Лекция 2

1. Математический базис с КОК

2. Генерирование простых чисел

Малая теорема Ферма

Если p – простое число и p не делит a, то $a^{p-1} = 1 \bmod p$

Доказательство. Заметим, что *0, a, 2a, … , (p – 1)a* различны по mod p. В противном случае, если предположить, что $i \cdot a = j \cdot a \bmod p$, при $i \neq j$, то $(i - j)a = 0 \bmod p$ и поэтому *p делит(i - j)a* . Но поскольку *p* не делит *a* и *i*, *j* < *p* , то сделано неверное предположение, и тогда числа *0, a, 2a, … , (p – 1)a* составляют всего лишь перестановку чисел *1, 2, … , p - 1*. Следовательно, справедливы следующие равенства:

$$a \cdot 2a \cdots (p-1)a = a^{p-1}(p-1)! \bmod p$$

$$1 \cdot 2 \cdots (p-1) = (p-1)! \bmod p$$

$$a^{p-1}(p-1)! \bmod p = (p-1)! \bmod p$$

Отсюда следует, что

Сокращая обе стороны тождества на (p-1)!

получаем : $a^{p-1} - 1 = 0 \bmod p$.

Функция Эйлера

Определение 3. Пусть n – целое натуральное число, тогда *функцией Эйлера* $\varphi(n)$ называется количество целых неотрицательных чисел, меньших n и взаимно простых с n, т. е.: $\varphi(n) = \#\{0 \le b < n; \gcd(b, n) = 1\}$

где $\#\{X\}$ означает количество элементов множества X.

Свойства:

1. $\varphi(1) = 1$,
2. $\varphi(p) = p - 1$, если p –простое число.
3. $\varphi(p^n) = p^n - p^{n-1}$ или $\varphi(p^n) = p^n\left(1 - \frac{1}{p}\right)$
4. Функция Эйлера мультипликативна $\varphi(m \cdot n) = \varphi(m) \cdot \varphi(n)$

$$\varphi(p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_s^{\alpha_s}) = p_1^{\alpha_1 - 1} \cdot p_2^{\alpha_2 - 1} \cdot \ldots \cdot p_s^{\alpha_s - 1}(p_1 - 1) \cdot (p_2 - 1) \cdot \ldots \cdot (p_s - 1)$$

Другая запись свойства 4

$$\varphi(n) = p_1^{\alpha_1}\left(1 - \frac{1}{p_1}\right)p_2^{\alpha_2}\left(1 - \frac{1}{p_2}\right)\ldots p_r^{\alpha_r}\left(1 - \frac{1}{p_r}\right) = n\prod_{p|n}\left(1 - \frac{1}{p}\right)$$

$$n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_s^{\alpha_s}$$

Теорема Эйлера (обобщение теоремы Ферма)

Если gcd(a,m) = 1 , то $a^{\varphi(m)} = 1 \bmod m$

где $\varphi(m)$ *– функция Эйлера* [1].

Теорема Ферма – это частный случай теоремы Эйлера. Действительно, если m = p – простое число, то по теореме Эйлера $\varphi(p) = p - 1$, что и дает утверждение теоремы Ферма: $a^{p-1} = 1 \bmod p$.

Утверждение 1. (Полезное для ускорения вычисления степени по модулю.) *Если gcd(a, m) = 1,* $n' = n \bmod \varphi(m)$, то $a^{n'} \bmod m = a^n \bmod m$ [3].

Утверждение 2. (Полезное для анализа стойкости криптосистем с открытым ключом.) *Пусть* $n = p \cdot q$, *где* $p \cdot q$ *– простые числа.* $p \neq q$ *Тогда числа p и q можно найти, если известно n и* $\varphi(n) = (p-1)\cdot(q-1)$

Доказательство. Будем рассматривать *p*, *q* как пару неизвестных целых чисел, для которых задано их произведение $p \cdot q = n$ и известна сумма, поскольку $n+1-\phi(n) = p \cdot q+1-(p-1)\cdot(q-1) = p+q = 2b$, где b – некоторое целое число.

