Математические основы информатики

Теория чисел — II.

Сергей Леонидович Бабичев

Китайская теорема об остатках

Theorem (Китайская теорема об остатках)

Пусть $p_1, p_2, \ldots, p_k$ — попарно различные простые числа и пусть $P = p_1 \cdot p_2 \cdot \ldots \cdot p_k$. Тогда существует единственное неотрицательное решение по модулю P системы уравнений

$$\begin{cases} x \equiv a_1 \pmod{p_1} \\ x \equiv a_2 \pmod{p_2} \\ \ldots \\ x \equiv a_k \pmod{p_k} \end{cases}$$

Китайская теорема об остатках: доказательство для двух уравнений

Как мы доказали её для двух уравнений.

- Мы рассмотрели таблицы, составленные из остатков по модулю $p_1 \cdot p_2$ чисел вида $p_1 \cdot a_1 + p_2 \cdot a_2$.
- Чисел в таблице ровно $p_1 \cdot p_2$.
- Каждое число в таблице в диапазоне $[0; p_1 \cdot p_2)$.
- Мы доказали, что если p_1 и p_2 — взаимно простые, в этой таблице все остатки будут различны.
- Это означает, что для произвольного числа $d \in [0, p_1 \cdot p_2)$ найдётся такая пара a_1, a_2, что $d \pmod{p_1} = a_1$ и $d \pmod{p_2} = a_2$ и решение будет единственным.

Дальнейшее доказательство можно провести по индукции.

Китайская теорема об остатках: доказательство в общем случае

- Не умаляя общности положим, что $a_i \in [0; p_i)$.
- Тогда существует ровно $P = p_1 \cdot p_2 \cdot \ldots \cdot p_k$ таких систем.
- Для каждого $x \in [0; P)$ существует кортеж a_i.
- Достаточно доказать, что не существует двух разных чисел x_1 и x_2, $x_1, x_2 \in [0; P)$, для которых a_i одинаковы.
- Пусть $x_1 < x_2$.

Китайская теорема об остатках: доказательство в общем случае

- Вычтем почленно систему

$$\begin{cases} x_1 \equiv a_1 \pmod{p_1} \\ x_1 \equiv a_2 \pmod{p_2} \\ \dots \\ x_1 \equiv a_k \pmod{p_k} \end{cases}$$

из системы

$$\begin{cases} x_2 \equiv a_1 \pmod{p_1} \\ x_2 \equiv a_2 \pmod{p_2} \\ \dots \\ x_2 \equiv a_k \pmod{p_k} \end{cases}$$

Китайская теорема об остатках: доказательство в общем случае

- Разность $x_2 - x_1 \in (0, P)$ и является решением системы

$$\begin{cases} x \equiv 0 \pmod{p_1} \\ x \equiv 0 \pmod{p_2} \\ \dots \\ x \equiv 0 \pmod{p_k} \end{cases}$$

Это означает, что $(x_2 - x_1)$ делится на каждое из p_i.
Наименьшее из таких чисел равно $p_1 \cdot p_2 \cdot \ldots \cdot p_k = P$. Противоречие.

Задача. Решить систему:

$$\begin{cases} x \equiv 2 \pmod 3 \\ x \equiv 1 \pmod 5 \end{cases}$$

Задача. Решить систему:

$$\begin{cases} x \equiv 2 \pmod 3 \\ x \equiv 1 \pmod 5 \end{cases}$$

Решение: [Наивное]
Составим таблицу остатков от деления на 3 и 5.

$x \pmod 5 \backslash x \pmod 3$	0	1	2
0	0	10	5
1	6	1	11
2	12	7	2
3	3	13	8
4	9	4	14

Найдём пересечение столбца с заголовком 2 и строки с заголовком 1.
Это 11. Значит, $x = 11 + 15t, t \in \mathbb{Z}$.

Китайская теорема об остатках

Решение: [Правильное]

- Что значит решить такую систему?
- С чего начинать решение?
- Начнём с первого уравнения.
- Если

$$x \equiv 2 \pmod 3,$$

то

$$x = 3k + 2,$$

где $k \in \mathbb{Z}$.

