

Содержание

| | |
|-----------------------|----|
| 1. О сложности | 2 |
| 2. О решётках | 3 |
| 3. О Минковском | 8 |
| 4. О LLL | 12 |
| 5. О ACVP | 18 |
| 6. О SVP | 19 |
| 7. О Айтай | 25 |

ИСП Решётки

1. О сложности

2. О решётках

Определение 2.1 (Абелева группа)

Множество G вместе с отображением

$$G \times G \rightarrow G$$

называемым **операцией** на группе G и записываемым $g_1 + g_2 = g$, называется **абелевой группой**, если выполнены соотношения:

- $g_1 + g_2 = g_2 + g_1$ – **коммутативность**
- $g_1 + (g_2 + g_3) = (g_1 + g_2) + g_3$ – **ассоциативность**
- Существует такой элемент $0 \in G$, что для всех $g \in G$ выполняется равенство $g + 0 = g$ – **существование нейтрального элемента**
- Для любого $g \in G$ существует $-g \in G$, для которого выполнено соотношение $g + (-g) = 0$ – **существование обратного элемента**



Определение 2.2. Абелева группа

Определение 2.2 (Кольцо)

Множество A с двумя операциями $+: A \times A \rightarrow A$ и $\times: A \times A \rightarrow A$ называется **кольцом**, если A абелева группа относительно операции $+$ и выполняются следующие условия

- **Ассоциативность:**

$$\forall a, b, c \in A : (ab)c = a(bc)$$

- **Дистрибутивность:**

$$\forall a, b, c \in A : (a + b)c = ac + bc; c(a + b) = ca + cb$$

Также, если $\forall a, b \in A$ выполняется $ab = ba$, то такое кольцо называется **коммутативным**.

Если существует элемент 1 , такой, что $1 \cdot a = a \cdot 1$, то такое кольцо называется **кольцом с единицей**.



Определение 2.3. Кольцо

Определение 2.3 (А-модуль)

Пусть A – кольцо. Абелева группа G называется **А-модулем**, если определена операция умножения $A \times G \rightarrow G$, для которой выполняются условия:

- **Ассоциативность:**

$$\forall a, b \in A : \forall g \in G : (ab)g = a(bg)$$

- **Дистрибутивность:**

$$\forall a, b \in A : \forall g \in G : (a + b)g = ag + bg$$

$$\forall a \in A : \forall g_1, g_2 \in G : a(g_1 + g_2) = ag_1 + ag_2$$

**Определение 2.4. А-модуль****Определение 2.4 (Система образующих)**

Множество M элементов аддитивной абелевой группы G называется **системой образующих** этой группы, рассматриваемой как \mathbb{Z} -модуль, если любой её элемент α можно представить в виде

$$\alpha = c_1 \alpha_1 + \dots + c_n \alpha_n \quad c_i \in \mathbb{Z}, \alpha_i \in M$$

Система образующих называется **базисом**, если такое представление единственно

**Определение 2.5. Система образующих****Определение 2.5 (Элемент конечного порядка)**

Элемент $a \neq 0$ аддитивной абелевой группы M называется **элементом конечного порядка**, если при некотором $c \in \mathbb{Z}, c \neq 0$:

$$\underbrace{a + \dots + a}_c = 0$$

Принято считать, что 0 также элемент конечного порядка

**Определение 2.6. Элемент конечного порядка****Теорема 2.6**

Если абелева группа без элементов конечного порядка имеет конечную систему образующих, то она имеет и базис.

Число элементов базиса является инвариантом группы.

**Теорема 2.7.**

Доказательство. Пусть $\alpha_1, \dots, \alpha_n$ – некоторая конечная система образующих.

Заметим, что при замене одной из образующих на новую, полученную добавлением к ней другой образующей, умноженной на произвольное целое число, снова получится система образующих.

Действительно, пусть $\alpha'_1 = \alpha_1 + k\alpha_2$. Тогда для любого $\alpha \in M$ имеем

$$\alpha = c_1\alpha_1 + \dots + c_n\alpha_n = c_1\alpha'_1 + (c_2 - kc_1)\alpha_2 + \dots + c_n\alpha_n$$

Если элементы $\alpha_1, \dots, \alpha_n$ линейно независимы, то они образуют базис M . Пусть они линейно зависимы, тогда существует ненулевая последовательность коэффициентов c_1, \dots, c_n разложения нуля.

Выберем среди ненулевых элементов коэффициент c_i с наименьшим абсолютным значением. БОО, можно считать, что это c_1 .

Пусть не все коэффициенты c_i делятся на c_1 , тогда $c_2 = c_1q + c'$, где $0 < c' < |c_1|$.

Перейдём к новой системе образующих, где $\alpha'_1 = \alpha_1 + q\alpha_2$. Тогда

$$c_1\alpha'_1 + c'\alpha_2 + \dots + c_n\alpha_n = 0$$

Продолжим данную процедуру до тех пор пока через конечное число шагов не получим соотношение

$$k_1\beta_1 + k_2\beta_2 + \dots + k_n\beta_n = 0$$

с целыми коэффициентами k_i , в котором один из коэффициентов, БОО k_1 , является делителем остальных. Сократив на k_1 , получим


$$\beta_1 + l_2\beta_2 + \dots + l_n\beta_n = 0$$

с целыми l_2, \dots, l_n . Следовательно, β_2, \dots, β_n – система образующих группы M , состоящая из $n - 1$ элемента.

Теперь мы можем применять этот алгоритм снова и получим либо базис, либо новую систему образующих с меньшим количеством элементов. Повторив эту процедуру конечное число раз, получим базис группы.

Инвариантность числа элементов базиса M следует из инвариантности размерности векторного пространства $M \otimes \mathbb{Q}$, в которое M вложено. \square

Следствие 2.6.1 (Свойства базисов абелевых групп)

Пусть $\omega_1, \dots, \omega_m$ и $\omega'_1, \dots, \omega'_m$ – два базиса модуля M . Тогда матрица перехода одного базиса в другой – целочисленна, порядка m с определителем единица. 

Следствие 2.6.2. Свойства базисов абелевых групп

Определение 2.7 (Ранг абелевой группы)

Максимальное количество линейно независимых элементов абелевой группы называется её **рангом** 

Определение 2.8. Ранг абелевой группы

Определение 2.8 (Решётка)

Решёткой называется подгруппа группы \mathbb{R}^n , порождённая системой линейно независимых над \mathbb{R} векторов-столбцов

$$b_1, \dots, b_n \in \mathbb{R}^n$$

Если $m = n$, то решётка называется **полной**, в противном случае – **неполной**. Базис группы в этом случае называется базисом решётки.