Два числа, сумма которых равна 2b , а произведение равно *n*, являются очевидно корнями уравнения $x^2 - 2bx + n = 0$ (теорема Виета). Тогда корни квадратного уравнения и есть необходимые числа *p* и *q*:

.

$$p = b + \sqrt{b^2 + n^2};\ q = b - \sqrt{b^2 + n^2}$$

Сложность решения этого уравнения – $O(\log^3 n)$

1.2. Китайская теорема об остатках

Пусть $\gcd(m_i, m_j)=1$ *для* $i \neq j$ *. Тогда система уравнений*

$$\left.\begin{aligned} x &= a_1 \bmod m_1; \\ x &= a_2 \bmod m_2; \\ &\vdots \\ x &= a_r \bmod m_r \end{aligned}\right\} \qquad (2.6)$$

имеет решение, и при этом если два числа x' *и* x'' *решения данной системы, то они удовлетворяют уравнению*

$$x' = x'' \bmod M \qquad (2.7)$$

где $M = m_1 \cdot m_2 \cdots m_r$

- **Доказательство.** Докажем однозначность решения по $\bmod M = m_1 \cdot m_2 \cdots m_r$ Предположим, что есть два решения системы (2.6) x' и x''. Обозначим $y = x' - x''$, тогда y удовлетворяет системе

$$\left.\begin{array}{l} y = O \bmod m_1; \\ y = O \bmod m_2; \\ \vdots \\ y = O \bmod m_r \end{array}\right\} \Rightarrow y = O \bmod M$$

- так как $m_1, m_2, ..., m_r$ – взаимно простые. Отсюда и следует, что
- $$x' = x'' \bmod M$$
- Покажем теперь, как сконструировать хотя бы одно решение x.
- .

Обозначим $M_i = \frac{M}{m_i}$. Очевидно, что $\gcd(m_i, M_i) = 1$, поэтому существует обратный элемент N_i к M_i по *mod* m_i, т. е. $M_i^{-1} = N_i$, $M_i \cdot N_i = 1 \bmod m_i$, который может быть найден по алгоритму Евклида для нахождения обратных элементов.

Положим теперь

$$x = \sum_{i=1}^{r} a_i M_i \cdot N_i \bmod M = (a_1 M_1 N_1 + a_2 M_2 N_2 + \cdots + a_r M_r N_r) \bmod M$$

Данное решение будет решением системы (2.6). Действительно, так как m_i делит M_j, $i \neq j$, видно, что все слагаемые будут равны нулю по *mod* m_i, за исключением i-го слагаемого.

Тогда получаем $x = a_i \underbrace{M_i N_i}_{1} \bmod m_i$, и поэтому $x = a_i \bmod m_i$, при $i = 1, 2, \ldots, r$, т. е. x – решение системы (2.6).

Пример решения системы уравнений

$$\begin{cases} \boldsymbol{x} = 2 \, mod \, 3 \\ \boldsymbol{x} = 3 \, mod \, 5 \\ \boldsymbol{x} = 2 \, mod \, 7 \end{cases}$$

1. $\boldsymbol{M} = 3 \cdot 5 \cdot 7 = 105$

2. $\boldsymbol{M}_1 = 105 / 3 = 35, \boldsymbol{M}_2 = 105 / 5 = 21, \boldsymbol{M}_3 = 105 / 7 = 15$

3. $\boldsymbol{N}_1 = 2, \boldsymbol{N}_2 = 1, \boldsymbol{N}_3 = 1,$

4. $\boldsymbol{x} = (2 \cdot 35 \cdot 2 + 3 \cdot 21 \cdot 1 + 2 \cdot 15 \cdot 1) = 23 \, mod \, 105$

1.3.Цепные дроби

- Цепная дробь $[a_0, a_{01}, \ldots, a_n \ldots]$ определяется как формальная сумма

$$a_0 = \cfrac{1}{a_1 + \cfrac{1}{a_2 + \ddots \cfrac{1}{a_{n-1} + \cfrac{1}{a_n}}}}$$

- Числа $a_0, a_1, \ldots, a_k$ *k=0,1,…..,n* называются неполными частными цепной дроби, а величины $\alpha_k = [a_k, a_{k+1}, \ldots, a_n \ldots]$

k=0,1,…..,n называются полным частными цепной дроби.

