Защита информации

Павел Юдаев

МГТУ им. Баумана, Кафедра ИУ-9

Москва, 2014

Раздел 12 - Теория чисел

Замеч.: $\forall n$, длина его записи - это $log\ n$. Поэтому полиномиальный алгоритм от длины входа - $poly(log\ n)$. Если вход - число n, то $O(n)$ - это экспоненциальная сложность от длины входа.

Замеч.: В группе порядка n ее элементы могут быть отождествлены с двоичными числами от 0 (или 1) до $n-1$ или n длиной не более $l = log_2(n)$. Групповая операция обрабатывает массивы длины l.

Утверждение

В любой группе G операция возведения элемента в степень k (k - натуральное число) имеет полиномиальную сложность от $log\ k$.

Док-во

Самостоятельно.
$\forall k = 2^n\ a^k$ вычисляется за n операций.

Пример

$22_{10} = 10110_2$. $a^{22} = (...(a^1)^2 a^0)^2 a^1)^2 a^1)^2 a^0$

Обобщенный алгоритм Евклида - НОД

Даны $n \geq m \in \mathbb{N}$. Найти n', m': $nn' + mm' = \text{НОД}(n, m)$.

2014 - без док-ва из-за нехватки времени. Этот алгоритм должен был быть на младших курсах.

1. Условие остановки. $\text{НОД}(n, 0) = n$
   $n \cdot 1 + 0 \cdot 0 = n$
   $n' := 1,\ m' := 0$
2. Шаг рекурсии (вниз). $n_i n_i' + m_i m_i' = \text{НОД}(n, m)$.
   $n_{i+1} := m_i$
   $m_{i+1} := n_i - [n_i / m_i] m_i = n_i \bmod m_i \qquad (*)$
   и снова верно
   $n_{i+1} n_{i+1}' + m_{i+1} m_{i+1}' = \text{НОД}(n, m)$.
   Шаг рекурсии сделан.

3. Вычисление n'_i, m'_i (вверх).
   В уравнение
   $n_{i+1}n'_{i+1} + m_{i+1}m'_{i+1} = \text{НОД}(n, m)$
   подставим (*):

   $$m_i n'_{i+1} + (n_i - [n_i/m_i]m_i)m'_{i+1} = \text{НОД}(n, m),$$
   перегруппируем:

   $$n_i \cdot m'_{i+1} + m_i \cdot (n'_{i+1} - [n_i/m_i]m'_{i+1}) = \text{НОД}(n, m).$$
   $$n_i \cdot n'_i + m_i \cdot m'_i = \text{НОД}(n, m)$$

   Значит
   $n'_i := m'_{i+1}, \;\; m'_i := n'_{i+1} - [n_i/m_i]m'_{i+1}.$

Обозн.: *ExtGCD*

Пример

Найти $9^{-1} mod 160$. $ExtGCD(160, 9)$:

$160 - [160/9] \cdot 9 = \mathbf{7}$

i	n	m	$[n/m]$	n'	m'
1	160	9	17	4	$-3 - [160/9] \cdot 4 = -71$
2	9	**7**	1	-3	4
3	7	2	3	1	-3
4	2	1	1	0	$\underline{1} - [2/1] \cdot \underline{0} = 1$
5	1	0		$\underline{1}$	$\underline{0}$

$\Rightarrow 9^{-1} mod 160 = -71 = 89 mod 160$

Утверждение

Расширенный алгоритм Евклида имеет полиномиальную сложность.

Док-во

Арифметические операции + - * и деление с остатком имеют полиномиальную сложность.
Докажем, что число шагов НОД(n_0, m_0), $n_0 \geq m_0$ равно $O(log\ m_0)$.

Шаги рекурсии:
$ExtGCD(n_0, m_0) \rightarrow extGCD(n_1, m_1) \rightarrow .. \rightarrow GCD(n_k, 0)$.
Докажем, что $m_{i+2} \leq m_i/2$.

