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

Дисциплина «Теоретико-числовые методы в криптографии»

ПРАКТИЧЕСКАЯ РАБОТА 5

КОЛЬЦО ВЫЧЕТОВ И ОБРАТНЫЕ ЭЛЕМЕНТЫ

Методические указания и задание к выполнению

Аудиторная нагрузка: 3 академических часа. Сначала изучите определения и разобранный пример, затем выполните свой вариант. В работе используются два модуля: малый q для полных таблиц операций и основной m для вычислений обратных элементов.

1 Цель работы

Научиться выполнять операции в кольце вычетов, различать противоположный и обратный элементы, находить обратные элементы расширенным алгоритмом Евклида и обосновывать обратимость аффинного преобразования.

2 Планируемые результаты

Студент составляет таблицы сложения и умножения, определяет обратимые элементы и ненулевые делители нуля, строит тождество Безу, решает сравнение с обратимым коэффициентом и проверяет восстановление исходного класса. Ручное решение сопровождается математическим обоснованием и программной проверкой.

3 Программное обеспечение

Достаточно Python 3.8 или новее; дополнительные пакеты не требуются. Все блоки Python из раздела 5 разместите последовательно в одном файле `practical05.py`. Запуск: `python practical05.py` или `python3 practical05.py` в зависимости от настройки системы. Вывод, помеченный как ожидаемый, не является кодом программы.

Аргументы — целые числа. В этой работе модули не меньше 2: так классы нуля и единицы различны. Рассмотренный ранее вырожденный случай модуля 1 находится вне области входных данных. Отрицательные представители допустимы и предварительно заменяются неотрицательными остатками.

4 Методичка с разобранными примерами

4.1 Операции над классами

Кольцо $\mathbb{Z}_m$ состоит из классов $[0]_m, \dots, [m-1]_m$. Операции задаются формулами

$$[a]_m + [b]_m = [a+b]_m, \qquad [a]_m[b]_m = [ab]_m.$$

В таблицах вместо классов записывают их стандартные представители. Сумма чисел a, b в таблице — это $(a+b) \bmod m$, произведение — $(ab) \bmod m$. Корректность следует из свойств сравнений: выбор других представителей тех же классов не меняет ответ.

Сложение и умножение ассоциативны и коммутативны; умножение распределительно относительно сложения. Эти свойства наследуются от целых чисел. Нейтральные элементы — 0 для сложения и 1 для умножения. Поэтому получается коммутативное кольцо с единицей.

4.2 Таблицы и два вида обратного действия

Для образца возьмём $q = 6$. В каждой таблице номер строки задаёт первый аргумент, номер столбца — второй.

Таблица сложения по модулю 6

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

Таблица умножения по модулю 6

×	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

Например, в строке 4 и столбце 5 стоят $4 + 5 \equiv 3 \pmod 6$ и $4 \cdot 5 \equiv 2 \pmod 6$. Симметрия относительно главной диагонали отражает коммутативность. Каждая строка сложения содержит все классы, поскольку прибавление фиксированного числа можно отменить вычитанием.

Противоположный элемент к $[a]_m$ равен $[-a]_m$; он существует для любого класса. **Обратный по умножению** $[a]_m^{-1}$ должен удовлетворять $[a]_m[a]_m^{-1} = [1]_m$. Он существует не всегда.

Например, по модулю 6 противоположный к 2 равен 4, поскольку $2 + 4 \equiv 0$. Но обратного по умножению к 2 нет: строка 2 таблицы умножения не содержит 1. Обратимые классы по модулю 6 — только 1 и 5, причём каждый обратен самому себе.

4.3 Критерий обратимости

Теорема. Класс $[a]_m$ обратим тогда и только тогда, когда $\gcd(a, m) = 1$.

Если обратный существует, то $au \equiv 1 \pmod m$, то есть $au + mv = 1$ для некоторого целого v. Общий делитель a, m должен делить 1. Обратно, при НОД 1 тождество Безу даёт $au + mv = 1$, и класс $[u]_m$ является обратным.

Обратный класс единственен. Если $au \equiv 1$ и $av \equiv 1$, то по правилам умножения $u \equiv u(av) \equiv (ua)v \equiv v \pmod m$. Поэтому достаточно указать единственный представитель обратного в диапазоне $0 \le I < m$.

