НАО «Восточно-Казахстанский технический университет имени Д. Серикбаева»

Дисциплина «Теоретико-числовые методы в криптографии»

ПРАКТИЧЕСКАЯ РАБОТА 11

Символы Лежандра и Якоби

Методические указания и задание к выполнению

Тема 11. Символы Лежандра и Якоби. Определение разрешимости сравнений второй степени. **Объём:** 3 академических часа.

Работа учит сначала устанавливать существование корней, а затем находить и проверять их. Основное различие: для нечётного простого модуля символ Лежандра полностью определяет разрешимость $x^2 \equiv a$; для составного модуля одного символа Якоби обычно недостаточно. Сначала изучите разделы 1–5 с методикой и примерами, затем выполните задания своего варианта.

1 Цель и организация работы

Нужно научиться вычислять оба символа, применять критерий Эйлера и закон взаимности, решать квадратное сравнение через дискриминант, учитывать общие делители правой части и модуля и проверять математический вывод программой.

Предварительно нужны модульное возведение в степень, обратный элемент, разложение на простые множители, КТО и поднятие корней. Для программных примеров достаточно Python 3.8 или новее; внешние пакеты не требуются. Все функции ниже следует последовательно поместить в один файл. Числа в индивидуальных вариантах малы, поэтому полный перебор используется как независимая проверка ответа.

Рекомендуемое распределение 135 минут: 35 минут на методику и примеры, 25 минут на разбор программ, 60 минут на индивидуальные задания и 15 минут на проверку и защиту.

2 Символ Лежандра и квадратное сравнение

2.1 Определение и критерий Эйлера

Пусть p — нечётное простое число. Используем расширенное определение символа Лежандра:

$$\left(\frac{a}{p}\right) = \begin{cases} 0, & p \mid a, \\ 1, & p \nmid a \text{ и } a \text{ является квадратом по модулю } p, \\ -1, & a \text{ не является квадратом по модулю } p. \end{cases}$$

Запись со скобками обозначает символ, а не деление. В некоторых учебниках его вводят только для $p \nmid a$; значение 0 добавлено здесь, чтобы алгоритм обрабатывал все входные данные.

Число корней сравнения $x^2 \equiv a \pmod p$ равно $1 + (a/p)$. При $a \equiv 0$ имеется один корень 0; ненулевой квадрат имеет два корня r и $-r$, которые различны при нечётном p; неквадрат не имеет корней.

Критерий Эйлера:

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

Для $p \nmid a$ это следует из цикличности группы ненулевых вычетов. Если $a = g^j$ для первообразного корня g, то a является квадратом ровно при чётном j, а $a^{(p-1)/2} = (-1)^j$ в поле. При $p \mid a$ обе части равны нулю. В вычислениях остаток $p - 1$ означает символ -1, а не значение символа $p - 1$.

2.2 Правила ручного вычисления

Символ зависит только от остатка числителя и мультипликативен:

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right).$$

Квадрат ненулевого вычета даёт символ $+1$. Дополнительные законы позволяют быстро обрабатывать множители -1 и 2:

$$\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2}, \qquad \left(\frac{2}{p}\right) = (-1)^{(p^2-1)/8}.$$

Второй символ равен $+1$ при $p \equiv 1,7 \pmod 8$ и -1 при $p \equiv 3,5 \pmod 8$. Для различных нечётных простых p и q квадратичный закон взаимности имеет вид

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{((p-1)/2)((q-1)/2)}.$$

При перестановке числителя и знаменателя знак меняется только тогда, когда оба числа сравнимы с 3 по модулю 4. В этой работе закон используется как теорема; его теоретическое изучение предусмотрено лекцией 12.

Пример 1. Вычислим $(5/19)$. Число 5 сравнимо с 1 по модулю 4, поэтому $(5/19) = (19/5) = (4/5) = 1$. Следовательно, сравнение имеет два корня. Критерий Эйлера подтверждает вывод: $5^9 \equiv 1 \pmod{19}$. Для сравнения, $(3/19) = -(19/3) = -(1/3) = -1$, поэтому квадратных корней из 3 по модулю 19 нет.