Док-во (Продолжение)

Из описания алгоритма $m_{i+1} < m_i \ \forall i$. Если $m_{i+1} \leq m_i/2$, то доказано.
Пусть $m_i > m_{i+1} \geq m_i/2$ (**). След. рекурсивный вызов:

$$ExtGCD(n_{i+1} = m_i,\ m_{i+1})$$

и на след. за ним шаге
$m_{i+2} = m_i \ mod \ m_{i+1} =^{(**)} m_i - m_{i+1} <^{(**)} m_i/2$.

След-но, число рекурсивных вызовов не более $2log(m_0)$.
Ч.т.д.

Утверждение

В группе $\mathbb{Z}_m^* \; \forall a \in \mathbb{Z}_n^*$ элемент a^{-1} может быть найден за время, полиномиальное от $n = |\{\mathbb{Z}_m^*\}|$.

Док-во

$aa' + mm' = 1 \Rightarrow aa' = 1 \; mod \; m$.

$ExtGCD$: полин. число шагов, каждый шаг - фикс. число опер., каждая из кот. им. полин. сложность.

Опр.

Функция Эйлера Пусть $n \in \mathbb{N}$. $\varphi(n) := |\mathbb{Z}_n^*|$.

Свойства:

- Если p - простое, $\varphi(p) = p - 1$.
- Пусть $p \neq q$ - простые. Легко видеть, что $\varphi(pq) = pq - p - q + 1 = (p-1)(q-1)$.

- Задача

Пусть НОД$(n, m) = 1$. Тогда $\varphi(nm) = \varphi(n) \cdot \varphi(m)$.

- Задача

Пусть $n = \prod\limits_i p_i^{e_i}$ - разложение на простые множители. Тогда $\varphi(n) = \prod\limits_i p_i^{e_i - 1}(p_i - 1)$. - выч-ся полиномиальным алгоритмом.

Теорема 1 (Эйлера)

$\forall x \in \mathbb{Z}_N^* \ \ x^{\varphi(N)} = 1$

Док-во

Как следствие малой т. Ферма.

Раздел 12 - Теория чисел

Корни из элементов в $\mathbb{Z}_m^*$

Пусть p - простое. $\mathbb{Z}_p$ - поле.

Линейное ур. $ax + b = 0$ всегда им. реш. в $\mathbb{Z}_p$: $x = -ba^{-1}$.
Можем ли решить $x^k - c = 0, \;\; k \geq 2$ в $\mathbb{Z}_p$?

Опр.

$x \in \mathbb{Z}_p : x^k = c$ наз. *корнем k-й степени из c в $\mathbb{Z}_p$*.

Пример

$\mathbb{Z}_{11}^*$:
$7^{1/3} = 6$
$3^{1/2} = 5$
$2^{1/2} \nexists$

В общем случае, полин. алг. вычисления $c^{1/a}$ в $\mathbb{Z}_n^*$ использует разложение n на множители.

Квадратичные вычеты

Лемма

Уравнение $x^2 \equiv 1 \bmod n$ имеет ровно 2 корня $\forall n$.

Док-во

1, -1 - корни. Степень ур-я равна 2, поэтому др. корней нет.

Если p - нечетное простое, то НОД$(2, p-1) \neq 1$.

Поэтому в $\mathbb{Z}_p$ отобр. $x \leftrightarrow x^2$ не явл. биекцией.
Это ф-ция 2 в 1: $x \to x^2$, $-x = p - x \to x^2$.

Опр.

эл-т x в $\mathbb{Z}_p^*$ наз. *квадратичным вычетом*, если $\exists\, x^{1/2}$.

Задача

Если p - нечетное простое, то число квадр. выч. в $\mathbb{Z}_p^*$ равно $(p-1)/2$, ровно половина элементов.

Пример

Отображение $x \to x^2$ в $\mathbb{Z}_{11}^*$.

Символ Лежандра

Пусть a - целое число, и p - простое нечетное число.

Опр.
$\left(\frac{a}{p}\right)$ - *символ Лежандра.* $\left(\frac{a}{p}\right) := a^{(p-1)/2} \in \{0, 1, -1\}$ в $\mathbb{Z}_p$.