- Подставим во второе уравнение.

$$3k + 2 \equiv 1 \pmod 5$$

Обратные числа в модулярной арифметике.

- Нам нужно определить, на какое число нужно умножить p, чтобы получился остаток 1 по модулю m.
- Это — обратное число в поле вычетов.
- Его можно найти через *расширенный алгоритм Евклида* или через малую теорему Ферма.

Обратные числа в модулярной арифметике.

- Имея обратное число p^{-1} по модулю m, мы можем решать любые системы вида $ax + by = c$.

$$ax = c - by$$
$$ax = c \pmod{b}$$
$$x = c \cdot a^{-1} \pmod{b}$$

.

Китайская теорема об остатках: решение уравнения

Дорешаем уравнение

$$3k + 2 \equiv 1 \pmod 5,$$

- $3k = (1 - 2) \pmod 5 = 4 \pmod 5$
- Установим, что $3^{-1} \pmod 5 = 2$.
- Действительно, $(3 \cdot 2) \pmod 5 = 1$.
- Тогда $k = (3^{-1} \cdot 4) \pmod 5 = (2 \cdot 4) \pmod 5 = 3$.
- Подстановкой в уравнение убеждаемся, что мы правы.
- Подставляем в уравнение $x = 3k + 2$ получаем $x = 11$.

Использование КТО на практике

- Выберем n простых чисел $p_i, i = 1, n$.
- $P = \prod_{i=1}^{n} p_i$.
- КТО гласит, что

$$\forall x \in [0 \dots P)\ \exists \{a_1, a_2, \dots, a_n\} : 0 \leqslant a_i < p_i \text{ и } a_i \equiv x \pmod{p_i}.$$

- Существует биекция $x \Leftrightarrow \{a_1, \dots, a_n\}$.
- Исследуем свойства кортежей $\{a_1, \dots, a_n\}$

Использование КТО на практике

Для $x \Leftrightarrow \{a_1, \ldots, a_n\}$ и $y \Leftrightarrow \{b_1, \ldots, b_n\}, x, y \in [0, P)$:

- $x + y \equiv \{(a_1 + b_1) \pmod{p_1}, \ldots, (a_n + b_n) \pmod{p_n}\} \pmod{P}$
- $x - y \equiv \{(a_1 - b_1) \pmod{p_1}, \ldots, (a_n - b_n) \pmod{p_n}\} \pmod{P}$
- $x \cdot y \equiv \{(a_1 \cdot b_1) \pmod{p_1}, \ldots, (a_n \cdot b_n) \pmod{p_n}\} \pmod{P}$

- Каждый кортеж — представитель x в непозиционной системе счисления по основаниям $\{p_1, \ldots, p_n\}$.
- Над кортежами производятся те же операции сложения, вычитания, умножения.
- Сложность всех этих операций $\Theta(n) \rightarrow$ арифметика КТО применима для реализации длинных чисел.

Длинные числа и КТО

- Выбираются n простых чисел, образующих непозиционную систему счисления (*систему вычетов*).
- В ней представимы все числа $0 \leqslant x < \prod_{i=1}^{n} p_i$.
- Перевод любого числа в систему есть нахождение n остатков.
- Перевод числа из системы счисления надо автоматизировать.
- Для этого имеется *алгоритм Гарнера*.

Алгоритм Гарнера

Задача: кортежу $\{a_i, \ldots, a_n\}$ в системе счисления p_i найти представителя x.

- Пока всё было не очень автоматизировано. Поставим вычисления на поток.
- Будем искать разложение x в сумму:

$$x = x_0 + p_1 \cdot x_1 + p_1 \cdot p_2 \cdot x_2 + \cdots + p_1 \cdot p_2 \cdot \cdots \cdot p_{n-1} \cdot x_{n-1} \tag{1}$$

Известно, что

$$\begin{cases} x \equiv a_1 \pmod{p_1} \\ x \equiv a_2 \pmod{p_2} \\ \ldots \\ x \equiv a_n \pmod{p_n} \end{cases} \tag{2}$$

Подставим уравнение (1) в первое уравнение из (2).