Набор базисных векторов-столбцов задаёт матрицу

$$B = [b_1 \mid \dots \mid b_m]$$

Матрица B называется матрицей, **соответствующей** решётки.

**Определение 2.9. Решётка****Определение 2.9**

Пусть b_1, \dots, b_m – базис решётки Λ в \mathbb{Z}^n .

Основным параллелепипедом этой решётки называется множество

$$T = T(\Lambda) = \{x \in \mathbb{R}^n \mid x = \alpha_1 b_1 + \dots + \alpha_m b_m \mid 0 \leq \alpha_i < 1\}$$

Детерминантом решётки Λ называется m -мерный объём этого множества и обозначается через $\det(\Lambda)$.

**Определение 2.10.****Теорема 2.10 (Критерий полноты решётки)**

Решётка M в линейном пространстве L полна тогда и только тогда, когда в L существует ограниченное множество U , сдвиги которого на векторы из M полностью заполняют всё пространство L .

**Теорема 2.11. Критерий полноты решётки**

Доказательство. Если решётка Λ полная, то в качестве U можно взять любой её основной параллелепипед.

Пусть теперь решётка Λ неполная, и пусть U – произвольное ограниченное подмножество в \mathbb{R}^n .

Тогда существует такое $r > 0$, что

$$\forall x \in U : \|x\| < r$$

Пусть $L_0 \subset \mathbb{R}^n$ – подпространство, порождённое решёткой Λ . Поскольку решётка неполная, то L_0 – собственное подпространство и, следовательно, существует вектор $y \in \mathbb{R}^n$, имеющий длину больше r и ортогональный подпространству L_0 .

Покажем, что y не покрывается сдвигами множества U .


Пусть это не так, тогда при некоторых $u \in U, z \in \Lambda$ выполняется равенство $y = u + z$. Тогда, согласно неравенству Коши-Буняковского

$$\|y\|^2 = (y, y) = (y, u) \leq \|y\| \|u\| < r \|y\|$$

откуда $\|y\| < r$ – противоречие. □

3. О Минковском

Определение 3.1 (Дискретная группа)

Подгруппа G группы \mathbb{R}^n называется **дискретной**, если в шаре $U(r) = \{x \in \mathbb{R}^n \mid \|x\| < r\}$ радиуса r имеется только конечное число элементов группы G . 

Определение 3.2. Дискретная группа


Лемма 3.2

Решётка является дискретной группой 

Лемма 3.3.

Лемма 3.3 (О разбиении на сдвиги решётки)

Если T – основной параллелепипед полной решётки M , то имеется разбиение


$$\mathbb{R}^n = \bigsqcup_{z \in M} (T + z)$$


Лемма 3.4. О разбиении на сдвиги решётки

Лемма 3.4

Пусть M – решетка. $U(r)$ – шар радиуса r .

Тогда

$$\forall r > 0 : N = \{z \in M \mid (z + T) \cap U(r) \neq \emptyset\} \text{ – конечно}$$


Лемма 3.5.

Доказательство. Пусть b_1, \dots, b_n – базис решётки M . Положим


$$d = \|b_1\| + \dots + \|b_n\|$$

Пусть $x = z + t \in U(r)$, где $z \in M$ и $t \in T$. Тогда

$$\|t\| = \|\alpha_1 b_1 + \dots + \alpha_n b_n\| \leq \alpha_1 \|b_1\| + \dots + \alpha_n \|b_n\| < d$$

и

$$\|z\| = \|x - t\| \leq \|x\| + \|t\| < r + d$$

то есть множество N лежит в шаре радиуса $r + d$ и согласно Лемма 3.5 это множество конечно. 

Лемма 3.5 (Минковского о выпуклом теле)

Пусть в n -мерном пространстве \mathbb{R}^n заданы полная решётка M , детерминант которой равен Δ и ограниченное центрально симметричное выпуклое множество X с объёмом $\mu(X)$.

Если $\mu(X) > 2^n \Delta$, то множество X содержит по крайней мере одну отличную от нуля точку решётки M .

**Лемма 3.6. Минковского о выпуклом теле**

Доказательство. Докажем вначале, что если множество $Y \subset \mathbb{R}^n$ таково, что все его сдвиги $Y_z = Y + z$ на векторы z из решётки M не пересекаются, то $\mu(Y) \leq \Delta$.

Рассмотрим основной параллелепипед T решётки M и рассмотрим пересечение $Y \cap T_{-z}$. Тогда по Лемма 3.6:

$$\mu(Y) = \sum_{z \in M} \mu(Y \cap T_{-z})$$

причём по Лемма 3.6, в этой сумме только конечное число слагаемых не равно нулю.

Сдвиг множества $Y \cap T_{-z}$ на вектор z равен $Y_z \cap T$, причём их объёмы будут совпадать. Следовательно

$$\mu(Y) = \sum_{z \in M} \mu(Y_z \cap T)$$

Поскольку все Y_z попарно не пересекаются, то сумма правой части не больше $\mu(T)$, что и требовалось доказать.

Рассмотрим теперь множество $\frac{1}{2}X$. Тогда из условия теоремы следует, что

$$\mu\left(\frac{1}{2}X\right) = \frac{1}{2^n} \mu(X) > \Delta.$$

Если все сдвиги множества $\frac{1}{2}X$ на элементы решётки попарно не пересекаются, то по доказанному выше должно выполняться неравенство

$$\mu\left(\frac{1}{2}X\right) \leq \Delta$$

Что противоречит условию теоремы. Значит,

$$\exists z_1, z_2 \in M : \left(\frac{1}{2}X + z_1\right) \cap \left(\frac{1}{2}X + z_2\right) \neq \emptyset$$

то есть

$$\exists x', x'' \in X : \frac{1}{2}x' + z_1 = \frac{1}{2}x'' + z_2$$

Тогда

$$z_1 - z_2 = \frac{1}{2}x'' - \frac{1}{2}x' = \frac{1}{2}x'' + \frac{1}{2}(-x')$$

Поскольку множество X центрально симметрично и выпукло, то разность $z_1 - z_2 \in M$ лежит также и в X . □

Теорема 3.6 (Неравенство Адамара)

Пусть $\det(\Lambda)$ – детерминант решётки и b_1, \dots, b_n – её базис.