Множество обратимых представителей обозначим

$$U(m) = \{r : 1 \le r < m,\ \gcd(r, m) = 1\}.$$

Его мощность равна $\varphi(m)$. Для $m = 26$ получаем

$$U(26) = \{1,3,5,7,9,11,15,17,19,21,23,25\}.$$

Подробное вычисление функции Эйлера является темой следующей практической работы.

4.4 Расширенный алгоритм Евклида

Найдём обратный к $a = -19$ по модулю $m = 26$. Сначала нормализуем аргумент: $A = a \bmod m = 7$. Затем выполняем деления:

$$26 = 3 \cdot 7 + 5, \qquad 7 = 1 \cdot 5 + 2, \qquad 5 = 2 \cdot 2 + 1.$$

Обратная подстановка даёт

$$1 = 5 - 2 \cdot 2 = 3 \cdot 5 - 2 \cdot 7 = 3 \cdot 26 - 11 \cdot 7.$$

Следовательно, $s = -11$, $t = 3$ удовлетворяют $As + mt = 1$, а неотрицательный представитель обратного равен $I = -11 \bmod 26 = 15$. Проверки: $7 \cdot 15 = 105 \equiv 1 \pmod{26}$ и $(-19) \cdot 15 = -285 \equiv 1 \pmod{26}$.

В реализации расширенного алгоритма вместе с остатками сохраняются коэффициенты равенств $r_0 = s_0 A + t_0 m$, $r_1 = s_1 A + t_1 m$. Шаг вычитания частного из одной пары остатков выполняется одновременно над обеими парами коэффициентов. После остановки r_0 равен НОД, а s_0, t_0 дают тождество Безу. Если начать с пары (A, m) при $A < m$, первое частное равно нулю; это допустимый начальный шаг.

Нельзя проверять полученные коэффициенты для исходного отрицательного a, если алгоритм запускался для A. Сначала проверяют именно $As + mt = 1$; для исходного a достаточно отдельной проверки $aI \equiv 1 \pmod{m}$.

4.5 Делители нуля и аннулятор

Ненулевой класс $[b]_m$ называется **делителем нуля**, если существует ненулевой класс $[z]_m$ с $bz \equiv 0 \pmod{m}$. В $\mathbb{Z}_m$ это в точности ненулевые необратимые классы.

Действительно, при $g = \gcd(b, m) > 1$ число $z = m/g$ лежит между 0 и m, а $bz = m(b/g)$ делится на m. Если же b обратим, умножение $bz \equiv 0$ на b^{-1} даёт $z \equiv 0$, что противоречит выбору ненулевого класса.

Для произвольного целого b все классы, которые при умножении на b дают нуль, образуют **аннулятор** $\mathrm{Ann}(b)$. В стандартных представителях

$$\mathrm{Ann}(b) = \{0, m/g, 2m/g, \ldots, (g-1)m/g\}, \qquad g = \gcd(b, m).$$

По правилу сокращения условие $bz \equiv 0 \pmod{m}$ равносильно $z \equiv 0 \pmod{m/g}$. В интервале от 0 до $m - 1$ таких чисел ровно g. Нуль входит в аннулятор, но в этой работе сам нулевой класс не называется делителем нуля.

Пример. Для $b = 10, m = 26$ имеем $g = 2, z = 13$ и $\mathrm{Ann}(10) = \{0,13\}$. Числа 10 и 13 ненулевые, но их произведение равно нулю по модулю 26. Это сразу объясняет отсутствие обратного к 10.

Если модуль прост, все ненулевые классы обратимы, и кольцо является полем. Если $m = uv$ составное, где $1 < u, v < m$, классы u, v дают ненулевые делители нуля, поэтому поля не получается. Так $\mathbb{Z}_7$ — поле, а $\mathbb{Z}_6$ — кольцо с делителями нуля.

4.6 Деление на обратимый элемент

Если a обратим, сравнение $ax \equiv c \pmod{m}$ имеет единственное решение среди классов:

$$x \equiv a^{-1} c \pmod{m}.$$

Для образца $a = -19$, $c = 9$, $m = 26$ получаем $x \equiv 15 \cdot 9 \equiv 5 \pmod{26}$. Проверка: $(-19) \cdot 5 - 9 = -104$ делится на 26. Все целые решения имеют вид $x = 5 + 26k$, $k \in \mathbb{Z}$.

