«Алгоритмы дискретного логарифмирования. Алгоритм согласования, алгоритм Полига-Хеллмана»

Подготовила: Гусева Екатерина (6373)

Задача дискретного логарифмирования

Пусть G – мультипликативная абелева группа, $a, b \ \in \ G$. Тогда задача нахождения решения уравнения

$$a^x = b$$

Называется задачей дискретного логарифмирования в группе G. Её решение x называется дискретным логарифмом элемента b по основанию a, если основание а фиксировано и если существует $\log_a b \in Z/|G|Z$, если $|\mathrm{G}| < \infty$.

Задача дискретного логарифмирования

Рассмотрим уравнение

$$a^x \equiv b (mod\ p) \quad (1)$$

В группе $(Z/pZ)^*$где p – простое число. Будем предполагать, что порядок $a(mod\ p)$ равен $p-1$. Тогда уравнение разрешимо, и решение x является элементом $Z/(p-1)Z$.

С помощью перебора уравнение (1) можно решить за $O(p)$

Арифметических операций. Но можно ли придумать более эффективный алгоритм?

Алгоритм согласования

(Алгоритм Гельфонда — Шенкса)

Теория

Идея алгоритма состоит в выборе оптимального соотношения времени и памяти, а именно в усовершенствованном поиске показателя степени.

$$a^x \equiv b(mod\ p)$$

Алгоритм поиска x основан на представлении x в виде $Hu - v\ mod\ (p-1)$, где $H = \left|\sqrt{p}\right| + 1$ и переборе $1 \ll u \ll H,\ 0 \ll v \ll H$.

Данный алгоритм имеет сложность $O(p^{1/2}\log p)$

Алгоритм

Шаг 1. $H = \left|\sqrt{p}\right| + 1$

Шаг 2. Найти $c \equiv a^{H} (mod\ p)$

Шаг 3. Составить таблицу значений $c^{u} (mod\ p)$, $1 \ll u \ll H$, упорядочить её

Шаг 4. Составить аналогичную таблицу $b * a^{v} (mod\ p)$,

$0 \ll v \ll H$, упорядочить

Шаг 5. Найти совпадающие элементы для 1 и 2 таблиц. Для них

$$c^{u} \equiv b * a^{v} (mod\ p)$$

Из шага 2 и нехитрых математических преобразований прямо следует, что $a^{Hu-v} \equiv b\ (mod\ p)$

Значит, **мы нашли** $\boldsymbol{x \equiv Hu - v\ mod(p-1)}$

Доказательство корректности алгоритма

Любое число x, $0 \ll x \ll p - 2$ можно представить в виде $\mathrm{x} \equiv \mathrm{Hu} - \mathrm{v} \bmod (\mathrm{p} - 1)$, где $1 \ll \mathrm{u} \ll \mathrm{H}$, $0 \ll v \ll H$. Действительно, набор чисел $H, H-1, H-2, \dots, H - H, 2H, 2H-1, \dots, 2H-H, \dots, H^2, H^2-1, \dots, H^2-H$ содержит в себе набор чисел $0,1,\dots,p-2$, поскольку H^2 $> p$. Из этого следует корректность алгоритма.

Примеры

Пусть $5^x \equiv 3 (mod\ 23)$

Тогда $H\ = \left|\sqrt{23}\right| + 1 = 5$, $c \equiv a^H (mod\ p) \equiv 5^5 (mod\ 23) = 20$

Составляем таблицу для $c^u (mod\ p)$, $1 \ll u\ \ll H$:

u	1	2	3	4	5
5^u	20	9	19	**12**	10

Составляем таблицу для $b * a^v (mod\ p)$, $0 \ll\ v \ll H$:

v	0	1	2	3	4	5
$3 * 5^V$	3	15	6	7	**12**	75

$$x = Hu\ - v = 5 * 4 - 4 = 16\ (mod\ 23)$$

$$5^{16} \equiv 3\ (mod\ 23)$$

Усовершенствование производительности алгоритма

Реализация

Существует способ улучшить производительность алгоритма Гельфонда — Шенкса. Он заключается в использовании эффективной схемы доступа к таблице. Лучший способ — использование хеш-таблицы. Следует производить хеширование по второй компоненте, а затем выполнять поиск по хешу в таблице. Так как доступ и добавление элементов в хеш-таблицу работает за время $\mathrm{O}(1)$(константа), то асимптотически это не замедляет алгоритм.

Время работы алгоритма оценивается как $O(\sqrt{n})$, что намного лучше, чем время работы полного перебора показателей степени

Алгоритм Полига - Хеллмана

Пусть задано уравнение

$$a^x \equiv b(mod\, p)$$

И известно разложение числа p-1 на простые множители:

$$p - 1 = \prod_{i=1}^{i=k} {q_i}^{\alpha_i}$$

Идея алгоритма

Суть алгоритма в том, что достаточно найти x по модулям ${q_i}^{\alpha_i}$ для всех i, а затем решение исходного сравнения можно найти с помощью китайской теоремы об остатках.

