

# Алгоритмы и структуры данных

Лекция 19

Числа.

Сергей Леонидович Бабичев

# Проверка на простоту. Факторизация. Длинная арифметика.

# Символ Лежандра

## Definition (Символ Лежандра)

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

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

**Задача.** Для большого  $n$ -чанкового числа  $x$  определить, является ли оно полным квадратом. Сложность операций сложения двух  $n$ -чанковых 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  $n$ -чанковых 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  $p_1, p_2, \dots, p_k \in \mathbb{P}$ .
- 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  $2^{-k}$ .

**Задача.** Хеш-таблица использует открытую адресацию с рехешированием. Для вновь пришедшего ключа вычисляется  $h = H(\text{key}) \bmod S$ , где  $S$  — размер таблицы. Если позиция  $h$  занята, вычисляется  $h1 = H1(\text{key}) \bmod S$ . 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?

# 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.

$C_i = M_i^E \pmod{N}$  — шифрование

$M_i = C_i^D \pmod{N}$  — дешифрование

# Корректность RSA

## Theorem

Пусть  $N = pq$  — произведение двух различных простых чисел,  $e$  и  $d$  — два натуральных числа, меньших  $(p-1)(q-1)$  таких, что  $ed \equiv 1 \pmod{p-1}$  и  $ed \equiv 1 \pmod{q-1}$ . Тогда для любого  $x \in \mathbb{Z}_n$  верно  $x^{ed} \equiv x \pmod{N}$ .

## Доказательство.

$$\varphi(N) = \varphi(p)\varphi(q) = (p-1)(q-1).$$

По китайской теореме об остатках отображение  $\mathbb{Z}_n \rightarrow \psi(x) = (x_p, x_q), p \in \mathbb{Z}_p, q \in \mathbb{Z}_q$  изоморфно.

Поэтому  $\psi(x^{ed}) = (x_1^{ed}, x_2^{ed})$ . Так как  $ed = 1 + k(p-1)$ , то

$$x_1^{ed} = x_1^{1+k(p-1)} = x_1 \cdot x_1^{k(p-1)} = x_1(x_1^{p-1})^k = x_1 \text{ в } \mathbb{Z}_p. \text{ Аналогично для } x_2. \quad \square$$

# Проверка числа на простоту

# Вероятностный тест Миллера-Рабина

## Definition (Псевдопростое число)

Натуральное число  $n$  — псевдопростое по основанию  $a \in \mathbb{Z}_n^+$ , если

$$a^{n-1} \equiv 1 \pmod{n}. \quad (1)$$

- Если для какого-то основания  $a$  не выполнено (1), то  $n$  — составное.
- Существуют такие составные  $n$  (числа Кармайкла), что для любых  $a$  выполняется (1). Их 225 среди первых 100'000'000 чисел.
- Косвенное подтверждение простоты  $n$  — выполнение для любого  $a \in \mathbb{Z}_n^+$  сравнений:

$$a^u \equiv 1 \pmod{n}, a^{2^r u} \equiv -1 \pmod{n}, 0 \leq r < s, \quad (2)$$

где  $s$  — максимальная степень 2, делящая  $n - 1$ .

- Тест Миллера-Рабина проверяет (1) и (2) для случайных  $a \in \mathbb{Z}_n^+$ .

# Алгоритм Миллера-Рабина простоты $n$ по основанию $a$

1. Тривиально:  $n$  — чётное — не простое. Конец.
2. Если  $\gcd(a, n) \neq 1$  — не простое. Конец.
3.  $b = a^n$ . Если  $b \equiv 1 \pmod{n} \vee b \equiv -1 \pmod{n}$  — возможно, простое. Конец.
4. Вычисляем по модулю  $n$   $b^2, b^4, \dots, b^{2^s}$ . Если хотя бы одно значение равно  $-1$ , то, возможно, простое. Иначе — точно не простое.

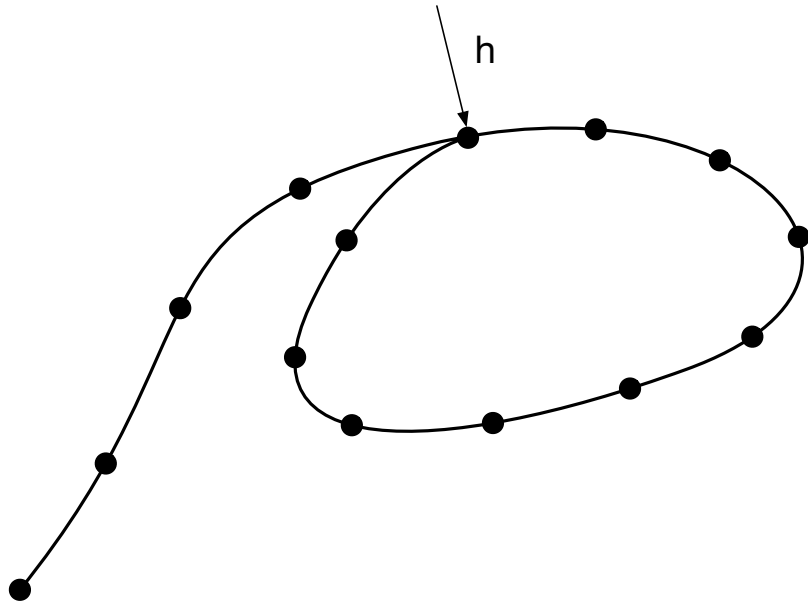
Алгоритм выполняется для случайных чисел  $a$  несколько раз. Вероятность каждого испытания —  $\frac{1}{2}$ .

# Факторизация

# Алгоритм $\rho$ -Полларда

- Пусть имеется функция  $f$ , отображающая множество  $\mathbb{Z}_n$  на множество  $\mathbb{Z}_n$ .
- Построим последовательность  $\{x_i\} : x_i \in \mathbb{Z}_n, x_{i+1} = f(x_i)$ .
- Для каких-то  $h > 0, k > 0$  числа  $x_1, x_2, \dots, x_h, x_{h+1}, \dots, x_{h+k-1}$  будут различны (их конечное количество), а  $x_{h+k} = x_h$ .
- Последовательность  $\{x_i\}$  станет периодической с периодом  $k$ , начиная с  $i = h$ .  $h$  — предпериод.

# $\rho$ для последовательностей





# Алгоритм $\rho$ –Полларда

- Среднее число итераций в  $\mathbb{Z}_n$  до появления повторений оценивается как  $O(\sqrt{n})$ .
- Если  $m$  — делитель  $n$ , то можно построить последовательность из остатков по модулю  $m$ .
- Среднее число членов  $x$  до повторения будет  $m$ .
- Для какого-то  $r$  порядка  $\sqrt{m}$   $x'_r = x'_s$ .
- Но  $x_s - x_r \not\equiv 0 \pmod{m}$ .
- Так как  $n \not\equiv 0 \pmod{m}$ , то  $n \not\equiv d = \gcd(|x_s - x_r|, n)$ .
- Если  $d \neq n$ , нетривиальный делитель  $n$  найден.
- Сложность порядка  $O(N^{\frac{1}{4}})$ .
- В качестве  $f(x)$  выбирается  $x^2 + a, a \neq 0 \vee 2$ .