Тогда справедливо неравенство

$$\det(\Lambda) \leq \|b_1\| \cdot \dots \cdot \|b_n\|$$

где $\|\cdot\|$ – евклидова норма, то есть $\|x\| = \sqrt{x^T x}$

**Теорема 3.7. Неравенство Адамара**

Доказательство. Пусть b_1, \dots, b_n – базис решётки. Рассмотрим процедуру ортогонализации базиса

$$b_1^* = b_1; b_2^* = b_2 - \frac{(b_1, b_2)}{(b_1^*, b_1^*)} b_1^*; \dots; b_n^* = b_n - \sum_{k=1}^{n-1} \frac{(b_n, b_k^*)}{(b_k^*, b_k^*)} b_k^*$$

Тогда

$$\begin{aligned} \|b_k^*\|^2 &= \left(b_k - \sum_{i=1}^{k-1} \frac{(b_k, b_i^*)}{(b_i^*, b_i^*)} b_i^*, b_k - \sum_{i=1}^{k-1} \frac{(b_k, b_i^*)}{(b_i^*, b_i^*)} b_i^* \right) = \\ &= \|b_k\|^2 - 2 \sum_{i=1}^{k-1} \frac{(b_k, b_i^*)^2}{(b_i^*, b_i^*)} + \sum_{i=1}^{k-1} \frac{(b_k, b_i^*)^2}{(b_i^*, b_i^*)} = \\ &= \|b_k\|^2 - \sum_{i=1}^{k-1} \frac{(b_k, b_i^*)^2}{(b_i^*, b_i^*)} \leq \|b_k\|^2 \end{aligned}$$

Следовательно, выполняются неравенства $\|b_k^*\| \leq \|b_k\|$. Тогда

$$\det(\Lambda) = \|b_1^*\| \cdot \dots \cdot \|b_n^*\| \leq \|b_1\| \cdot \dots \cdot \|b_n\|$$

**Определение 3.7 (Последовательность минимумов)**

Пусть $B_m(0, r)$ – открытый шар радиуса r в пространстве \mathbb{R}^m и Λ – решётка.

Определим **последовательность минимумов** $\lambda_1, \dots, \lambda_n$ формулой

$$\lambda_i(\Lambda) = \inf\{r \mid \dim\langle \Lambda \cap B_m(0, r) \rangle \geq i\}$$

**Определение 3.8. Последовательность минимумов****Теорема 3.8 (Вторая теорема Минковского)**

Существуют независимые векторы решётки, для которых выполняется неравенство

$$\|x_1\| \cdot \dots \cdot \|x_n\| \leq \frac{2^n}{\sqrt{n}} \det(\Lambda)$$

**Теорема 3.9. Вторая теорема Минковского**

Доказательство. В силу определения последовательных минимумов для решётки, достаточно доказать неравенство

$$\lambda_1 \cdot \dots \cdot \lambda_n \leq \frac{2^n}{\sqrt{n}} \det(\Lambda)$$

Пусть x_1, \dots, x_n – линейно независимые векторы решётки, для которых достигаются последовательные минимумы решётки $\lambda_1, \dots, \lambda_n$ и предположим, что

$$\prod_{i=1}^n \lambda_i > \frac{2^n}{V_n} \det(\Lambda)$$

Пусть векторы x_i^* получены с помощью процедуры ортогонализации Грамма-Шмидта. Введём преобразование T :

$$T\left(\sum_{i=1}^n c_i x_i^*\right) = \sum_{i=1}^n \lambda_i c_i x_i^*$$

Пусть $S = B_n(0, 1) \cap \langle \Lambda \rangle$ – n -мерный шар в $\langle \Lambda \rangle$. Тогда

$$\mu(T(S)) = \left(\prod_{i=1}^n \lambda_i\right) \mu(S) > \frac{2^n}{V_n} \det(\Lambda) \mu(S) = 2^n \det(\Lambda)$$

Следовательно, по Лемма 3.9 в $T(S)$ имеется ненулевая точка решётки y . Следовательно, существует точка $x \in S$, для которой $T(x) = y$.

Из определения S следует, что $\|x\| < 1$. При этом

$$x = \sum_{i=1}^n c_i x_i^*; \quad y = \sum_{i=1}^n \lambda_i c_i x_i^*$$

Поскольку $y \neq 0$, то при некотором i выполняется неравенство $c_i \neq 0$.

Пусть k – максимальное значение индекса, при котором $c_k \neq 0$ и k' – минимальное значение индекса, при котором $\lambda_{k'} = \lambda_k$.

Отметим, что элемент y линейно независим от $x_1, \dots, x_{k'-1}$, поскольку $(x_k^*, y) = \lambda_k c_k \|x_k^*\| \neq 0$ и элемент x_k^* ортогонален $x_1, \dots, x_{k'-1}$.

Покажем теперь, что $\|y\| \leq \lambda_k$. Действительно

$$\begin{aligned} \|y\|^2 &\stackrel{\substack{= \\ \forall i > k: c_i = 0}}{=} \left\| \sum_{i \leq k} \lambda_i c_i x_i^* \right\|^2 = \sum_{i \leq k} \lambda_i^2 c_i^2 \|x_i^*\|^2 \leq \sum_{i \leq k} \lambda_k^2 c_i^2 \|x_i^*\|^2 = \\ &\lambda_k^2 \left\| \sum_{i \leq k} c_i x_i^* \right\|^2 = \lambda_k^2 \|x\|^2 < \lambda_k^2 \end{aligned}$$

Полученное неравенство противоречит определению $\lambda_{k'}$. □

Следствие 3.8.1 (Оценка длины кратчайшего вектора)

Для первого минимума λ_1 выполняется неравенство

$$\lambda_1 \leq \frac{2}{\sqrt[n]{V_n}} \sqrt[n]{\det(\Lambda)}$$



Следствие 3.8.2. Оценка длины кратчайшего вектора

4. O LLL

Определение 4.1 (SVP)

Задачей нахождения кратчайшего вектора решётки будем именовать **SVP (Shortest Vector Problem)**

По заданному базису $B \in \mathbb{Z}^{m \times n}$ найти ненулевой вектор Bx , где $x \in \mathbb{Z}^n \setminus \{0\}$, такой, что

$$\forall y \in \mathbb{Z}^n \setminus \{0\} : \|Bx\| \leq \|By\|$$



Определение 4.2. SVP

Определение 4.2 (CVP)

Задачей нахождения ближайшего вектора решётки будем именовать **CVP (Closest Vector Problem)**.