Запись «c/a по модулю m» в этом случае означает умножение на обратный класс. Обычное вещественное деление / не выполняет эту операцию. Если НОД не равен 1, данная формула

неприменима: сравнение может иметь несколько решений или не иметь ни одного. Например, $2x \equiv 2 \pmod 6$ имеет классы решений 1 и 4, а $2x \equiv 1 \pmod 6$ не имеет решений. Общая теория линейных сравнений будет рассмотрена отдельно.

4.7 Аффинное преобразование и восстановление

Для $E(x) = ax + c \pmod m$ при обратимом a обратное преобразование имеет вид

$$D(y) = a^{-1}(y - c) \pmod m.$$

Подстановка даёт $D(E(x)) \equiv a^{-1}ax \equiv x \pmod m$. Это базовая модель для разбора аффинной подстановки в криптографии.

В образце выберем $x = (c + 2) \bmod m = 11$. Тогда $E(11) = (-19 \cdot 11 + 9) \bmod 26 = 8$, а $D(8) = 15(8 - 9) \bmod 26 = 11$.

Если заменить a на необратимый $b = 10$, классы 0 и 13 дадут один результат: $E_b(0) = E_b(13) = 9$. По такому выходу нельзя однозначно восстановить вход. Это та же причина, по которой строка необратимого элемента в таблице умножения имеет повторения.

5 Программные примеры

5.1 Таблицы операций

Функция возвращает два списка строк. Индексы строк и столбцов начинаются с нуля и совпадают с представителями классов. Чтобы вывести всю таблицу, последовательно выведите каждую строку полученного списка.

```
from math import gcd

def check_modulus(m):
    if m < 2:
        raise ValueError("В этой работе требуется модуль не меньше 2")

def operation_tables(m):
    check_modulus(m)
    add = [[(i + j) % m for j in range(m)] for i in range(m)]
    mul = [[(i * j) % m for j in range(m)] for i in range(m)]
    return add, mul
```

5.2 Расширенный алгоритм

Аргументы этой функции должны удовлетворять $A \geq 0, m > 0$. В последующем вызове они получаются после проверки модуля и нормализации. Список `steps` содержит тройки «делимое, делитель, частное»; конечный НОД и коэффициенты возвращаются отдельно.

```
def extended_gcd(a, b):
    # Вход: a >= 0, b > 0.
    r0, r1 = a, b
    s0, s1 = 1, 0
    t0, t1 = 0, 1
    steps = []
    while r1 != 0:
        q = r0 // r1
        steps.append((r0, r1, q))
        r0, r1 = r1, r0 - q * r1
        s0, s1 = s1, s0 - q * s1
        t0, t1 = t1, t0 - q * t1
    return r0, s0, t0, steps
```

5.3 Поиск обратного элемента

`None` означает отсутствие обратного. Если он существует, возвращается его неотрицательный представитель. Нормализация сохраняет НОД и искомый обратный класс.

```
def inverse_mod(a, m):
    check_modulus(m)
    r = a % m
    g, s, t, steps = extended_gcd(r, m)
    if g != 1:
        return None
    return s % m
```

5.4 Классификация и аннулятор

Список `zeros` содержит только ненулевые делители нуля. Аннулятор, напротив, всегда включает нуль. При $b = 0$ он состоит из всех классов.

```
def classify_residues(m):
    check_modulus(m)
    units = [r for r in range(1, m) if gcd(r, m) == 1]
    zeros = [r for r in range(1, m) if gcd(r, m) > 1]
    return units, zeros

def annihilator(b, m):
    check_modulus(m)
    g = gcd(b, m)
    z = m // g
    return g, z, list(range(0, m, z))
```

5.5 Решение и обратное преобразование

Первая и третья функции требуют обратимости коэффициента. Прямое отображение вычисляется для любого целого коэффициента, что позволяет показать потерю однозначности на контрпримере.

```
def solve_unit(a, c, m):
    inv = inverse_mod(a, m)
    if inv is None:
        raise ValueError("Коэффициент не является обратимым")
    return inv * c % m

def affine_forward(a, c, x, m):
    check_modulus(m)
    return (a * x + c) % m

def affine_inverse(a, c, y, m):
    inv = inverse_mod(a, m)
    if inv is None:
        raise ValueError("Преобразование не является обратимым")
    return inv * (y - c) % m
```