Чтобы найти x по каждому из таких модулей, нужно решить сравнение

$$(a^x)^{\frac{(p-1)}{{q_i}^{\alpha_i}}} \equiv b^{\frac{(p-1)}{{q_i}^{\alpha_i}}} (mod\, p)$$

Упрощенный вариант описания

Лучший путь, чтобы разобраться с алгоритмом – рассмотреть крайний случай, когда p **раскладывается на** $\mathbf{2^n+1}$

Учитываем, что, по определению, a имеет степень $p-1$, следовательно:

$$a^{(p-1)} \equiv 1 (mod\ p)\ (1)$$

Когда $p\ =\ 2^n+1$, то легко определить x через двоичное разложение с коэффициентами $\{q_0, q_1 \ldots q_{n-1}\}$, например:

$$x = \sum_{i=1}^{i=n-1} q_i 2^i = q_0 +\ q_1 * 2^1 + \cdots + q_{n-1} * 2^{n-1}$$

Следствие из (1):

$$a^{(p-1)/2} \equiv \pm 1\ (mod\ p)$$

Но $a^{(p-1)/2}$ по определению принимает значение, отличное от 1, значит, остаётся одно сравнение:

$$\boldsymbol{a^{(p-1)/2} \equiv -1\ (mod\ p)}\ (2)$$

Упрощенный вариант описания

Теперь возведём $a^x \equiv b(mod\ p)$ в степень $\frac{p-1}{2}$:

$$(a^x)^{(p-1)/2} \equiv b^{\frac{p-1}{2}}(mod\ p)$$

Из выкладки (2) следует:

$$(-1)^x \equiv b^{(p-1)/2}(mod\ p)$$

Равенство $(-1)^x = 1$ справедливо, если x – четное, то есть, если в разложении x в виде многочлена свободный член $q_0 = 0$. Соответственно, $(-1)\hat{}x = (-1)$, если $q_0 = 1$.

Значит, q_0 всегда можно определить по $b^{(p-1)/2}$ таким образом:

$$\boldsymbol{b^{(p-1)/2}\ (mod\ p) \equiv \begin{cases} 1, q_0 = 0 \\ -1, q_0 = 1 \end{cases}} \quad (3)$$

Упрощенный вариант описания

Теперь преобразуем $b \equiv a^x (mod\ p) \equiv a^{q0+x1}\ (mod\ p)$

Где x_1 – многочлен (x – q_0)

Можно ввести новую переменную $z_1 \equiv b * a^{-q0} \equiv a^{x1}\ (mod\ p)$

Рассуждая образом, схожим с тем, что привел нас к выводу (3), приходим к выводу, что

$$z_1{}^{(p-1)/4} \equiv \begin{cases} 1, q_1 = 0 \\ -1, q_1 = 1 \end{cases} \quad (4)$$

Откуда находим q_1.

Вполне чётко вырисовывается общий алгоритм нахождения всех , q_i:

Обозначим степень за $m_i = (p-1)/2^{i+1}$

$$z_i \equiv b * a^{-q0-q1*2-\cdots-qn*2^{i-1}} \equiv a^{xi}\ (mod\ p)$$

Где

$$x_i = \sum_{k=i}^{n-1} q^k * 2^k$$

$z_i{}^{m_i} \equiv (-1)^{qi}\ (mod\ p)$

См. (4), легко находим q_i.

В результате нетрудно вывести $\boldsymbol{x = q_0 + q_1 * 2^1 + \cdots + q_{n-1} * 2^{n-1}}$.

Алгоритм (основной)

1 шаг. Для каждого простого числа $q, q \mid p -$ 1 составляем таблицу чисел

$$r_{q,j} \equiv a^{\frac{j(p-1)}{q}} (mod\ p) \quad j = 0, \dots, q-1$$

2 шаг. Для каждого просто числа q, $q, q^{\propto} \mid p-1$ находим $\log_a b\ (mod\ q^{\propto})$

Пусть $x \ \equiv \log_a b\ (mod\ q^{\propto}) \equiv x_0 + x_1 q + \cdots + x_{\propto-1} q^{\propto-1} (mod\ q^{\propto})$, где $0 \ll x_i \ \ll q-1$. Тогда из (1) следует, что

$$b^{\frac{(p-1)}{q}} \equiv a^{\frac{x_0(p-1)}{q}} (mod\ p)$$

С помощью таблицы шага 1 находим x_0. Тогда выполнено сравнение:

$$(ba^{-x_0})^{\ (p-1)/q} \equiv a^{\frac{x1(p-1)}{q}} (mod\ p)$$

Аналогично находим остальные x. Находим $\log_a b\ (mod\ p-1)$с помощью китайской теоремы об остатках

Об эффективности применения

Для применения алгоритма Полига-Хеллмана необходимо знать разложение на множители. В общем случае задача факторизации — достаточно трудоёмкая, однако если делители числа — небольшие (в том смысле, о котором сказано выше), то это число можно быстро разложить на множители даже методом последовательного деления. Таким образом, в том случае, когда эффективен алгоритм Полига-Хеллмана, необходимость факторизации не усложняет задачу.