По заданному базису $B \in \mathbb{Z}^{m \times n}$ и вектору-цели $t \in \mathbb{Z}^m$ найти вектор решётки Bx , такой, что

$$\forall y \in \mathbb{Z}^n : \|Bx - t\| \leq \|By - t\|$$



Определение 4.3. CVP

Определение 4.3 (Приведённый базис)

Пусть a, b – базис двумерной решётки. Этот базис называется **приведённым** относительно нормы $\|\cdot\|$, если выполняются неравенства

$$\|a\|, \|b\| \leq \|a + b\|, \|a - b\|$$



Определение 4.4. Приведённый базис

Определение 4.4 (Вполне упорядоченный базис)

Базис двумерной решётки a, b называется **вполне упорядоченным**, если выполняются неравенства

$$\|a\| \leq \|a - b\| < \|b\|$$



Определение 4.5. Вполне упорядоченный базис

Теорема 4.5 (Критерий приведённости)

Пусть a, b – базис двумерной решётки и λ_1, λ_2 последовательные минимумы решётки.

Тогда базис a, b приведён тогда и только тогда, когда нормы векторов a и b равны значениям λ_1, λ_2 соответственно.



Теорема 4.6. Критерий приведённости

Определение 4.6 (Обобщённый алгоритм Гаусса)

Вначале определим операцию $\text{find}(a, b)$:

$$\mu \in \mathbb{Z} : \forall \mu' \in \mathbb{Z} : \|b - \mu a\| \leq \|b - \mu' a\|$$

Теперь рассмотрим сам алгоритм:

Вход: произвольный базис двумерной решётки (a, b)

Выход: приведённый базис

```

if norm(a) > norm(b)
  let (a, b) = (b, a);
if norm(a - b) > norm(a + b)
  let b = -b;
if norm(b) <= norm(a - b)
  return (a, b);
if norm(a) <= norm(a - b)
  goto loop;
if norm(a) = norm(b)
  return (a, a - b);
let (a, b) = (b - a, b);
loop {
  let mu = find(a, b);
  let (a, b) = (a, b - mu * a);
  if norm(a - b) > norm(a + b)
    let b = -b;
  let (a, b) = (b, a);

  if приведённый(a, b)
    return (a, b);
}

```

**Определение 4.7. Обобщённый алгоритм Гаусса****Лемма 4.7**

В начале каждого цикла итераций в алгоритме Гаусса базис (a, b) вполне упорядочен.



Лемма 4.8.

Лемма 4.8

Рассмотрим три точки на прямой: $x, x + y, x + \alpha y$, где $\alpha \in (1, +\infty)$. Для любой нормы $\|\cdot\|$:

$$\|x\| \leq \|x + y\| \Rightarrow \|x + y\| \leq \|x + \alpha y\|$$

$$\|x\| < \|x + y\| \Rightarrow \|x + y\| < \|x + \alpha y\|$$



Лемма 4.9.

Теорема 4.9 (Полиномиальность алгоритма Гаусса)

Алгоритм Гаусса заканчивает работу за конечное число шагов. Число итераций в алгоритме Гаусса для базиса (a, b) не превосходит $2 + \log_2(\|a\| + \|b\|)$

**Теорема 4.10. Полиномиальность алгоритма Гаусса**

Доказательство. Пусть k – число итераций в алгоритме Гаусса и (a_k, a_{k+1}) – вполне упорядоченный базис в начале первой итерации.

Тогда справедлива следующая оценка:

$$\forall i \geq 3 : \|a_i\| < \frac{1}{2} \|a_{i+1}\|$$

Доказательство. Рассмотрим последовательность векторов $(a_{i-1}, a_i, a_{i+1}) = (a, b, c)$.

Тогда выполняются неравенства

$$\|a\| < \|b\| < \|c\|$$

и при некотором целом $\mu \geq 1$ и $\varepsilon = \pm 1$ выполняется равенство $a = \varepsilon(c - \mu b)$. Тогда $c = \varepsilon a + \mu b$. Докажем, что $|c| > 2|b|$:

- Пусть $\mu = 1$. Тогда выполняется неравенство $\|c - b\| = \|a\| < \|b\|$, противоречащее вполне упорядоченности базиса (b, c) . Следовательно $\mu \neq 1$
- Пусть $\varepsilon = -1, \mu = 2$. Тогда $\|c - b\| = \|-a + b\|$. Поскольку базис (a, b) вполне упорядочен, выполняется неравенство $\|a - b\| < \|b\|$ и, следовательно, $\|c - b\| < \|b\| < \|c\|$, что противоречит упорядоченности базиса (b, c) .
- Пусть $\varepsilon = -1, \mu > 2$. Тогда, учитывая неравенство $\|a\| < \|b\|$, получим

$$\|c\| = \|-a + \mu b\| \geq \mu \|b\| - \|a\| > \mu \|b\| - \|b\| = (\mu - 1)\|b\| \geq 2\|b\|$$

- Пусть $\varepsilon = 1, \mu \geq 2$. Поскольку базис (a, b) вполне упорядочен, выполняется неравенство $\|b - a\| < \|b\|$. Тогда по Лемма 4.10, выполняется неравенство $\|b\| < \|b + a\|$, а из упорядоченности базиса (a, b) следует неравенство $\|a\| \leq \|b - a\|$, поэтому $\|a\| < \|b + 1\|$.

Наконец, используя Лемма 4.10 получим

$$\|a\| \leq \|a + b\| < \|a + 2b\| \leq \|a + \mu b\| = c$$

Итак, доказано неравенство $\|c\| = \|a + \mu b\| \geq \|2b + a\|$. Для доказательства леммы достаточно проверить выполнение неравенства $\|2b + a\| > 2\|b\|$.

Используя неравенство $\|a - b\| < \|b\|$ (упорядоченность (a, b)), из неравенства треугольника получаем

$$\|2b - a\| \leq \|b\| + \|b - a\| < \|b\| + \|b\| = 2\|b\|$$

Снова воспользовавшись Лемма 4.10:

$$\|2b - a\| < \|2b\| = \|2b - a + a\| < \|2b - a + 2a\| = \|2b + a\|$$

□

Воспользовавшись леммой, получаем, что при $i \geq 3$ выполняется неравенство

$$\|a_i\| \geq 2^{i-3} \|a_3\|$$

В частности, для любых базисных векторов a, b выполняется неравенство

$$\|a\| + \|b\| \geq \|a_{k+1}\| \geq 2^{k-2} \|a_3\| \geq 2^{k-2}$$

Следовательно, $k \leq 2 + \log_2(\|a\| + \|b\|)$ □

Определение 4.10 (LLL-приведённый базис)

Базис $B = (b_1, \dots, b_n) \in \mathbb{R}^{m \times n}$ называется LLL-приведённым, относительно параметра $\frac{1}{4} < \delta < 1$, если

1. $\mu_{ij} \leq \frac{1}{2}$ при $i > j$, где μ_{ij} – коэффициенты матрицы ортогонализации Грамма-Шмидта
2. Для любой последовательной пары векторов b_i, b_{i+1} выполняется неравенство

$$\delta \|\pi_i(b_i)\|^2 \leq \|\pi_i(b_{i+1})\|^2$$

где π_i – проекция на линейную оболочку $\langle b_i^*, \dots, b_n^* \rangle$.