- Числа $\delta_k = [a_0, a_1, \ldots, a_k]$ называются подходящими дробями к цепной дроби

Пример цепной дроби

$$[-3,2,1,4] = -3 + \cfrac{1}{2 + \cfrac{1}{1 + \cfrac{1}{4}}} = -\frac{47}{13}$$

Неполные частные имеют вид $a_0 = -3 \; a_1 = 2 \quad a_2 = 1 \quad a_3 = 4$

полные частные имеют вид $\alpha_0 = [-3,2,1,4] = -3 + \cfrac{1}{2 + \cfrac{1}{1 + \cfrac{1}{4}}} = -\frac{47}{13}$

$$\alpha_1 = [2,1,4] == 2 + \cfrac{1}{1 + \cfrac{1}{4}} = \frac{14}{5}$$

$$\alpha_2 = [1,4] = 1 + \frac{1}{4} = \frac{5}{4}$$

$$\alpha_3 = [4] = 4$$

Пример цепной дроби (продолжение)

- Цепная дробь, образованная отбрасыванием всех элементов после некоторого номера k, называется k-ой подходящей дробью

$$\delta_0 = [-3] = -3$$

$$\delta_1 = [-3, 2] = -3 + \frac{1}{2} = -\frac{5}{2}$$

$$\delta_2 = [-3, 2, 1] = -3 + \frac{1}{2 + \frac{1}{1}} = -\frac{8}{3}$$

$$\delta_3 = [-3, 2, 1, 4] = -3 + \frac{1}{2 + \frac{1}{1 + \frac{1}{4}}} = -\frac{47}{13}$$

- Подходящие дроби можно представить рациональными числами $\frac{P_k}{Q_k}$ *k=0,1,…..,n* .

$$\delta_0 = \frac{P_0}{Q_0} = \frac{a_0}{1} \qquad \delta_1 = \frac{P_1}{Q_1} = \frac{a_0 a_1 + 1}{a_1} \qquad \delta_k = \frac{P_k}{Q_k} = \frac{a_k P_{k-1} + P_{k-2}}{a_k Q_{k-1} + Q_{k-2}} \text{ для } \quad k \geq 2$$

Свойства:

1. $P_n Q_{n-1} - P_{n-1} Q_n = (-1)^{n-1}$
2. $P_n Q_{n-2} - P_{n-2} Q_n = (-1)^{n-1} a_n$
3. $(P_n Q_n) = 1$
4. $1 = Q_0 \leq Q_1 < Q_2 < \ldots$
5. *Если* $P_0 > 1$, *то* $P_1 > P_2 > \ldots$
6. $\delta_n - \delta_{n-1} = \frac{(-1)^{n-1}}{Q_n Q_{n-1}}$
7. *Если* $a = [a_0, a_1, \ldots, a\ , \ldots]$, *то*
$\delta_0 < \delta_2 < \ldots < \delta_{2k} < \ldots \leq a \leq \ldots \delta_{2k+1} < \ldots < \delta_3 < \delta_1$
8. *Если* $a = [a_0, a_1, \ldots, a\ , \ldots]$, *то* $\left|a - \delta_n\right| \leq \frac{1}{Q_n Q_{n-1}}$ 1

Вычисление цепной дроби

Достаточно выписать Алгоритм Евклида для чисел P иQ, и взять столбец , полученный при этом целых частных в качестве неполных частных искомой дроби

$$\frac{P}{Q}=\frac{173}{281}$$

$$173 = 281\cdot 0 + 173$$
$$281 = 173\cdot 1 + 108$$
$$173 = 108\cdot 1 + 65$$
$$108 = 65\cdot 1 + 43$$
$$65 = 43\cdot 1 + 22$$
$$43 = 22\cdot 1 + 421$$
$$22 = 21\cdot 1 + 1$$
$$21 = 1\cdot 21 + 0$$

$$\frac{P}{Q}=\frac{173}{281}=[0,1,1,1,1,1,1,21]$$

Применение цепных дробей

1. Для приближения действительных чисел рациональными с наилучшим приближением.
2. Для сокращения обыкновенных дробей
3. Для решения диофантовых уравнений с двумя неизвестными.
4. Для решения сравнений первой степени $ax \equiv b(\mathrm{mod}\, m)$.