2.3 Нахождение корней при простом модуле

Если $p \equiv 3 \pmod 4$ и $(a/p) = 1$, один корень вычисляется без перебора:

$$r \equiv a^{(p+1)/4} \pmod p.$$

Действительно, $r^2 \equiv a^{(p+1)/2} = a \cdot a^{(p-1)/2} \equiv a$. Второй корень равен $p - r$. Если символ равен 0, ответом служит только 0; если он равен -1, корней нет. Формулу нельзя переносить на простые $p \equiv 1 \pmod 4$.

В примере 1 получаем $r = 5^5 \bmod 19 = 9$. Полный ответ: 9 и 10, поскольку $9^2 \equiv 10^2 \equiv 5 \pmod{19}$.

2.4 Общее квадратное сравнение

Рассмотрим $Ax^2 + Bx + C \equiv 0 \pmod p$ при $A \not\equiv 0 \pmod p$. После умножения на $4A$ получаем эквивалентное сравнение

$$(2Ax + B)^2 \equiv \Delta \pmod p, \qquad \Delta = B^2 - 4AC.$$

Переход между x и $2Ax + B$ взаимно однозначен, так как $2A$ обратимо. Поэтому число корней равно $1 + (\Delta/p)$, а при найденных квадратных корнях r дискриминанта решение задаётся формулой

$$x \equiv (-B + r)(2A)^{-1} \pmod p.$$

Пример 2. Решим $3x^2 + 4x - 1 \equiv 0 \pmod{19}$. Дискриминант равен $28 \equiv 9$, его корни — 3 и 16. Обратный элемент к 6 равен 16. Получаем $x \equiv (-4 + 3)16 \equiv 3$ и $x \equiv (-4 + 16)16 \equiv 2$. Проверка подстановкой даёт 19 и 38, кратные модулю.

Вырожденный случай. Если $A \equiv 0$, сначала решают линейное сравнение $Bx + C \equiv 0$. При $B \not\equiv 0$ корень единственный; при $B \equiv 0$ и $C \equiv 0$ подходят все p классов; при $B \equiv 0$ и $C \not\equiv 0$ решений нет. Формулу через обратный к $2A$ в этих случаях применять нельзя.

3 Символ Якоби и границы его применения

3.1 Определение и выводы о разрешимости

Пусть $n > 1$ — нечётное число с разложением $n = \prod p_i^{e_i}$. Символ Якоби определяется через символы Лежандра:

$$\left(\frac{a}{n}\right) = \prod_i \left(\frac{a}{p_i}\right)^{e_i}.$$

Для $n = 1$ принимают $(a/1) = 1$ как значение пустого произведения. Если $\gcd(a, n) > 1$, символ равен 0. При простом n он совпадает с символом Лежандра.

Если $\gcd(a, n) = 1$ и a является квадратом по модулю n, то каждый местный символ равен $+1$, следовательно, символ Якоби тоже равен $+1$. Поэтому значение -1 исключает разрешимость. Обратный вывод при $+1$ неверен: отрицательные множители могут сократиться в произведении, а чётная степень простого множителя вообще скрывает его знак.

Пример 3. $(2/15) = (2/3)(2/5) = (-1)(-1) = 1$, однако $x^2 \equiv 2$ неразрешимо уже по модулю 3. Напротив, $(4/15) = 1$, и сравнение имеет четыре корня: 2, 7, 8 и 13. Ещё один источник ложного вывода — квадрат знаменателя: $(2/9) = (2/3)^2 = 1$, но корня по модулю 9 нет.

Значение 0 также не определяет ответ: $x^2 \equiv 9 \pmod{15}$ имеет корни 3 и 12, а $x^2 \equiv 3 \pmod{15}$ не имеет корней из-за неразрешимости по модулю 5. Оба символа Якоби равны 0.