Алгоритм Гарнера

$$x_0 + p_1 \cdot x_1 + p_1 \cdot p_2 \cdot x_2 + \cdots + p_1 \cdot p_2 \cdot \cdots \cdot p_{n-1} \cdot x_{n-1} \equiv a_1 \pmod{p_1} \qquad (3)$$

Отсюда следует $x_0 = a_1$.
Подставим уравнение (1) во второе уравнение из (2)

$$a_1 + p_1 \cdot x_1 + p_1 \cdot p_2 \cdot x_2 + \cdots + p_1 \cdot p_2 \cdot \cdots \cdot p_{n-1} \cdot x_{n-1} \equiv a_2 \pmod{p_2} \qquad (4)$$

Отсюда, так как мы в поле вычетов по каждому из p_i:

$$x_0 + p_1 \cdot x_1 \equiv a_2 \pmod{p_2}$$
$$p_1 \cdot x_1 \equiv a_2 - x_0 \pmod{p_2}$$
$$x_1 \equiv (a_2 - x_0) \cdot {p_1}^{-1} \pmod{p_2}$$

Введём обозначение $r_{i,j} = p_i^{-1} \pmod{p_j}$.
Тогда

$$x_1 \equiv (a_2 - x_0) \cdot r_{1,2} \pmod{p_2}$$

Алгоритм Гарнера

$$x_1 \equiv (a_2 - x_0) \cdot r_{1,2} \pmod{p_2}$$

Подстановка x_1 в третье уравнение 2 даёт:

$$x_0 + x_1 \cdot p_1 + x_2 \cdot p_2 \equiv a_3 \pmod{p_3}$$

$$x_1 \cdot p_1 + x_2 \cdot p_1 \cdot p_2 \equiv a_3 - x_0 \pmod{p_3}$$

Домножаем на $r_{1,3}$:

$$x_1 \cdot p_1 \cdot r_{1,3} + x_2 \cdot p_1 \cdot p_2 \cdot r_{1,3} \equiv (a_3 - x_0) \cdot r_{1,3} \pmod{p_3}$$

$$x_1 + x_2 \cdot p_2 \equiv (a_3 - x_0) \cdot r_{1,3} \pmod{p_3}$$

$$x_2 \cdot p_2 \equiv (a_3 - x_0) \cdot r_{1,3} - x_1 \pmod{p_3}$$

Домножаем на $r_{2,3}$:

$$x_2 \cdot p_2 \cdot r_{2,3} = x_2 \equiv ((a_3 - x_0) \cdot r_{1,3} - x_1) \cdot r_{2,3} \pmod{p_3}$$

Алгоритм Гарнера: рекуррента

$$x_n \equiv (((a_{n+1} - x_0) \cdot r_{1,n+1} - x_1) \cdot r_{2,n+1}) - x_2 \ldots \pmod{p_{n+1}}$$

Другая запись при раскрытии скобок:

$$x_n = \frac{a_{n+1} - (x_0 + x_1 \cdot p_1 + x^2 \cdot p_1 \cdot p_2 + \cdots + x_{n-1} \cdot p_1 \cdot p_2 \ldots p_{n-1})}{p_1 \cdot p_2 \cdot \cdots \cdot p_n} \pmod{p_{n+1}}$$

Задача. Найти $334^{882} \bmod 77$.
Решение: Разложим 77 на простые множители: $77 = 7 \cdot 11$.

$$x \equiv 334^{832} \pmod{77}$$

Перейдём к системе

$$\begin{cases} x \equiv 334^{832} \pmod{7} \\ x \equiv 334^{832} \pmod{11} \end{cases}$$

