственно числами Ферма и числами **Марена Мерсенна**.

Эти числа оказали существенное влияние
на развитие алгоритмов разложения на множители
и проверки простоты целых чисел.

Замечание. Ферма предположил (1640), что любое число F_m , $m \geq 1$
является простым.

Однако Эйлер доказал (1729), что число F_5 не является простым,
т.к. оно делится на число 641.

Разложение целых чисел

Метод Ферма

Если m – положительное нечётное число,
то существует взаимно однозначное соответствие
между разложениями m на множители вида $m = p * q$, $p \geq q > 0$
и представлениями m в виде $m = u^2 - v^2$,
где u , v – неотрицательные целые числа.

Это соответствие задаётся следующими уравнениями :

$$u = (p + q) / 2 \quad v = (p - q) / 2$$
$$p = u + v \qquad q = u - v$$

Используя вышеуказанное соответствие, можно разложить число m
на множители, вычисляя для $k = 1 , 2 , \ldots$ числа $a_k = [\sqrt{m}] + k$
и проверяя, является ли $u^2 - n = v^2$ полным квадратом
(разумеется полагая при этом, что $u = a_k$).

Замечание. Метод Ферма является обоснованием метода Лежандра.

Разложение целых чисел

Метод цепных дробей

Замечание. Этот метод отличается от метода Лежандра только тем, что величины a_k выбираются с применением цепных дробей (здесь также используются системы множителей).

Метод цепных дробей основан на следующих свойствах.

(1) Если $x > 1$ – вещественное число, разложение которого в цепную дробь имеет подходящие дроби p_i / q_i, то $|(p_i)^2 - x^2 * (q_i)^2| < 2 * x$ для всех i.

(2) Если m – положительное целое число, не являющееся полным квадратом, и p_i / q_i – подходящие дроби в разложении $\sqrt{m}$ в цепную дробь, то наименьший по абсолютной величине вычет $p^2 \pmod{m}$ меньше, чем $2 * \sqrt{m}$.

Замечание. Свойство (2) является ключевым в методе цепных дробей. Согласно ему, перебирая <u>числители</u> подходящих дробей в разложении $\sqrt{m}$ в цепную дробь можно найти последовательность чисел p_i, квадраты которых имеют малые вычеты. (Ещё раз подчеркнём, что не нужно находить точное значение подходящей дроби, достаточно определить только её числитель p_i).

Спасибо за внимание !

Вопросы ?