Иначе это условие задаётся соотношением

$$\delta \|b_i^*\|^2 \leq \|b_{i+1}^* + \mu_{i+1,i} b_i^*\|^2 = \|b_{i+1}^*\|^2 + \mu_{i+1,i}^2 \|b_i^*\|^2$$



Определение 4.11. LLL-приведённый базис

Теорема 4.11 (Свойства LLL-приведённого базиса)

Пусть b_1, \dots, b_n – LLL-приведённый базис решётки L . Тогда

1. $\det L \leq \prod_{i=1}^n \|b_i\| \leq \left(\frac{4}{4\delta-1}\right)^{\frac{n(n-1)}{4}} \det L$
2. $\|b_j\| \leq \left(\frac{4}{4\delta-1}\right)^{\frac{j-1}{2}} \|b_i^*\|$ при $1 \leq j \leq i \leq n$
3. $\|b_1\| \leq \left(\frac{4}{4\delta-1}\right)^{\frac{n-1}{4}} (\det L)^{\frac{1}{n}}$
4. Если $x \neq 0$ – элемент решётки, то $\|b_1\| \leq \left(\frac{4}{4\delta-1}\right)^{\frac{n-1}{2}} \|x\|$



Теорема 4.12. Свойства LLL-приведённого базиса

Определение 4.12 (LLL-алгоритм)

Вход: Базис решётки $B = (b_1, \dots, b_n) \in \mathbb{Z}^{m \times n}$

Выход: LLL-приведённый базис решётки

```

for  $i = 1, \dots, n$ 
  for  $j = i - 1, \dots, 1$ 
     $b_i := b_i - c_{i,j} b_j$  где  $c_{i,j} = \lfloor (b_i, b_j) / (b_j, b_j) \rfloor$ 
  if  $\delta \| \pi_i(b_i) \|^2 > \| \pi_i(b_{i+1}) \|^2$  для некоторого  $i$ 
  then  $\text{swap}(b_i, b_{i+1})$  go to (loop)
else  $B$  — ВЫХОД

```

Определение 4.13. LLL-алгоритм

Предложение 4.13

LLL-алгоритм корректен и работает за полиномиальное количество шагов

Предложение 4.14.

Доказательство. Определим целые числа d_i формулой:

$$d_i(b) = \det \begin{pmatrix} (b_1, b_1) & \dots & (b_1, b_i) \\ \dots & \ddots & \dots \\ (b_i, b_1) & \dots & (b_i, b_i) \end{pmatrix}$$

Согласно доказанному ранее об объёме основного параллелепипеда, выполняется равенство

$$d_i(b) = \prod_{j=1}^n \|b_j^*\|^2$$

Введём также обозначение

$$D(b) = \prod_{j=1}^{n-1} d_j(b)$$

Заметим, что если в процессе выполнения алгоритма не выполняется перестановка векторов, то величины d_i , являющиеся детерминантами базисов соответствующих решёток, не изменяются. Следовательно, и величина D в этом случае не изменяется.

Рассмотрим теперь шаг алгоритма, на котором выполняется перестановка двух соседних элементов базиса. А именно, пусть векторы b_1, \dots, b_i определяют LLL-приведённый базис в решётке $\langle b_1, \dots, b_i \rangle$, порождённой этими векторами.

Пусть также векторы b_1, \dots, b_{i+1} представляют базис, для которого выполняется условие 1, но не выполняется условие 2 Определения 4.14.

Тогда, согласно LLL-алгоритму, выполняется перестановка векторов b_i, b_{i+1} . Назовём новый базис \tilde{b} .

Посмотрим, как изменится при этом значение величины D . Отметим, что значения $d_k, k \neq i$ остаются неизменными. Запишем соответствующее преобразование базиса

$$(\tilde{b}_1, \dots, \tilde{b}_i) = (b_1, \dots, b_{i+1}, b_i)$$

поэтому

$$\frac{D(\tilde{b})}{D(b)} = \prod_{k=1}^n \frac{d_k(\tilde{b})}{d_k(b)} = \frac{d_i(\tilde{b})}{d_i(b)} = \frac{\|\pi_i(b_{i+1})\|^2}{\|b_i^*\|^2}$$

Поскольку выполнялась перестановка, второе условие Определение 4.14 не выполняется, то есть

$$\frac{\|\pi_i(b_{i+1})\|^2}{\|b_i^*\|^2} = \frac{\|\pi_i(b_{i+1})\|^2}{\|\pi_i(b_i)\|^2} \leq \delta$$

Поэтому выполняется неравенство

$$D(\tilde{b}) \leq \delta D(b)$$

Пусть $D_0 = D(d_1, \dots, d_n)$ – значение целозначной функции D на исходном базисе решётки на входе LLL-алгоритма, а D_k – соответствующее значение после k -й итерации.

Тогда из формулы выше следует соотношение $D_k \leq \delta^k D_0$.

Поскольку D – целозначная положительная функция и $\delta < 1$, выполняется неравенство

$$k \leq \frac{\log D_0}{\log(\frac{1}{\delta})}$$

Следовательно, если $\delta < 1$ – константа, то число итераций полиномиально от длины входа. \square

5. О ACVP

Определение 5.1 (Задача ACVP)

Пусть задан вектор $b \in \mathbb{R}^m$ и n -мерная решётка с базисом (b_1, \dots, b_n) .

Требуется найти вектор $b_0 \in \langle b_1, \dots, b_n \rangle$, для которого выполнено соотношение.

$$\|b - b_0\| \leq 2 \left(\frac{2}{\sqrt{3}} \right)^n \min_{x \in \langle b_1, \dots, b_n \rangle} \|x - b\|$$



Определение 5.2. Задача ACVP

Определение 5.2 (ACVP-алгоритм)

Вход: Базис решётки $B = (b_1, \dots, b_n) \in \mathbb{Z}^{m \times n}$ и вектор $t \in \mathbb{Q}^m$.

Выход: решение задачи ACVP.

В начале нужно выполнить LLL-алгоритм для δ_n (найти LLL-приведённый базис (p_1, \dots, p_n) решётки B).