П.1 Любая подходящая дробь $\delta_k = [a_0, a_1, \ldots, a_k],\ k = 0,1,2\ldots$

является наилучшим приближением к действ. числу

$a = [a_0, a_1, \ldots, a_n, \ldots].$

В основе практического применения используется свойство $\left|a - \delta_n\right| \le \left|\delta_{n+1} - \delta_n\right| = \frac{1}{Q_{n+1}Q_n}$ Для нахождения наилучшего приближения с точностью Δ, рассматриваются знаменатели тех подходящих дробей, для которых $Q_{n+1}Q_n > \Delta^{-1}$

Применение цепных дробей (продолжение)

П.3. Если НОД (a,b)=1, то подходящая дробь может быть использована для решения диофантового уравнения

$$ax + by = d$$

Целочисленные решения уравнения можно найти как

$$x = (-1)^{n-1} Q_{n-1}, \quad y = (-1)^{n-1} P_{n-1}$$

П.4. Для решения сравнения $ax \equiv b (\mathrm{mod}\, n)$, полагая что (a,n)=1.

Разложим n/a в цепную дробь $[a_0, a_1, \ldots, a_k]$, тогда

$$x = (-1)^k \cdot b \cdot P_k \, (\mathrm{mod}\, n)$$

есть искомое решение уравнения.

1. 4. Квадратичные вычеты

Квадратичные вычеты. Рассмотрим поле *GF(p)*, где *p* – простое число, *GF(p)* состоит из элементов: 0, 1, 2, 3, … , *p - 1* . Предположим, что *p>2* . Ставится вопрос: какие из элементов этого поля являются квадратами этих или других элементов этого поля?

Определение 1. Если $a \in GF(p)$ является квадратом некоторого элемента $b \in GF(p)$, т. е. $a = b^2$, $b \in GF(p)$, то такой элемент поля *a* называется ***квадратичным вычетом.*** Остальные элементы поля, не представимые в таком виде, называются *квадратичными невычетами*.

Пример 1. Если *p = 11*, то вычетами в таком поле являются 1, 4, 9, 5, 3, так как $1^2 = 1$, $2^2 = 4$, $3^2 = 9$, $4^2 = 5$, $5^2 = 3$. Элементы 2, 6, 7, 8, 10 (как легко проверить) будут невычетами.

Если записать ненулевые элементы поля *GF(p)* как степени примитивного элемента α , $\alpha^1, \alpha^2, \alpha^3, \ldots, \alpha^{p-1} = 1$, то в этом случае квадратичные вычеты имеют вид: α^j , где *j* – четное число.

Чтобы определить, является ли элемент $a \in GF(\hat{p})$ квадратичным вычетом, используются *символы Лежандра*.

Определение 2. Символом Лежандра числа a и простого числа p называется

$$\left(\frac{a}{p}\right)=\begin{cases}0, \text{ если } p \mid a;\\ +1, \text{ если } a \text{ квадратичный вычет в } GF(p);\\ -1, \text{ если } a \text{ невычет в } GF(p).\end{cases}$$

Утверждение 4. Символ Лежандра может быть вычислен по формуле

$$\left(\frac{a}{p}\right)=a^{(p-1)/2} \bmod p \qquad [2, 3].$$

Однако данный метод не позволяет найти квадратный корень из a по *mod* p, даже если известно, что a – вычет.

Нахождение вычетов

Нахождение вычета равносильно решению задачи нахождения квадратного корня уравнения

$r=\sqrt{a}$ modp

Простое число p может быть представлено

либо как p=4k+3, либо как p=4k+1, где k положительное целое число.

В **первом случае** корень находится просто:

-найти r=a^(p+1)/4modp,

-выдать в качестве ответа (r,-r).

Пример для 1-го случая.

r^2=3mod23, 23=4*5+3

Находим 3^(24/4)=3^6mod23=16.

r=+16, r=-16.

Проверка 16^2=256mod23=3

Во **втором случае** задача усложняется.

Известен алгоритм решения задачи $\sqrt{a} \bmod p$, если найдено некоторое другое число $b \in GF(p)$, которое дает $\left(\frac{b}{p}\right) = -1$, т. е. b – невычет.

Хотя сейчас не известен полиномиальный алгоритм, решающий задачу нахождения невычета, однако с вероятностью 50% при случайном выборе элемента $b \in GF(p)$ будем попадать на невычет. Следовательно, несколько попыток случайного выбора b с высокой вероятностью даст невычет.