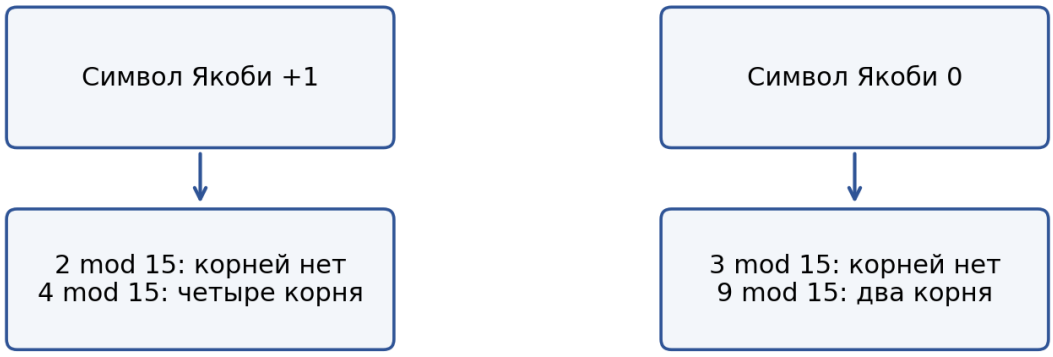


Рисунок 1 — Одинаковое значение символа Якоби может соответствовать разной разрешимости

Для составного n критерий Эйлера нельзя просто переписать с символом Якоби: например, $2^7 \equiv 8 \pmod{15}$, хотя $(2/15) = 1$.

3.2 Вычисление без разложения знаменателя

Для нечётного положительного знаменателя символ Якоби сохраняет мультипликативность и оба дополнительных закона с заменой p на n. Для положительных нечётных взаимно простых a, n действует закон взаимности с тем же правилом знака. В алгоритме общий делитель может существовать: тогда он сохраняется при преобразованиях, и итоговый символ будет нулём.

Алгоритм работает с текущим числителем a, нечётным знаменателем n и накопленным знаком s, первоначально равным 1.

1. Заменить a его остатком по модулю n.
2. Пока a положительно, выделить все множители 2. При удалении каждого из них менять знак, если $n \equiv 3$ или $5 \pmod 8$.
3. Для оставшегося нечётного a изменить знак, если одновременно $a \equiv n \equiv 3 \pmod 4$.

4. Выполнить перестановку с редукцией: $(a, n) \leftarrow (n \bmod a, a)$ и повторить шаг 2.
5. Когда $a = 0$, вернуть s, если $n = 1$, и 0 в противном случае.

Преобразования сохраняют исходный символ в виде $s(a/n)$. Знаменатель после полной итерации уменьшается, поэтому алгоритм завершается. Ненулевой конечный знаменатель больше 1 означает общий делитель; как и в алгоритме Евклида, вычислять факторизацию для этого не нужно.

Пример 4. Вычислим $(211/391)$, не раскладывая 391 на множители. При первой перестановке оба числа сравнимы с 3 по модулю 4:

$$\left(\frac{211}{391}\right) = -\left(\frac{180}{211}\right) = -\left(\frac{45}{211}\right) = -\left(\frac{31}{45}\right).$$

Далее знаменатель 45 сравним с 1 по модулю 4, а $(2/31) = 1$:

$$-\left(\frac{31}{45}\right) = -\left(\frac{14}{31}\right) = -\left(\frac{7}{31}\right) = \left(\frac{3}{7}\right) = -1.$$

Следовательно, квадратного корня из 211 по модулю 391 нет. Для вычисления символа достаточно было редукций, выделения степеней двойки и изменения знака.

4 Полный критерий по нечётному составному модулю

Если $\gcd(a, n) = 1$, сравнение $x^2 \equiv a \pmod{n}$ разрешимо тогда и только тогда, когда $(a/p) = 1$ для каждого простого делителя p числа n. Два корня по каждому p единственным образом поднимаются до его степени, а затем объединяются по КТО. Число корней равно $2^{\omega(n)}$, где $\omega(n)$ — число различных простых делителей.

Если общий делитель есть, нужно применить правило лекции 11 к каждой простой степени $q = p^e$. Приведите a по модулю q. Для нулевого остатка число местных корней равно $p^{\lfloor e/2 \rfloor}$. Для ненулевого остатка запишите $a = p^v u$, где $p \nmid u$:

- нечётный v означает отсутствие корней;
- при чётном $v = 2s$ и $(u/p) = -1$ корней также нет;
- при чётном $v = 2s$ и $(u/p) = 1$ существует $2p^s$ местных корней.