Уменьшим основания:

$$\begin{cases} 334 \equiv 5 \pmod{7} \\ 334 \equiv 4 \pmod{11} \end{cases}$$

По МТФ уменьшим показатели:

$$\begin{cases} 832 \equiv 4 \pmod{6} \\ 832 \equiv 2 \pmod{10} \end{cases}$$

Задача свелась к

$$\begin{cases} x \equiv 5^4 \equiv (-2)^4 \equiv 16 \equiv 2 \pmod 7 \\ x \equiv 4^2 \equiv 5 \pmod{11} \end{cases}$$

Это — КТО. $p_1 = 7, p_2 = 11$, $a_1 = 2, a_2 = 5$.
Решение ищем в виде $x_0 + 7x_1$.
$x_0 = a_1 = 2$
$x_1 \equiv (a_2 - x_0) \cdot r_{1,2} \equiv (5 - 2) \cdot (7^{-1} \pmod{11}) \equiv 3 \cdot 8 \equiv 24 \equiv 2 \pmod{11}$
$x = 2 + 2 \cdot 7 = 16$.

- Пусть $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0$ — многочлен с целыми коэффициентами.
- Нас будут интересовать

$$f(x) \equiv 0 \pmod{m} \tag{5}$$

- Если $m = p_1^{\alpha_1} \dots p_n^{\alpha_n}$, то (5) эквивалентно

$$\begin{cases} f(x) \equiv 0 \pmod{p_1^{\alpha_1}} \\ \dots \\ f(x) \equiv 0 \pmod{p_n^{\alpha_n}} \end{cases} \tag{6}$$

Theorem (Теорема об уменьшении порядка многочлена)

Сравнение

$$f(x) \equiv 0 \pmod{p}, \deg f(x) \geqslant p$$

можно заменить на сравнение

$$g(x) \equiv 0 \pmod{p}, \deg g(x) < p.$$

Теорема об уменьшении порядка многочлена

Доказательство.

Разделим $f(x)$ на $x^p - x$ с остатком:

$$f(x) = q(x) \cdot (x^p - p) + g(x)$$

Здесь $\deg g(x) < p$.
Все классы вычетов по модулю p удовлетворяют сравнению:

$$(x^{p-1}) \cdot x \equiv 0 \pmod{p}.$$

Отсюда сравнения $f(x) \equiv 0 \pmod{p}$ и $g(x) \equiv 0 \pmod{p}$ имеют одинаковые множества решений. □

Задача. Найдите все x, для которых

$$f(x) = 3x^7 + 2x^6 + x^5 - 3x^3 - x^2 - x - 1 \equiv 0 \pmod 5$$

Решение: Разделим $f(x)$ с остатком на $x^5 - x$.

$$f(x) = (3x^2 + 2x + 1)(x^5 - x) + x^2 - 1 \equiv 0 \pmod 5.$$

Отсюда $x \equiv \pm 1 \pmod 5$.

Definition (Квадратичные вычеты)

Для $a, m \in \mathbb{Z}, \gcd(a, m) = 1$ число a называется *квадратичным вычетом по модулю* m, если сравнение $x^2 \equiv a \pmod{m}$ разрешимо.
Иначе a называется *квадратичным невычетом* по модулю m.

Lemma (Симметрия квадратичных вычетов)

Если $m > 2, m \in \mathbb{P}$, a *— квадратичный вычет по модулю* m, *то* $x^2 \equiv a \pmod{m}$ *имеет два решения.*

Lemma (Количество квадратичных вычетов)

В приведённой системе вычетов по модулю m *количество квадратичных вычетов равно количеству квадратичных невычетов.*

Lemma (Мультипликативные свойства квадратичных вычетов)

Произведение двух вычетов — вычет, произведение двух невычетов — вычет, произведение вычета и невычета — вычет.