Свойства:

- $\left(\frac{a}{p}\right) = 0$, если $a = kp,\ k \in N$
- $\left(\frac{a}{p}\right) = 1$, если a - квадр. вычет в $\mathbb{Z}_p^*$, т.е. $a = x^2,\ x \neq 0$
- $\left(\frac{a}{p}\right) = -1$, если a - не явл. квадр. вычетом в $\mathbb{Z}_p^*$.
- это коммутативная функция: $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$

Док-во (Все свойства)

$\forall a \in \mathbb{Z}_p^*$

$a^{p-1} - 1 = 0$

$(a^{(p-1)/2} - 1)(a^{(p-1)/2} + 1) = 0$

Если a - кв.в., то $\exists b \in \mathbb{Z}_p^* : a = b^2$, $a^{(p-1)/2} = b^{p-1} = 1$.
Ровно $(p-1)/2$ кв.в. - корни первого сомножителя.
Значит, все кв. невычеты - корни второй скобки.

Коммутативность для произведения двух невычетов следует из того, что числа - корни уравнения (см. выше), для остальных - очевидно.
Ч.т.д.

Вычисление квадратного корня в $\mathbb{Z}_p^*$, $p = 3 \; mod \; 4$, p - простое.

Лемма

Если $c \in \mathbb{Z}_p^*$ - квадр. вычет, то $c^{1/2} = c^{(p+1)/4}$

Док-во

$(c^{(p+1)/4})^2 = c^{(p+1)/2} = c^{(p-1)/2} \cdot c = 1 \cdot c = c$
Использовали факт, что $(p+1)/4$ - целое. Ч.т.д.

Замеч.: если p - простое, $p = 1 \; mod \; 4$, также сущ. полиномиальный алгоритм выч. квадр. корня.

Раздел 12 - Теория чисел

2014 - без нее из-за нехватки времени?

Утверждение

Пусть G, H - группы. Определим прямое произведение групп $Q = G \times H$:
- элементы Q - все упорядоченные пары эл-тов (g, h)
- групповая операция: $q_1, q_2 \in Q$, $q_1 \circ_Q q_2 = (g_1 \circ_G g_2, h_1 \circ_H h_2)$

Тогда $Q = G \times H$ - группа.

Док-во

Проверить аксиомы группы. Самостоятельно.

Очев., это обобщается на любое конечное кол-во групп.

100 г. до н.э., Sun-Tsu: Найти число, дающее при делении на 3, 5, 7 остатки 2, 3, 2 соотв.

Теорема 2 (Китайская теорема об остатках)

Если числа $a_1, ...a_n$ попарно взаимно просты,
то для любых остатков $r_1, ...r_n : 0 \leq r_i < a_i$
найдется число $m : m = r_i \bmod a_i \; \forall i = 1, ...n$

Лит-ра: Introduction to Modern Cryptography, J.Katz, Y.Lindell. 2007. Pp. 241-244-248.

Теорема 3 (Китайская теорема об остатках)

Пусть $n = pq$, НОД$(p, q) = 1$. Тогда $Z_n \simeq Z_p \times Z_q$ и $Z_n^* \simeq Z_p^* \times Z_q^*$.
Кроме того, пусть $f : Z_n \to Z_p \times Z_q$, $f(x) = (x \; mod \; p, \; x \; mod \; q)$.
Тогда f - изоморфизм $Z_n \to Z_p \times Z_q$ и изоморфизм $Z_n^* \to Z_p^* \times Z_q^*$.

Док-во

Очевидно, что f действ. из Z_n в $Z_p \times Z_q$ и образ x единственный. Докажем, что f - биекция.

1) Инъективность. Док. для Z_n^*. От противного.
Пусть $x \in Z_n^*,\ f(x) = (x_p, x_q)$. Пусть $x_p \notin Z_p^*$. Тогда НОД$(x\ mod\ p,\ p) \neq 1$.
След-но, НОД$(x, p) \neq 1$. Против. с тем, что $x = (pq) \cdot k + 1$.
След-но, $(x_p, x_q) \in Z_p^* \times Z_q^*$.

Док-во (Продолжение)

2) Сюръективность. Док. для Z_n и Z_n^*.
$f(x)$ опр. однозначно. Пусть
$x \neq x' \in Z_n,\ f(x) = (x_p, x_q) = f(x')$.
Тогда $x = x_p = x' \bmod p$ и $x = x_q = x' \bmod q$.