Общее число корней равно произведению местных количеств. Нулевой множитель сразу даёт неразрешимость. Это правило относится к нечётным простым p; модуль 2 и его степени здесь не рассматриваются.

Пример 5. Для $x^2 \equiv 25 \pmod{125}$ символ Якоби равен 0. Но $v_5(25) = 2$, $s = 1$ и $u = 1$, поэтому существуют десять корней. Они имеют вид $5 + 25t$ и $20 + 25t$ при $t = 0,1,2,3,4$. Для $x^2 \equiv 50 \pmod{125}$ показатель делимости тот же, однако $(2/5) = -1$, поэтому корней нет.

Чтобы вручную объединить местные корни по взаимно простым модулям m_1, m_2, используйте

$$x = r_1 + m_1 t, \qquad t \equiv (r_2 - r_1) m_1^{-1} \pmod{m_2}.$$

Переберите все пары местных корней и приведите результаты к диапазону от 0 до $m_1 m_2 - 1$. Для единственной простой степени достаточно поднятия или формулы с параметром из лекции 11.

5 Программные примеры

5.1 Символ Лежандра

Проверка простоты ниже предназначена для небольших учебных модулей. Функция `pow(a, k, m)` вычисляет модульную степень, а не полное целое число a^k.

```
from math import isqrt

def is_prime(p):
    return p >= 2 and all(p % d for d in range(2, isqrt(p) + 1))

def legendre(a, p):
    if p == 2 or not is_prime(p):
        raise ValueError("p должно быть нечётным простым")
    t = pow(a % p, (p - 1) // 2, p)
    return -1 if t == p - 1 else t
```

5.2 Символ Якоби

Следующая функция реализует описанные преобразования без факторизации. Начальная редукция также позволяет передавать отрицательный числитель. Знаменатель должен быть положительным и нечётным; при знаменателе 1 результат равен 1.

```
def jacobi(a, n):
    if n < 1 or n % 2 == 0:
        raise ValueError("n должно быть положительным нечётным")
    a, sign = a % n, 1
    while a:
        while a % 2 == 0:
            a //= 2
            if n % 8 in (3, 5):
                sign = -sign
        if a % 4 == n % 4 == 3:
            sign = -sign
        a, n = n % a, a
    return sign if n == 1 else 0
```

5.3 Корни по простому модулю

Во всех вариантах для задания 1 выбраны простые $p \equiv 3 \pmod 4$. Множество в фигурных скобках исключает повтор корня 0.

```
def sqrt_prime_3mod4(a, p):
    s = legendre(a, p)
    if p % 4 != 3:
        raise ValueError("Нужен модуль p = 3 mod 4")
    if s == -1:
        return []
    r = pow(a % p, (p + 1) // 4, p)
    return sorted({r, (-r) % p})
```

Для общего квадратного сравнения следующий код работает с любым нечётным простым модулем. После обработки линейного случая он перебирает только возможные корни дискриминанта и переводит их в корни исходного сравнения.

```
def quadratic_roots(A, B, C, p):
    legendre(1, p)  # проверка простоты и нечётности p
    A, B, C = A % p, B % p, C % p
    if A == 0:
        if B:
            return [(-C * pow(B, -1, p)) % p]
        return list(range(p)) if C == 0 else []
    delta = (B * B - 4 * A * C) % p
    # Здесь перебор только квадратных корней дискриминанта.
    roots = [r for r in range(p) if r * r % p == delta]
    inverse = pow(2 * A, -1, p)
    return sorted({(-B + r) * inverse % p for r in roots})
```

5.4 Точное число корней по составному модулю

Сначала разложим небольшой модуль. В программе ниже факторизация нужна для критерия разрешимости; функция `jacobi` в ней не нуждается.

```
def factor_small(n):
    factors, p = [], 2
    while p * p <= n:
        e = 0
        while n % p == 0:
            n //= p
            e += 1
        if e:
            factors.append((p, e))
        p += 1
    if n > 1:
        factors.append((n, 1))
    return factors
```