И в качестве ответа взять вектор

$$\sum_{j=1}^n \left\lceil \frac{(t, p_j^*)}{(p_j^*, p_j^*)} \right\rceil p_j$$



Определение 5.3. ACVP-алгоритм

Теорема 5.3

При $\delta_n = \left(\frac{1}{4}\right) + \left(\frac{3}{4}\right)^{\frac{n}{n-1}}$ ACVP-алгоритм решает задачу ACVP



Теорема 5.4.

6. О SVP

Определение 6.1 (Задача поиска CVP)

Найти вектор решётки $x \in \Lambda$, минимизирующий величину $\|x - t\|$.



Определение 6.2. Задача поиска CVP

Определение 6.2 (Задача оптимизации CVP)

Найти минимум $\|x - t\|$ по всем $x \in \Lambda$



Определение 6.3. Задача оптимизации CVP

Определение 6.3 (Задача распознавания)

По заданному рациональному числу $r > 0$ определить, существует ли вектор решётки x , для которого $\|x - t\| \leq r$.



Определение 6.4. Задача распознавания

Теорема 6.4

Пусть \mathcal{A} – оракул, решающий распознавательный вариант задачи CVP. Тогда существует полиномиальный алгоритм с оракулом \mathcal{A} для решения задачи поиска для CVP.



Теорема 6.5.

Доказательство. На вход алгоритма подаются решётка $B \in \mathbb{Z}^{n \times m}$ и вектор $t \in \mathbb{Z}^n$. Требуется найти вектор $x = (x_1, \dots, x_m)$, такой, что

$$\|Bx - t\| = \min_{y \in \langle B \rangle} \|t - y\|$$

Рассмотрим решётку $B' = [2b_1, b_2, \dots, b_m]$. Заметим, что

$$L(B') \subset L(B) \Rightarrow \rho(t, \langle B' \rangle) \leq \rho(t, \langle B \rangle)$$

Чтобы проверить, выполняется ли строгое неравенство, воспользуемся оракулом \mathcal{A} .

БОО, можно считать, что $t \in \overline{B}$. Иначе, вычтем из вектора его проекцию на B , не уменьшив расстояние. Воспользовавшись бинарным поиском и оракулом \mathcal{A} найдём за полиномиальное время такое целое r , для которого выполняются неравенства

$$r < \rho(t, \langle B \rangle) \leq r + 1$$

Обратимся теперь к оракулу \mathcal{A} на входе $(B', t, \text{ratapp}(\sqrt{r+1}))$, где $\text{ratapp}(\sqrt{r})$ – рациональное число из полуинтервала $(\sqrt{r}, \sqrt{r+1}]$.

В случае ответа NO, выполняются неравенства

$$\rho(t, \langle B' \rangle) > \sqrt{r+1} \geq \rho(t, \langle B \rangle)$$

В случае же ответа YES выполняются неравенства

$$\sqrt{r} < \rho(t, \langle B \rangle) \leq \rho(t, \langle B' \rangle) \leq \sqrt{r+1}$$

и следовательно, $\rho(t, \langle B \rangle) = \rho(t, \langle B' \rangle)$, поскольку оба числа целые.

Заметим теперь, что если для некоторого ближайшего к t вектора x координата x_1 чётная, то $\rho(t, \langle B \rangle) = \rho(t, \langle B' \rangle)$, если же координата x_1 нечётна для всех ближайших векторов x , то $\rho(t, \langle B \rangle) < \rho(t, \langle B' \rangle)$.

Поэтому результат сравнения величин $\rho(t, \langle B \rangle), \rho(t, \langle B' \rangle)$ позволяет определить чётность координаты x_1 для некоторого ближайшего вектора Bx .

Теперь, зная младший бит координаты x_1 для некоторого ближайшего вектора, найдём следующие по значению биты этой координаты с помощью следующей процедуры.

Положим $t' = t - \varepsilon b_1$, где $\varepsilon = 0$ для выбора чётного бита и $\varepsilon = 1$ в случае нечётного.

Применяем теперь описанную процедуру для решётки B' и вектора t' . Отметим, что число требуемых шагов оценивается при помощи правила Крамера и полиномиально относительно размера входа (B, t) . \square

Следствие 6.4.1

Три варианта задачи CVP – полиномиально эквивалентны.



Следствие 6.4.2.

Определение 6.5 (Базовые понятия теории сложности)

Пусть имеется алфавит Σ и множество строк $\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n$.

Пусть задано подмножество $L \subset \Sigma^*$ называемое **языком**. Для заданного элемента $\sigma \in \Sigma^*$ требуется определить, принадлежит ли он этому языку.

Класс задач, для которых задача распознавания может быть решена полиномиальным алгоритмом, называется **полиномиально разрешимым классом** задач, или **классом P**.

Язык L принадлежит **классу NP**, если существует такое подмножество $R \subset \Sigma^* \times \Sigma^*$, что принадлежность $(x, y) \in R$ проверяется за полиномиальное время от длины x и для каждого $x \in L$ существует y , что $(x, y) \in R$.



Определение 6.6. Базовые понятия теории сложности

Определение 6.6 (Сводимость по Карпу)

Пусть имеются две задачи распознавания A и B . Полиномиально вычислимая функция $F : \Sigma^* \rightarrow \Sigma^*$ называется **сведением по Карпу** задачи A к задаче B , если принадлежность $x \in A$ эквивалентна условию $f(x) \in B$.

Задача распознавания A называется NP-трудной, если любая другая задача B из класса NP сводится к задаче A .

Если A находится в классе NP, то она называется NP-полной.

**Определение 6.7. Сводимость по Карпу****Определение 6.7 (Задача о рюкзаке)**

Пусть заданы $n + 1$ целых чисел $\{a_1, \dots, a_n, s\}$. Найти подмножество $J \subset \{1, \dots, n\}$, для которого $\sum_{i \in J} a_i = s$.

Задача распознавания КР заключается в проверке существования такого подмножества J .

**Определение 6.8. Задача о рюкзаке****Теорема 6.8**

Задача распознавания КР является NP-полной.

**Теорема 6.9.****Теорема 6.9**

Для всех $p \geq 1$ задача распознавания CVP является NP-полной для любой нормы l_p .

**Теорема 6.10.**

Доказательство. Задача CVP, очевидно, принадлежит классу NP. Достаточно продемонстрировать полиномиальную сводимость распознавательного варианта задачи КР к распознавательному варианту задачи CVP.