Символ Лежандра

Definition (Символ Лежандра)

Символ Лежандра $\left(\frac{a}{m}\right)$

$$\left(\frac{a}{m}\right) = \begin{cases} 1, \text{ если } a - \text{ квадратичный вычет по модулю } m \\ -1, \text{ если } a - \text{ квадратичный невычет по модулю } m \\ 0, \text{ если } a \vdots m \end{cases}$$

Формула Эйлера:

$$\left(\frac{a}{m}\right) \equiv a^{\frac{m-1}{2}} \pmod{m}$$

Свойства символа Лежандра

1. Для $a \equiv b \pmod{m}$ верно $\left(\frac{a}{m}\right) = \left(\frac{b}{m}\right)$;
2. $\left(\frac{ab}{m}\right) = \left(\frac{a}{m}\right) \cdot \left(\frac{b}{m}\right)$;
3. $\left(\frac{1}{m}\right) = 1$;
4. $\left(\frac{-1}{m}\right) = (-1)^{\frac{m-1}{2}}$;
5. $\left(\frac{a^2}{m}\right) = 1$.
6. Для $m = p_1 \cdot p_2 \cdot \dots \cdot p_k, p_i > 2$ верен символ Якоби

$$\left(\frac{a}{m}\right) = \left(\frac{a}{p_1}\right) \cdot \left(\frac{a}{p_1}\right) \cdot \dots \cdot \left(\frac{b}{p_k}\right).$$

Задача. Для большого n-чанкового числа x определить, является ли оно полным квадратом. Сложность операций сложения двух n-чаковых чисел $\Theta(n)$, операций умножения — $\Theta(n^2)$. Операции извлечения квадратного корня — нет. Придумайте способ, как для очень большого числа запросов получать ответы максимально быстро. Полагаем, что число правильных квадратов во входящей последовательности невелико.

Задача. Для большого n-чанкового числа x определить, является ли оно полным квадратом. Сложность операций сложения двух n-чаковых чисел $\Theta(n)$, операций умножения — $\Theta(n^2)$. Операции извлечения квадратного корня — нет. Придумайте способ, как для очень большого числа запросов получать ответы максимально быстро. Полагаем, что число правильных квадратов во входящей последовательности невелико.

Решение:

- Поймём, что сложность нахождения остатка от операции деления n-чанкового числа на короткое $\Theta(n)$.
- Составим таблицы $M_{i,j} = \left(\frac{a}{p_i}\right)$ для всех $a \in \mathbb{Z}_{p_i}$ по модулям $p_1, p_2, \ldots, p_k \in \mathbb{P}$.
- Для каждого проверяемого числа x находим $x \bmod p_i$.
- Если табличное значение равно -1, то число — не точный квадрат.
- Если число не являлось точным квадратом, то каждая проверка завершит работу с вероятностью 0.5.
- После k проверок вероятность, точного квадрата будет 2^{-k}.

Подгруппы

- В поле вычетов по модулю p для каждого элемента a существует обратный по операции умножения: $ax \equiv 1 \pmod{p}$.
- Если S образует группу по операции $\circ$, $S' \subseteq S$ и S' образует группе по операции $\circ$, то S' — *подгуппа группы S по операции $\circ$*.
- Пример — множество чётных чисел — подгруппа $\mathbb{Z}$ по операции сложения.

Theorem (Существование подгруппы)

Если S — конечная группа по операции $\circ$, S' — непустое подмножество S такое, что $a \circ b \in S'$, то S' — подгруппа S по операции $\circ$.

Theorem (Лагранжа)

Если S — конечная группа по операции $\circ$, S' — подгруппа S по операции $\circ$, то $|S| \vdots |S'|$.

Генераторы подгрупп

Здесь и далее речь идёт о группах по операции $\circ$.

- Для элемента $a \in S$ определим все элементы, которые из него могут получиться.
- $a^{(k)} = \underbrace{a \circ a \cdots \circ a}_{k}$
- Введём обозначение $\langle a \rangle$ для такой подгруппы.
- Порядок элемента в подгруппе $\langle a \rangle$ называемый $\mathrm{ord}(a)$ есть минимальное $k \in \mathbb{N}_1 : a^{(k)} = e$.