След-но $(x - x')$ делится на p и на q. НОД$(p, q) = 1,\ pq = n$ след-но $(x - x')$ делится на n. Это невозм., т.к. $0 < |x - x'| < n$.

След-но f - отображение 1 в 1. Множества Z_n, $Z_p \times Z_q$ конечные, равной мощности. Поэтому f - биекция.

Док-во (Продолжение)

3) f сохраняет операцию. Док. для Z_n.
По опр., $(a_p,\ a_q) \circ (b_p, b_q) = ((a_p +_p b_p),\ (a_q +_q b_q))$.

По опр. функции f имеем
$f(a +_n b) = ((a +_n b)\ mod\ p,\ (a +_n b)\ mod\ q) =$
$\{$т.к. $n = pq$, то $c \bmod n \bmod p = c \bmod p$ $\}$

$= ((a +_p b),\ (a +_q b)) = (a\, mod\, p, a\, mod\, q) \circ (b\, mod\, p, b\, mod\, q) =$
$= f(a) \circ f(b)$.

Сохранение операции для групп по умножению док-ся аналогично.
Ч.т.д.

Для $n = p_1 \cdot \ldots \cdot p_t$ утверждение доказывается, применив эту теорему $(t-1)$ раз.

Отсюда следует, что переход $x \to (x_p, x_q)$ - биекция и сохраняет операцию.

Как сделать обратный переход, от вектора остатков к числу?

$x \leftrightarrow (x_p, x_q) = x_p \cdot (1, 0) \circ x_q \cdot (0, 1)$

Т.е. найдутся такие эл-ты в $\mathbb{Z}_n$, что $(1, 0) \leftrightarrow e_p \in \mathbb{Z}_n$, $(0, 1) \leftrightarrow e_q \in \mathbb{Z}_n$.

$(x_p, x_q) \leftrightarrow x_p \cdot e_p +_n x_q \cdot e_q = x \in \mathbb{Z}_n$.

Как найти $e_p \in \mathbb{Z}_n$? По обобщ. алг. Евклида, сущ.

$p', q' : p'p + q'q = 1$. Тогда $e_p = q'q$ (очевидно из этого равенства), $e_q = p'p$

Т.е. нашли алгоритм обратного переход от пары (x_p, x_q) к числу x.

Для $n = p_1 \cdot \ldots \cdot p_t$ применить обобщ. алг. Евклида $(t-1)$ раз. Первый раз - для $p_1 \cdot \ldots \cdot p_{t-1}$ и p_t, и т.д., отщепляя по одному числу.

Закончили обоснование операций с большими числами с использованием теоремы об остатках.

Применение: $a_1, ..., a_n$: НОД$(a_i, a_j) = 1 \ \forall i \neq j$.
Система ур-й
$x = r_i \bmod a_i$, $i = 1..n$
всегда имеет единственное решение x, меньшее чем
$m = a_1 * .. * a_n$

Операции с большими числами:

Числа $\mapsto$ векторы остатков.
Операции над векторами остатков - по модулям a_i.
Результат $\mapsto$ число.

Пример

$n = 9 * 20$, вычислить $7 * 11$.
Имитируем работу с большими числами через векторы остатков. Важно, что $(7 * 11) < n$, иначе получили бы рез-т по модулю n, или надо было бы добавлять третье число к набору вз. простых чисел $(9, 20)$.
$7 * 11 \mapsto (7, 7) * (2, 11) = (7 * 2 \bmod 9, 7 * 11 \bmod 20) = (5, 17)$.
$ExtGCD \Rightarrow -11 * 9 + 5 * 20 = 1$
$(1, 0) \leftrightarrow 5 * 20 = 100 \bmod n, \ (0, 1) \mapsto -99 = 81 \bmod n$
$7 * 11 \leftrightarrow 5 * 100 + 17 * 81 = 140 + 117 = 77 \bmod n$.

В реальных библиотеках для работы с большими числами n равно произведению некоторого количества больших простых чисел, каждое из которых укладывается в простой тип данных.

Раздел 12 - Теория чисел

Поиск генератора в конечном поле.