5.6 Запуск образца

Добавьте блок после всех определений. Для своего варианта измените только первую строку. Строки таблиц в примере служат краткой проверкой; в индивидуальном отчёте нужны обе полные таблицы по q.

```
q, m, a, b, c = 6, 26, -19, 10, 9
add, mul = operation_tables(q)
print("Строка 2 сложения:", add[2])
print("Строка 2 умножения:", mul[2])
A = a % m
print("Нормализация и НОД:", A, gcd(A, m))
print("Расширенный алгоритм:", extended_gcd(A, m)[:3])
print("Обратный элемент:", inverse_mod(a, m))
units, zeros = classify_residues(m)
print("Обратимые элементы:", units)
print("Число ненулевых делителей нуля:", len(zeros))
print("Аннулятор:", annihilator(b, m))
print("Решение a*x = c:", solve_unit(a, c, m))
x = (c + 2) % m
Y = affine_forward(a, c, x, m)
print("Преобразование и восстановление:",
      x, Y, affine_inverse(a, c, Y, m))
g, z, kernel = annihilator(b, m)
print("Совпавшие образы:",
      affine_forward(b, c, 0, m), affine_forward(b, c, z, m))
```

Ожидаемый вывод

```
Строка 2 сложения: [2, 3, 4, 5, 0, 1]
Строка 2 умножения: [0, 2, 4, 0, 2, 4]
Нормализация и НОД: 7 1
Расширенный алгоритм: (1, -11, 3)
Обратный элемент: 15
Обратимые элементы: [1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25]
Число ненулевых делителей нуля: 13
Аннулятор: (2, 13, [0, 13])
Решение a*x = c: 5
Преобразование и восстановление: 11 8 11
Совпавшие образы: 9 9
```

5.7 Общие проверки

Следующие проверки добавляются после демонстрации. Их исходные числа при смене индивидуального варианта не меняют.

```
assert inverse_mod(-19, 26) == 15
assert inverse_mod(7, 26) == 15
assert inverse_mod(13, 26) is None
assert inverse_mod(0, 7) is None
assert inverse_mod(1, 2) == 1
assert classify_residues(8) == ([1, 3, 5, 7], [2, 4, 6])
assert classify_residues(7)[1] == []
assert annihilator(10, 26) == (2, 13, [0, 13])
assert annihilator(0, 5) == (5, 1, [0, 1, 2, 3, 4])
assert solve_unit(-19, 9, 26) == 5
assert affine_forward(-19, 9, 11, 26) == 8
assert affine_inverse(-19, 9, 8, 26) == 11
assert operation_tables(6)[0][4][5] == 3
assert operation_tables(6)[1][4][5] == 2
print("Все проверки пройдены")
```

Отдельно проверьте обработку ошибочного ввода: модуль меньше 2 вызывает `ValueError`; вызов `solve_unit` или `affine_inverse` с необратимым коэффициентом также вызывает `ValueError`. Функция `inverse_mod` для допустимого модуля и необратимого числа возвращает `None`.

6 Порядок выполнения работы

1. Разобрать определения, таблицы и пример с обратным элементом — 20 минут.
2. Выписать данные варианта и составить таблицы по модулю q — 25 минут.
3. Выполнить вычисления по основному модулю m и доказать результаты — 40 минут.
4. Выполнить программные проверки и сопоставить их с ручным решением — 30 минут.
5. Оформить выводы и ответить на контрольные вопросы — 20 минут. Общая продолжительность — 135 минут.

7 Индивидуальные задания

Задание 1. Таблицы по малому модулю.

По модулю q составьте полные таблицы сложения и умножения. Найдите нейтральные элементы и противоположный к каждому классу. Выпишите все обратимые элементы вместе с обратными и все ненулевые делители нуля. Обоснуйте, является ли $\mathbb{Z}_q$ полем. Не подменяйте q основным модулем m.

Задание 2. Классификация по основному модулю.

Для m выпишите $U(m)$ и множество всех ненулевых делителей нуля. Проверьте, что вместе с нулём они разбивают все m классов. Укажите число обратимых элементов и объясните его связь с $\varphi(m)$.

Задание 3. Обратный элемент и тождество Безу.