Вспомогательная функция `count_prime_power` предполагает, что p — нечётное простое число и $e \geq 1$. Внешняя функция передаёт ей пары из разложения нечётного модуля, обеспечивая эти условия.

```
def count_prime_power(a, p, e):
    a %= p ** e
    if a == 0:
        return p ** (e // 2)
    v = 0
    while a % p == 0:
        a //= p
        v += 1
    if v % 2 or legendre(a, p) != 1:
        return 0
    return 2 * p ** (v // 2)
```

Умножим местные количества и добавим независимый полный перебор для проверки. Перебор требует n подстановок и подходит для чисел из таблицы вариантов; для большого модуля такой способ нахождения корней непрактичен.

```
def square_root_count(a, n):
    if n < 3 or n % 2 == 0:
        raise ValueError("Нужен нечётный модуль n >= 3")
    count = 1
    for p, e in factor_small(n):
        count *= count_prime_power(a, p, e)
    return count

def square_roots_small(a, n):
    if n < 1:
        raise ValueError("n должно быть положительным")
    return [x for x in range(n) if (x * x - a) % n == 0]
```

5.5 Запуск примеров и самопроверка

После определений функций добавьте вызовы:

```
print("Лежандр:", legendre(5, 19))
print("Корни mod 19:", sqrt_prime_3mod4(5, 19))
print("Квадратное:", quadratic_roots(3, 4, -1, 19))
for a, n in [(2, 15), (4, 15), (9, 15), (25, 125)]:
    print(a, n, jacobi(a, n), square_root_count(a, n),
          square_roots_small(a, n))
```

Ожидаемый вывод

```
Лежандр: 1
Корни mod 19: [9, 10]
Квадратное: [2, 3]
2 15 1 0 []
4 15 1 4 [2, 7, 8, 13]
9 15 0 2 [3, 12]
25 125 0 10 [5, 20, 30, 45, 55, 70, 80, 95, 105, 120]
```

В каждой строке составного примера последовательно выведены a, n, символ Якоби, точное число корней и список корней. Совпадение последних двух полей по количеству является проверкой, но не заменяет обоснование критерия.

Следующий блок проверяет случаи $+1$ без корней, 0 с корнями, вырожденное сравнение и совпадение формулы количества с перебором для всех нечётных модулей от 3 до 149.

```
assert jacobi(2, 15) == 1 and square_root_count(2, 15) == 0
assert jacobi(9, 15) == 0 and square_root_count(9, 15) == 2
assert quadratic_roots(0, 0, 0, 7) == list(range(7))
assert quadratic_roots(0, 0, 1, 7) == []
for n in range(3, 150, 2):
    for a in range(n):
        assert square_root_count(a, n) == len(square_roots_small(a, n))
print("Проверки пройдены")
```

При успешном завершении последняя строка — `Проверки пройдены`.

6 Порядок выполнения и индивидуальные задания

Получите номер варианта и возьмите строку с этим номером из обеих таблиц раздела 7. Сначала выполните ручную часть, затем запустите код. Все ответы записывайте полными наборами различных остатков в стандартном диапазоне.

Задание 1. Символ Лежандра и корни. Вычислите (a/p) с помощью критерия Эйлера. Дополнительно подтвердите знак одним подходящим способом: укажите квадрат, примените дополнительный закон или закон взаимности; случай $a \equiv 0$ выделите отдельно. Определите число корней $x^2 \equiv a \pmod{p}$ и найдите их с помощью формулы для $p \equiv 3 \pmod 4$.

Задание 2. Квадратное сравнение. Решите $Ax^2 + Bx + C \equiv 0 \pmod{p}$. Сначала проверьте, сохраняется ли квадратная степень после редукции коэффициентов. Если да, найдите дискриминант, его символ Лежандра и все корни. Если нет, решите получившееся линейное или постоянное сравнение. Выполните подстановку найденных корней.

Задание 3. Символ Якоби. Вычислите (b/n) без разложения знаменателя, показывая редукции, выделение двоек и изменения знака. Запишите, какой вывод о разрешимости можно сделать только по полученному символу.

Задание 4. Полное решение по составному модулю. Разложите n на простые степени. Для каждой из них определите разрешимость $x^2 \equiv b$, местные корни и их количество. Найдите все корни по модулю n; при нескольких множителях используйте КТО и подробно покажите восстановление хотя бы для одной пары. При одной простой степени используйте поднятие или формулу с параметром. Если корней нет, укажите конкретное местное препятствие. Сопоставьте итог с символом Якоби.