$GF^*(q)$, случайное $a \in GF^*(q)$, $a \neq 0, 1$.

- прямой перебор степеней a.
- пусть $q - 1 = p_1^{i_1} * .. * p_k^{i_k}$ - разложение на простые множители. Вычислим $b_j = a^{(q-1)/p_j}$, $j = 1, ..., k$.
  Если $\exists b_j = 1$, то a - не генератор.
  Иначе a - генератор $GF^*(q)$.

Задача

обосновать второй способ.

Пример

$GF(13)$. $12 = 2 \cdot 2 \cdot 3$.
$2^4 \equiv 3 \bmod 13$; $2^6 \equiv 12 \bmod 13$. 2 - генератор $GF^*(13)$
$5^4 \equiv 1 \bmod 13$. 5 - не генератор $GF^*(13)$

Проверка чисел на простоту

1) Простейший - перебор чисел, выполнение деления

2) Тест на основе малой т. Ферма
Если $\exists a : 1 \leq a < n : \ a^{n-1} \neq 1 \ mod \ n$, то n - составное.

Есть псевдопростые числа!

Тест на основе малой т. Ферма и алгоритма Евклида

- выбираем случайное a из $\{1, .., n-1\}$ и проверяем условие НОД$(a, n) = 1$
- если не выполнено, n - составное.
- проверяем $a^{n-1} \equiv 1 \mod n$
- если не выполнено, n - составное. Назовем число a свидетелем этого.
- иначе ответ не известен, но можно повторить тест еще раз

2014 - без этого утверждения.

Утверждение

(*) Если n составное, и в $\mathbb{Z}_n^*$ сущ. свидетель того, что n - составное, то кол-во свидетелей в $\mathbb{Z}_n^*$ не меньше $|\mathbb{Z}_n^*|/2$.

Док-во

(*) пусть $B \subseteq \mathbb{Z}_n^*$, все элементы B - не свидетели простоты n.
Заметим, что $1^{n-1} = 1 \ mod \ n$, т.е. $1 \in B$.
Если $a, b \in B$, то $a^{n-1} * b^{n-1} = 1 * 1 = 1$ т.е. $a * b \in B$.
Значит B - подгруппа $\mathbb{Z}_n^*$ и $|B| \leq |\mathbb{Z}_n^*|/2$.

(*)
Пусть в $\mathbb{Z}_n^*$ сущ. свидетели того, что n - составное.
Оценим вер-ть, что на одном шаге теста мы выберем
либо свидетеля того, что n - составное,
либо число не взаимно простое с n.

$$p \geq \frac{|\mathbb{Z}_n^*|/2 + \left((n-1) - |\mathbb{Z}_n^*|\right)}{n-1} = 1 - \frac{|\mathbb{Z}_n^*|/2}{n-1} \geq 1 - \frac{|\mathbb{Z}_n^*|/2}{|\mathbb{Z}_n^*|} = \frac{1}{2}$$

Повторив алгоритм t раз, вер-ть ложно положительного теста на простоту будет не более 2^{-t}.

Опр.

Числа Кармайкла - это составные числа n:
$b^{n-1} = 1\ mod\ n\ \forall b : \text{НОД}(b, n) = 1$

Т.е. нет свидетелей того, что они составные.
Первые числа К.: 561, 1105, 1729, 2465, 2821, ...
Они "плохие" для этого алгоритма.
Среди первых 10^8 чисел их 255. Всего их беск. много.

Поэтому надо улучшить алг-м.

Вероятностный тест Миллера-Рабина.

Факт: пусть p - простое. Тогда ур-е $x^2 = 1 mod\ p$ имеет только тривиальные корни 1, -1.

Пусть n - простое нечетное и $s, t: \ n - 1 = t2^s$, где t - нечетное.

$a^{n-1} = 1\ mod\ n\ \forall a \in \mathbb{Z}_n^*$. Поэтому, извлекая из a^{n-1} кв. корни, мы на каждом шаге обязательно получим одно из двух: 1, -1.

Если на нек. шаге получили -1, то для нек. $0 \leq q \leq s - 1$ $a^{t2^q} = -1 mod\ n$.