Имея в своем распоряжении метод генерирования невычетов b, можно использовать следующую конструкцию для нахождения $\sqrt{a} \bmod p$ [2, 3]:

1) генерировать случайные числа $b \in Z_p$, $Z_p = \{0, 1, 2, 3, \ldots, p-1\}$, до тех пор, пока $b^2 - 4a$ не окажется квадратичным невычетом по *mod p*, т. е. $\left(\frac{b^2 - 4a}{p}\right) = -1$

2) найти $r = x^{(p+1)/2} \bmod\left(x^2 - bx + a\right)$,

где $\left(x^2 - bx + a\right)$ - полином над полем $GF(p)$

3) выдать ответ: r, $-r$ – как решение задачи $\sqrt{a} \bmod p$.

Сложность нахождения $\sqrt{a} \bmod p$ составляет $O\left((\log p)^3\right)$ битовых операций.

Когда ***n* составное число** $n = p \cdot q$, нахождение $\sqrt{a} \bmod n$ является весьма трудной задачей, и до сих пор не известно ни одного полиномиально сложного алгоритма ее решения, если p и q неизвестно.

- Общий порядок такой. 1. сначала нужно найти решения уравнения по простым модулям (сомножителям n)

2. затем, используя китайскую терему о остатках, получить решение системы.

(Нужно решить 4 системы из двух уравнений) и путем подстановки в исходное уравнение найти правильное решение.

Доказано, что по сложности эта задача нахождения вычета эквивалентна задаче факторизации чисел. Если p и q известны, то задача извлечения решается довольно просто по алгоритму, рассмотренному выше. Данный факт эффективно используется в криптосистемах с открытым ключом.

2. Генерирование простых чисел

В криптографии с открытым ключем необходимо уметь находить простые числа. Обычно эта задача решается в два этапа:

1) генерирование случайного или псевдослучайного (если число не является секретным ключом) числа, которое по размерности удовлетворяет предъявленным требованиям;

2) проверка, является ли выбранное нечетное число простым. Если является, то оно принимается. Если же это число не является простым, тогда нужно повторять эти этапы до появления успешного результата.

Возникает вопрос: сколько потребуется сделать попыток (в среднем) для генерирования простого числа заданной размерности?

9) Количество простых чисел

N	Количество простых чисел	%
10^2	25	25
10^4	1 229	12,3
10^6	78 498	7,8
10^8	5 761 455	5,8
10^{10}	455 052 511	4,6
10^{12}	37 607 912 018	3,8
10^{14}	3 204 941 750 802	3,2
10^{16}	279 238 341 033 925	2,8

Ответом на данный вопрос является следующая теорема.

Теорема [5]. Пусть Π(n) – число простых чисел, которые $\leq n$, тогда

$$\lim_{n\to\infty} \frac{\Pi(n)}{n/\ln n} = 1$$

Количество простых чисел ограничено сверху с снизу

$$[n/(lnn)] \leq \pi(n) \leq [n/(lnn - 1.08366)]$$

$\bar{s}$

Пример: n=1 000 000

$$72383 \leq \Pi(n) \leq 78543$$

Фактически $\Pi(n)$ = 78449

Из этой теоремы можно получить аппроксимацию доли нечетных
l-разрядных простых чисел в виде $\frac{2}{l \ln 10}$ т. е. среднее число попыток для генерирования *l*-разрядного простого числа равно $\bar{s} = \frac{l \cdot \ln(10)}{2}$.
(Для доказательства этого факта достаточно лишь заметить, что количество в точности *l*-разрядных нечетных чисел равно
$(10^l - 10^{l-1})/2$.)

Пример 1. Пусть *l = 100*, тогда $\bar{s} = \frac{l \cdot \ln(10)}{2} = \frac{100 \cdot \ln(10)}{2} = 115$

Важнейшие тесты по проверке простоты чисел

Все тесты делятся на *детерминированные* и *вероятностные*. Детерминированные тесты дают определенный ответ, является ли данное число простым или составным. Случайные (вероятностные) тесты дают такой же ответ, но с некоторой вероятностью (обычно близкой к 1) того, что он будет правильным.