Пусть требуется решить задачу КР на входе $\{a_1, \dots, a_n, s\}$. Определим векторы b_i, t формулами

$$b_i = \left(a_i, \underbrace{0, \dots, 0}_{i-1}, 2, \underbrace{0, \dots, 0}_{n-i}, 0 \right)^T$$

и

$$t = \left(s, \underbrace{1, \dots, 1}_n \right)^T$$

Сводим теперь задачу КР на входе $\{a_1, \dots, a_n, s\}$ к задаче распознавания для CVP на входе $(B, t, \sqrt[p]{n})$. Здесь под $\sqrt[p]{n}$ будем понимать любое рациональное число, принадлежащее интервалу

$$[\sqrt[p]{n}, \sqrt[p]{n+1})$$

при $p = +\infty \Rightarrow \lim_{p \rightarrow \infty} \sqrt[p]{n} = 1$ – подставляем везде.

Сначала предположим, что существует решение задачи КР, то есть при некоторых $x_i \in \{0, 1\}$:

$$\sum_{i=1}^n x_i a_i = s$$

Тогда

$$Bx - t = \begin{pmatrix} \sum_{i=1}^n a_i x_i - s \\ 2x_1 - 1 \\ \vdots \\ 2x_n - 1 \end{pmatrix}$$

И p -я степень l_p нормы этого вектора равна

$$\|Bx - t\|_p^p = \underbrace{\left| \sum_{i=1}^n a_i x_i - s \right|^p}_0 + \underbrace{\sum_{i=1}^n |2x_i - 1|^p}_{\pm 1} = n$$

Поэтому расстояние от вектора t до решётки $\langle B \rangle \leq \sqrt[p]{n}$ и следовательно результатом задачи распознавания CVP будет YES.

Предположим теперь, что результатом задачи распознавания CVP на входе $(B, t, \sqrt[p]{n})$ будет YES. Следовательно, существует такой целочисленный вектор x , такой, что $\|Bx - t\|_p \leq \sqrt[p]{n}$. Тогда

$$\|Bx - t\|_p^p = \left| \sum_{i=1}^n a_i x_i - s \right|^p + \sum_{i=1}^n |2x_i - 1|^p \leq n$$

Причём для второго слагаемого выполняется соотношение $\sum_{i=1}^n |2x_i - 1|^p = n$, поскольку все величины $2x_i - 1$ по модулю равны 1. Поэтому данное неравенство возможно лишь когда $s = \sum_{i=1}^n a_i x_i$. \square

Лемма 6.10

Пусть $v = \sum_{i=1}^n c_i b_i$ – кратчайший вектор в решётке $\Lambda = \langle B \rangle$. Тогда при некотором i коэффициент c_i нечётный.



Лемма 6.11.

Доказательство. Пусть $v = \sum_{i=1}^n c_i b_i$ – кратчайший вектор в решётке и все коэффициенты c_i чётные. Тогда вектор

$$\frac{1}{2}v = \sum_{i=1}^n \frac{1}{2}c_i b_i$$

также вектор решётки и его длина вдвое меньше длины вектора v . \square

Лемма 6.11

Пусть $v = \sum_{i=1}^n c_i b_i$ – вектор в решётке $\Lambda = \langle B \rangle$, причём для некоторого j число c_j нечётно. Тогда

$$u = \frac{c_j+1}{2}(2b_j) + \sum_{i \neq j}^n c_i b_i$$

принадлежит решётке

$$B^{(j)} = (b_1, \dots, b_{j-1}, 2b_j, b_{j+1}, \dots, b_m)$$

и расстояние между u, b_j равно длине вектора v .



Лемма 6.12.

Лемма 6.12

Пусть $u = c'_j(2b_j) + \sum_{i \neq j}^n c_i b_i$ – вектор в решётке $\Lambda = \langle B^{(j)} \rangle$. Тогда

$$v = (2c'_j - 1)b_j + \sum_{i \neq j}^n c_i b_i$$

ненулевой вектор решётки $\langle B \rangle$ и длина v равна расстоянию между u и b_j .



Лемма 6.13.

Теорема 6.13

Задача SVP полиномиально эквивалентна CVP



Теорема 6.14.

Доказательство. По заданному базису $B = (b_1, \dots, b_m)$ построим m задач CVP следующим образом. Задача с номером j задаётся базисом

$$B^{(j)} = (b_1, \dots, b_{j-1}, 2b_j, b_{j+1}, \dots, b_m)$$

и вектором b_j . В задаче поиска используем m обращений к оракулу для CVP и из полученных m ответов $v_i, i = 1, \dots, m$ выбираем такой b_j , на котором достигается минимум $\|v_i - b_i\|$. Для задачи распознавания в качестве входа добавляется параметр r и выдаётся ответ YES, тогда и только тогда, когда хотя бы в одной из задач получен ответ YES.

Докажем корректность в случае задачи распознавания.

Пусть на входе (B, r) задачи SVP получаем YES и $v = \sum_{i=1}^n c_i b_i$ – кратчайший вектор в решётке $\langle B \rangle$. Тогда $\|v\| \leq r$ и согласно Лемма 6.14 при некотором коэффициенте c_j нечётный.

Тогда, согласно Лемма 6.14 вектор

$$u = \frac{c_j+1}{2}(2b_j) + \sum_{i \neq j}^n c_i b_i$$

принадлежит решётке $\langle B^{(j)} \rangle$ и расстояние $\|u - b_j\| = \|v\| \leq r$, что означает исход YES для запроса оракула на входе $(B^{(j)}, b_j, r)$.

Предположим теперь, что на входе $(B^{(j)}, b_j, r)$ задачи CVP получаем YES, то есть при некотором $u \in \langle B \rangle$ выполняется соотношение $\|u - b_j\| \leq r$. Тогда согласно лемме Лемма 6.14 для ненулевого вектора

$$v = (2c'_j - 1)b_k + \sum_{i \neq j}^n c_i b_i$$

решётки выполняется нужное соотношение, что означает исход YES, для задачи SVP \square

7. О Айтай

Определение 7.1 ((d, M)-решётка)

Пусть заданы натуральное n и вещественные $M > 0, d > 0$. Полная решётка $L \subset \mathbb{Z}^n$, содержащая $(n-1)$ -мерную подрешётку L' , для которой выполняются свойства

1. L' имеет базис из векторов, длины которых не более M .
2. Если $(n-1)$ -мерное подпространство $\langle L' \rangle = H \subset \mathbb{R}^n$ и H' – сдвиг H , неравный ему, имеет непустое пересечение с L , то расстояние между H и H' не менее d

называется **(d, M)-решётка**.