Theorem (Порядок группы)

Для любой конечной группы S *и* $\forall a \in S \quad \mathrm{ord}(a) = |\langle a \rangle|$.

Задача. Хеш-таблица использует открытую адресацию с рехешированием. Для вновь пришедшего ключа вычисляется $h = H(key) \bmod S$, где S — размер таблицы. Если позиция h занята, вычисляется $h1 = H1(key) \bmod S$. После это делаются попытки вставить ключ в позиции $(h + h1) \bmod S$, $(h + 2 \cdot h1) \bmod S$, $(h + 3 \cdot h1) \bmod S$ — до успеха. Каким условиям должны удовлетворять S, h и $h1$, чтобы множество возможных точек вставки ключа было максимальным?

Порядки мультипликативных групп

Здесь и далее речь идёт о группах $\mathbb{Z}_m$ по операции умножения.
Свойства групп и их порядков:

1. Если $a \equiv b \pmod{m}$, то $\operatorname{ord}(a) = \operatorname{ord}(b)$.
2. Числа $a^0, a_1, \ldots, a_k - 1$ различны по модулю m, $k = \operatorname{ord}(a)$.
3. $\varphi(m) \vdots \operatorname{ord}(a)$.
4. Если $a^n \equiv 1 \pmod{m}$ то $m \vdots \operatorname{ord}(a)$.
5. Если $a^k \equiv a^l \pmod{m}$, то $k \equiv l \pmod{\operatorname{ord}(a)}$.
6. В поле вычетов по модулю p для каждого элемента a существует обратный по операции умножения: $ax \equiv 1 \pmod{p}$.

Первообразные корни и дискретные логарифмы

Definition (Первообразный корень)

Число a есть *первообразный корень по модулю* m, если $\operatorname{ord}(a) = \varphi(m)$.

- Пусть p — фиксированное простое число, g — некоторый первообразный корень по модулю p.
- g^0, g^1, g^2, g^{p-2} образуют приведённую систему вычетов по модулю p.
- Для всякого $a \in \mathbb{Z}_p$ найдётся такое $l \in [0, p-1)$, что $a = g^l \pmod{p}$.
- l — *дискретный логарифм* числа a по основанию g и модулю p.

$$l = \operatorname{ind}_g a$$

Алгоритм Диффи-Хеллмана

- Имеются два несекретных числа:
  - p — простое число;
  - g — первообразный корень по модулю p.
- Основан на том, что $g^{ab} \bmod p = g^{ba} \bmod p$ и невозможности за разумное время по известным $g^a \bmod p$ и $g^b \bmod p$ вычислить $g^{ab} \bmod p$ при больших p, a, b.
  Задача дискретного логарифмирования трудноразрешима.

Алгоритм Диффи-Хеллмана

- A и B хотят взаимно получить число, известное лишь им.
- Они выбрали $p = 13, g = 6$.
  - A выбирает произвольно приватный ключ $a = 10$.
  - A вычисляет $A' = g^a \bmod p = 6^{10} \bmod 13 = 4$ и посылает его B.
  - B выбирает произвольно приватный ключ $b = 3$.
  - B вычисляет $B' = g^b \bmod p = 6^3 \bmod 13 = 8$ и посылает его A
  - A вычисляет $s = B'^a \bmod p = 8^{10} \bmod 13 = 12$.
  - B вычисляет $s = A'^b \bmod p = 4^3 \bmod 13 = 12$.
- s — искомый секрет, известный лишь двоим.
- Всё остальное может быть известно всем.

Алгоритм RSA

1. Находится пара больших простых чисел P и Q
2. $N = P \cdot Q$.
3. $Z = (P-1)(Q-1)$.
4. Выбирается $E : \gcd(E, Z) = 1$.
5. Вычисляется $D : D = E^{-1} \pmod{Z}$
6. Пара $P = E, N$ — публичный ключ.
7. Пара $S = D, N$ — приватный (секретный) ключ.

$C_i = M_i^E \bmod N$ — шифрование
$M_i = C_i^d \bmod N$ — дешифрование