До недавнего времени (до 2002 г.) не было известно ни одного детерминированного алгоритма с полиномиальной сложностью. В 2002 г. три индийских математика нашли такой метод [6]. Его сложность оказывается равной $O((\log n)^{12})$, хотя для специальных чисел вида $2p+1$ сложность будет значительно меньше, а именно: $O((\log n)^{6})$. Ввиду значительной сложности этого алгоритма предпочтение, однако, отдается вероятностным алгоритмам, удовлетворяющим следующему условию.

Если *n* простое, то оно всегда *проходит тест* (т. е. то, что оно простое, определяется однозначно), если же оно составное, то может случиться, что оно пройдет тест, однако вероятность такого события может быть сделана сколь угодно малой.

Рассмотрим далее два важнейших примера подобных алгоритмов тестирования чисел на простоту.

Тест Ферма

Вспомним малую теорему Ферма, которая гласит, что если n простое число и n не делит a, то $a^{n-1} = 1 \bmod n$. Поэтому необходимо выполнить следующие шаги:

1. Сгенерировать тестируемое число n и выбрать параметр «секретности» t.

2. Сгенерировать случайное число a_1 : $2 \le a_1 \le n-1$.

3. Вычислить $r = a_1^{n-1} \bmod n$.

4. Если $r \neq 1$, тогда n – составное число.

Если $r = 1$, то перейти к шагу 2 и повторить все то же самое с числом a_2 и так далее, вплоть до повторения t шагов. При получении $a_1^{n-1} = 1 \bmod n$, $a_2^{n-1} = 1 \bmod n$, …, $a_r^{n-1} = 1 \bmod n$, считать n простым числом.

Данный тест может привести к ошибке, когда на всех шагах это условие выполняется, но число n тем не менее является составным.

Пример 2. Если $n = 341 = 11 \cdot 31$, то легко проверить, что

$2^{340} \bmod 341 = 1$.

Случай, когда для любых чисел $a_1, a_2, \ldots, a_t$ составное число n проходит тест, является особым. Такие числа n называются *числами Кармайкла* при условии, что $gcd\ (a, n) = 1$. Наименьшее число Кармайкла – это число $n = 561 = 3 \cdot 11 \cdot 17$. Числа Кармайкла встречаются, однако, довольно редко.

Всего имеется 2163 числа Кармайкла в диапазоне 1 до $25 \cdot 10^9$, а в диапазоне 1 до $1 \cdot 10^5$ всего 16 таких чисел: 561,1105,1729….75361. В тесте Ферма эти числа не различимы.

Утверждение 5. При использовании теста Ферма, если число n не является числом Кармайкла, вероятность ошибки тестирования будет равна 2^{-t} , где t – число шагов.

Таким образом, выбирая параметр «секретности» t достаточно большим, можно обеспечить высокую надежность тестирования простых чисел.

Тест Миллера–Рабина.

Пусть заданы тестируемое нечетное число n и параметр «секретности» t. Данный тест базируется на утверждении, доказываемом в теории чисел [2, 3].

Утверждение 6. *Пусть* n *нечетное простое число и пусть для него справедливо представление:* $n - 1 = 2^s \cdot r$ *, где* s, r *– числа, причем* r *– нечетное. Пусть* a *– такое, что* $\gcd(a, n) = 1$, *тогда:* $a^r = 1 \bmod n$ *или*

$$a^{2^j \cdot r} = -1 \bmod n \quad \text{, где} \quad 0 \le j \le s-1$$

Тест Миллера–Рабина представляет комбинацию двух тестов:

проверки квадратным корнем и теста Ферма

Тест проверка квадратным корнем

В модульной арифметике, если n –простое число, то квадратный корень из единицы $\sqrt{1}$modn = +1 или -1. Если n составное число, то квадратный корень может быть +1, -1 и другие числа.

(Напомним, что в модульной арифметике

-1=n-1(modn)

Примеры

n=7, x=1 найти $\sqrt{x}$mod7=?