Если ни разу не получили -1, в т.ч. $a^t \neq -1 mod\ n$, то $a^t = 1 mod\ n$.

Пусть n - простое и $s, t: \ n-1 = t2^s$, где t - нечетное.
Пусть $a \in \mathbb{Z}_n^*$. Тогда

- или $a^t = 1 \ mod \ n$ (*)
- или в наборе чисел $a^t, a^{2t}, ..a^{t(2^{s-1})}$ найдется число, равное $-1 \ mod \ n$ (**).

Это необходимое условие.

Опр.

Пусть n - составное. $b < n$ - *лжесвидетель* того, что n - простое, если вып. (*) и (**).

Теорема 4 (Рабина)

Пусть n - нечетное составное. Тогда количество лжесвидетелей того, что n - простое, не более $\varphi(n)/4$.

Без док-ва.
См.
`http://www.uic.unn.ru/~zny/compalg/Lectures/rabin.pdf`

Тест Миллера-Рабина на простоту числа.

Представим $n-1$ в виде $n-1 = t2^s$, где t - нечетное.

- выбираем случ. a из 1, ..., $n-1$, проверяем НОД$(a, n) =? 1$.
- если не выполняется, n - составное.
- выч. $a^t \ mod \ n$.
- если $a^t = 1$ или $a^t = -1 \ mod \ n$, то n м.б. простым. Повторить тест.
- выч. $a^{2t}, ..a^{(2^{s-1})t}$ пока не появится -1.
- если нашлось -1, то n м.б. простым. Повторить тест.
- иначе, т.е. ни одно из этих чисел не равно -1. Ответ - составное.

Детерминированный полиномиальный критерий простоты

$\exists$ несколько детерминированных алгоритмов - тестов простоты. Самый быстрый из них - полиномиальный.

Т.е. алгоритм полиномиален относительно длины входа - числа бит записи n. Всегда дает точный ответ, не использует не доказанную гипотезу.

(*) Это алгоритм AKS, 2002.
Agrawal, Kayal, Saxena, "PRIME is in P".

(*) Описание и обоснование алгоритма - в статье [3].

Поиск простых чисел.

Для ряда криптосистем с открытым ключом нужны большие простые числа.

Выбирается случайное число, и проверяется по одному из вероятностных тестов. Например:

1. выберите n-битовое число p
2. установите старший и младший биты в 1
3. выполните тест Рабина-Миллера пять раз для разных (случайных?) a

Если p не проходит хотя бы одну из проверок, p - составное. Оценить сверху вероятность того, что составное число будет признано простым.

Maple: `isprime()` - детерминир., тест Миллера-Рабина для $a = 2, 3, 5, 7, 11$.

Дискретные логарифмы в конечном поле
Найти $x : a^x = b \; mod \; n$

Если $F = \mathbb{Z}_p$ и p - простое,
то сложность выч. дискр. логарифма - того же порядка, что сложность задачи факторизации числа $n \approx p$, при этом $n = st$, s, t - простые примерно одного порядка.

Известны субэкспоненциальные алгоритмы факторизации.

Субэкспоненциальная сложность:
$\lim\limits_{x\to\infty} f(x)x^{-n} = \infty \; \forall n$, и
$\lim\limits_{x\to\infty} (log\ f(x))/x = 0$
Напр., 2^{n^α}, $0 < \alpha < 1$.

Замечание: если p - простое, но $p - 1$ имеет *только* малые простые множители, то есть быстрый алгоритм факторизации числа $p - 1$. Поэтому выбираются p: у $p - 1$ есть хотя бы один большой простой множитель.

Литература к лекции

1. Обоснование теста Миллера-Рабина
http://www.uic.unn.ru/~zny/compalg/Lectures/rabin.pdf

2. V. Shoup, *A Computational Introduction to Number Theory and Algebra*, 2008

3. J.Katz, Y.Lindell. *Introduction to Modern Cryptography*, 2007. Pp. 241-248.

4*. Описание тестов простоты: Rene Schoof (2004), “Four primality testing algorithms”, Algorithmic Number Theory: Lattices, Number Fields, Curves and Cryptography, Cambridge University Press, ISBN 0-521-80854-5