Задание 5. Программная проверка. Подставьте данные своего варианта в функции из раздела 5. Сравните результаты с ручным решением; проверьте каждый корень и полноту списка полным перебором. Дополнительно приведите один пример с символом Якоби $+1$ без корней и один с символом 0 при наличии корней, не совпадающие с вашей строкой таблицы 2. Объясните оба примера через простые множители.

7 Варианты

Таблица 1 — Данные для символа Лежандра и сравнения $Ax^2 + Bx + C \equiv 0 \pmod{p}$

№	p	a	A	B	C
1	7	2	2	3	6
2	11	5	3	5	5
3	19	2	4	7	8
4	23	13	1	9	3
5	31	0	2	11	7
6	43	2	3	13	10
7	47	34	4	15	13
8	59	41	1	17	28
9	67	2	2	19	3
10	71	2	3	21	3
11	79	11	4	23	29
12	83	2	1	25	11
13	7	1	2	6	5
14	11	3	3	7	0
15	19	0	4	12	12
16	23	2	1	10	2
17	31	20	2	4	21
18	43	2	3	37	41

№	p	a	A	B	C
19	47	18	0	3	5
20	59	12	0	0	1

Таблица 2 — Данные для символа Якоби и сравнения $x^2 \equiv b \pmod{n}$

№	b	n
1	2	15
2	4	15
3	3	15
4	9	15
5	2	9
6	4	45
7	6	35
8	11	35
9	9	45
10	3	45
11	25	175
12	5	175
13	18	77
14	23	77
15	16	99
16	10	99
17	25	125
18	50	125
19	49	343
20	0	135

8 Требования к отчёту и оценивание

В отчёт включите номер варианта, данные обеих таблиц, ручное вычисление символов, решение квадратного сравнения, местные корни для составного модуля, восстановление по КТО или параметризацию, полный ответ, код и фактический вывод программы. Отдельно сформулируйте, какие выводы позволяют сделать значения символа Якоби -1, 0 и $+1$.

Таблица 3 — Критерии оценивания

Составляющая	Баллы
Символ Лежандра и корни в задании 1	15
Дискриминант или корректная обработка вырождения в задании 2	20
Вычисление символа Якоби с объяснением изменений знака	20
Местные условия, все корни и обоснование полноты в задании 4	25
Программная проверка и два объяснённых примера	15
Полнота отчёта и ответы на защите	5
Итого	100

9 Контрольные вопросы

1. Почему в определении символа Лежандра знаменатель должен быть нечётным простым?
2. Как из символа Лежандра получить точное число квадратных корней?
3. Как интерпретировать остатки 0, 1 и $p - 1$ в критерии Эйлера?
4. В каких случаях при перестановке числителя и знаменателя меняется знак?
5. Когда применима формула $a^{(p+1)/4}$ для квадратного корня?
6. Почему перед вычислением дискриминанта нужно проверить старший коэффициент?
7. Чем определения символов Лежандра и Якоби отличаются друг от друга?
8. Почему значение символа Якоби $+1$ не гарантирует существования корня?
9. Что означает нулевой символ Якоби и почему этого недостаточно для решения сравнения?
10. Почему алгоритму вычисления символа Якоби не нужна факторизация знаменателя?
11. Как показатель делимости правой части влияет на число корней по простой степени?
12. Как обосновать полноту найденного набора корней по составному модулю?

10 Литература

1. Мартынов Л. М. Алгебра и теория чисел для криптографии. 2020. §§ 7.7.1–7.7.2, с. 237–245: символы Лежандра и Якоби и их свойства. Используйте номера страниц, напечатанные в книге.
2. Trappe W., Washington L. C. Introduction to Cryptography with Coding Theory. 2nd ed. 2006. §§ 3.9–3.10, с. 86–93: квадратные корни, критерий Эйлера, символы и закон взаимности.
3. Лекция 11 данного курса: условие поднятия, число корней по нечётной простой степени и объединение решений по КТО.