Перебором находим

1^2=1mod7

2^2=4mod7

3^2=2mod7

-1^2=1mod7= 6^2=1mod7

-2^2=4mod 7

-3^2=2mod7

$\sqrt[2]{1}$ mod7=1 или -1

n=8, x=1 найти $\sqrt[2]{1}\,\mathrm{mod}8 =?$

Перебором находим

1^2=1mod8

2^2=4mod8

3^2=1mod8

4^2=0mod8

5^2=1mod8

-1^2=1mod8= 7^2=1mod8

корни: +1,-1, 3,5,

n=22, x=1 найти $\sqrt[2]{1}\,\mathrm{mod}22 =?$

Перебором находим

1^2=1mod22

-1^2=1mod22

И все других корней нет.

Тест Миллера-Рабина - комбинация теста Ферма и квадратного корня

- Запишем n-1= m2^k
- Тест Ферма при основании a можно записать a^{n-1}=a^{m2^k}=$(a^m)^{2^k}$=

$$=((((a^m)^2)^2\ldots)^2$$

k раз

Идея алгоритма РМ

1шаг. Выбрать *a* такое, что НОД(*a,n)=1*

Проверить a^m=$\begin{Bmatrix} T = \pm 1, n - \text{возможно простое, шаг1} \\ T \neq \pm 1, \text{перейти к шагу 2} \end{Bmatrix}$

2 шаг Положить i=1. Найти $(T)^{2^i}$

$(T)^{2^i}$=$\begin{Bmatrix} = 1, n - \text{составное, поскольку корень из 1} \\ \text{может быть только 1 или } -1, \text{но не T} \\ = -1, n - \text{возможно простое, шаг1} \\ (\text{потому что при следующем возведении} \\ \text{в квадрат, будет 1} \\ \neq \pm 1, \text{перейти к шагу 2, положив } i = i + 1 \end{Bmatrix}$

Когда i=k-1, перейти к шагу 1 , выбрав новое *a.*

Итог

1. Представить $n - 1$ в виде $2^s \cdot r$, где r – нечетное число.

2. Сгенерировать случайное число a, такое что $2 \le a \le n-1$.

3. Вычислить $y = a^r \mod n$:

а) если $y = \pm 1$, то n прошло тест и возможно является простым (повторяем этот тест для другого случайно выбранного числа a);

б) если $y \neq \pm 1$, то вычисляются $y^2 \mod n$, $y^4 \mod n$, y^{2j} для $j < s$ до тех пор, пока не получится -1 для некоторого j. Если такое событие происходит, повторить тест для следующего a.

4. Если ни при каких j не выполняется шаг 3б, то число n – составное и отбрасывается как не прошедшее тест.

.

Доказывается [2, 3], что вероятность ошибки при использовании теста Миллера–Рабина аппроксимируется величиной $1/4^t$. Видно, что этот показатель значительно лучше, чем для теста Ферма, и все операции, необходимые для проведения этого теста, имеют полиномиальную сложность.

Полиномиальный тест AKS

- Предложен в 2002г. индийскими математиками Agrawal M., Kayal N., Saxena N.
- Центральная идея опирается на следующий факт. Натуральное n при условии НОД(a,n)=1, является простым в том случае, когда

$$(x-a)^n = (x^n - a)\text{modn} \qquad (1)$$

$$(x-a)^n = \sum_{i=0}^{n} C_n^k x^k (-a)^{n-k} \bmod n = (x^n - a^n) mod n$$

$$= (x^n - a \cdot a^{n-1}) \bmod n = (x^n - a \cdot 1) \bmod n$$

- Для уменьшения трудоемкости вычислений выражение (1) делят на многочлен $x^r - 1$ и находят остатки.
- $$(x-a)^n = (x^n - a)\text{modn, mod}(x^r - 1)).), (2)$$

Теорема 2. Пусть натуральное n и простое r таковы, что

i) порядок n в группе Z_r больше $(logn)^2$,

Ii) n не делится на простые числа меньшие r,

Iii) тождество (2) выполняется для всех $a \in \left[1, \sqrt{r}\ \ log_2 n\right]$,

Тогда n – степень простого числа.

Алгоритм AKS

1. Делимость n на числа от 2 до $\lfloor (logn)^5 \rfloor$проверяется в лоб;

2. Ищется r$\leq \lfloor (logn)^5 \rfloor$, для которого выполняется (i)в теореме 2;

3. Проверяется (iii);

4. Проверяется не извлекается ли из n целый корень.

Доказано, что сложность алгоритма довольно высока $\sim (logn)^{7,5}$ поэтому практической ценности алгоритм пока не имеет.