Найдите $A = a \bmod m$. Выполните расширенный алгоритм Евклида для A, m, приведите таблицу делений и коэффициенты s, t с $As + mt = 1$. Вычислите $I = s \bmod m$ и проверьте $aI \equiv 1 \pmod m$. Встроенную функцию разрешено использовать для дополнительной проверки после собственного расчёта.

Задание 4. Необратимый элемент.

Вычислите $g = \gcd(b, m)$, докажите отсутствие обратного к b. Найдите наименьший положительный z с $bz \equiv 0 \pmod m$ и выпишите все представители $\mathrm{Ann}(b)$ от 0 до $m - 1$. Обоснуйте их число и проверьте произведения.

Задание 5. Применение обратного элемента.

Решите $ax \equiv c \pmod m$ и запишите все целые решения. Для $x_* = (c + 2) \bmod m$ вычислите $y = E(x_*) = (ax_* + c) \bmod m$ и восстановите x_* по формуле D. Докажите обратимость для любого входного класса. Затем покажите совпадение образов разных классов 0 и z из задания 4 при замене коэффициента a на b.

Задание 6. Программа и проверка.

Выполните программу со своими данными и общими проверками. Выведите обе полные таблицы по q. Для каждого представителя $x = 0, \ldots, m - 1$ проверьте равенство `affine_inverse(a, c, affine_forward(a, c, x, m), m) == x`. Объясните различие между отсутствием обратного, наличием нескольких решений сравнения и ошибкой недопустимого модуля.

8 Варианты индивидуальных заданий

В каждом варианте a обратим по m, а b является ненулевым делителем нуля. Это нужно подтвердить вычислением НОД. Параметр c используется как правая часть сравнения и как сдвиг аффинного преобразования.

Вариант	q	m	a	b	c
1	5	8	-13	2	-16
2	6	9	-25	3	-9
3	7	10	-7	4	-2
4	8	12	-19	8	5
5	5	14	-39	6	12
6	6	15	-8	10	19
7	7	16	-25	12	26
8	8	18	-49	6	33
9	5	20	-13	15	40
10	6	21	-34	14	47
11	7	22	-57	12	54
12	8	24	-19	18	61
13	5	25	-43	10	68
14	6	26	-71	13	75
15	7	27	-19	18	82
16	8	28	-47	21	89
17	5	30	-79	18	96
18	6	32	-19	24	103
19	7	33	-52	22	110
20	8	35	-93	20	117

9 Требования к отчёту

Укажите дисциплину, название работы, Ф. И. О., группу, номер варианта и исходные данные. Приложите таблицы по q, классификацию по m, расширенный алгоритм с проверкой тождества Безу, аннулятор, решение сравнения, прямое и обратное преобразования, код и результаты проверок. Все конечные представители записывайте в диапазоне от 0 до соответствующего модуля минус один.

10 Критерии оценивания

Составляющая работы	Баллы
Таблицы по q и вывод о поле	20
Классификация элементов по m	15
Расширенный алгоритм и обратный элемент	25
Делитель нуля и полный аннулятор	15
Решение сравнения и аффинное восстановление	15
Код, проверки и оформление	10
Итого	100

11 Контрольные вопросы

1. Почему операции над классами не зависят от выбранных представителей?
2. Чем противоположный элемент отличается от обратного по умножению?
3. Почему ненулевой класс может не иметь обратного?
4. Как тождество Безу даёт обратный элемент?
5. Почему обратный класс единственен?
6. Для каких аргументов нужно проверять коэффициенты, полученные после нормализации?
7. Как по НОД построить ненулевой множитель, дающий произведение нуль?
8. Почему аннулятор содержит ровно $\gcd(b, m)$ классов?
9. Когда кольцо вычетов является полем?
10. Почему модульное деление не выполняется оператором $/$?
11. Почему необратимость коэффициента не всегда означает отсутствие решений линейного сравнения?
12. Что доказывает формула обратного аффинного преобразования и что показывают совпавшие образы?

Литература

Мартынов Л. М. **Алгебра и теория чисел для криптографии**. Лань, 2020. § 4.4, с. 112–114: кольцо вычетов и критерий обратимости; §§ 3.1–3.2: тождество Безу и алгоритм Евклида. Trappe W., Washington L. C. **Introduction to Cryptography with Coding Theory**. 2nd ed., 2006. § 3.3, с. 71–75: таблицы операций, деление и вычисление обратных элементов.