Определение 7.2. (d, M)-решётка

Предложение 7.2 (Генерация ключей)

1. Порождаем $(n-1)$ -мерную решётку L' с базисом (b_1, \dots, b_{n-1}) с условием $\|b_i\| \leq M$. Пусть H – линейная оболочка L' .
2. Выбираем $d \geq n^C M$
3. Выбираем из большого куба случайный вектор b_n с расстоянием $d \leq d_L \leq 2d$ от H
4. Секретный ключ – вектор b_n^*
5. Открытый ключ – случайный базис B' в

$$L = L(b_1, \dots, b_n)$$



Предложение 7.3. Генерация ключей

Предложение 7.3 (Алгоритм шифрования)

Вход: сообщение x , i -й бит которого равен x_i .

Выход: набор Y векторов, i -й вектор которого $y_i = (y_1^i, \dots, y_n^i)$ такой, что $y_j^i = \frac{z_j^i}{n}$, где z_j^i – целое.

Выполнение:

Для каждого бита $x_i = 1$ сообщения x выбираем случайный вектор $y^i = (y_1^i, \dots, y_n^i)$ в соответствии с равномерным распределением в параллелепипеде $\mathcal{P}(b_1^*, \dots, b_n^*)$.

Для каждой координаты y_j^i вектора y_i вычисляем её рациональное приближение $\tilde{y}_j^i = \frac{z_j^i}{n}$, такое, что $|y_j^i - \tilde{y}_j^i| < \frac{1}{n}$.

Биту $x_i = 0$ сообщения ставим в соответствие сумму случайного вектора $z_i \in \mathbb{R}^n$, выбранного в соответствии с нормальным распределением с функцией плотности

$$\rho(w) = e^{-\pi\|w\|^2}, w \in \mathbb{R}^n$$

Вектор z_i определяет единственный элемент $y_i \in \mathcal{P}(b_1^*, \dots, b_n^*)$, такой, что $y_i - z_i \in L$. Для каждой координаты y_j^i вектора y_i также вычисляется его рациональное приближение.

Предложение 7.4. Алгоритм шифрования**Предложение 7.4 (Алгоритм дешифрования)**

Через $[[x]]$ обозначим расстояние до ближайшего целого к x .

Вход: Набор Y , векторов y_i .

Выход: Последовательность x' битов x'_i .

Выполнение:

Для всех i находим скалярное произведение $\alpha_i = (y_i, b_n^*)$. Если $[[\alpha_i]] \geq \tilde{c}\sqrt{\log n}$, то получаем бит $x'_{i'} = 1$, иначе – 0.

Предложение 7.5. Алгоритм дешифрования

Предложение 7.5 (Алгебраический NTRU)

Элементы кольца

$$R = \frac{\mathbb{Z}[X]}{X^n - 1}$$

будем представлять многочленом или вектором в \mathbb{Z}^n вида

$$f = \sum_{i=0}^{n-1} f_i X^i = [f_0, f_1, \dots, f_{n-1}]$$

Произведение в этом кольце описывается формулой

$$f * g = [f_0, f_1, \dots, f_{n-1}] * [g_0, g_1, \dots, g_{n-1}] = [h_0, h_1, \dots, h_{n-1}]$$

где

$$h_k = \sum_{i=0}^k f_i g_{k-i} + \sum_{i=k+1}^{n-1} f_i g_{n+k-i}$$

Пусть p и q – два небольших взаимно простых числа. В кольцах $R_p = \frac{R}{p}$ и $R_q = \frac{R}{q}$ коэффициенты многочленов представляются остатками в диапазонах $[0, p-1]$ и $[0, q-1]$.

Рассмотрим также множество многочленов $\mathcal{P}_p(N)$, коэффициенты которого уже $\in (-\frac{p}{2}, \frac{p}{2}]$

**Предложение 7.6. Алгебраический NTRU****Предложение 7.6 (Генерация ключа)**

Выбираем два случайных многочлена $f, g \in R$ с маленькими коэффициентами (например, из множества $\{-1, 0, 1\}$) и взаимно простые числа p, q такие, что для многочлена f существуют обратные элементы f_p^{-1} и f_q^{-1} в кольцах R_p, R_q соответственно.

С вероятностью близкой к единице, случайный многочлен f удовлетворяет этому условию. Причём обратные элементы строятся с помощью алгоритма Евклида.

Затем вычисляется многочлен h в R_q :

$$h \equiv p f_q^{-1} * g \mod q$$

Открытым ключом шифрования объявляется многочлен h и числа q, p . Секретным ключом – пара f, g .

**Предложение 7.7. Генерация ключа**

Предложение 7.7 (Шифрование)

Пусть имеется пара – текст m (предполагаем, что текст представляет собой многочлен с маленькими коэффициентами, например, $\{-1, 0, 1\}$) и случайный вектор r .

Тогда зашифрованный текст t получается по формуле

$$t \equiv r * h + m \bmod q$$

**Предложение 7.8. Шифрование****Предложение 7.8 (Дешифрация)**

Пусть t – шифртекст и f – секретный ключ. Сначала вычислим многочлен a по формуле

$$a \equiv f * t \bmod q$$

причём коэффициенты многочлена a выбираются из интервала $(-\frac{q}{2}, \frac{q}{2}]$. Рассматривая многочлен a как многочлен с целыми коэффициентами, вычислим многочлен $m' \in R_p$ по формуле

$$m' = f_p^{-1} * a \bmod p$$

**Предложение 7.9. Дешифрация****Определение 7.9**

Целочисленная решётка, содержащая решётку $q\mathbb{Z}^n$ называется **q -модулярной**.

**Определение 7.10.****Определение 7.10**

Вектор $(x_n, x_1, \dots, x_{n-1})$ называется **циркулянт** вектора $x = (x_1, \dots, x_n)$ и обозначается $\text{rot}(x)$

**Определение 7.11.****Определение 7.11**

Пусть $x, y \in \mathbb{Z}^n$ и $z = (x, y) \in \mathbb{Z}^{2n}$. Определим **бициркулянт** формулой

$$\text{rot}_2(z) = (\text{rot}(x), \text{rot}(y))$$

**Определение 7.12.**

Определение 7.12

Целочисленная решётка \mathcal{L} размерности $2n$ называется **бициклической**, если из $x \in \mathcal{L}$ следует, что $\text{rot}_2(x) \in \mathcal{L}$.



Определение 7.13.

Лемма 7.13

Пересечение сохраняет свойства q -модулярности и бицикличности. В частности, для любого множества векторов S определена минимальная q -модулярная бициклическая решётка, содержащая множество S .



Лемма 7.14.

Определение 7.14

Квадратная невырожденная матрица $A \in \mathbb{Z}^{n \times n}$ называется **эрмитовой**, если

- A верхнетреугольная матрица
- Все диагональные элементы матрицы A строго положительны
- Все недиагональные элементы приведены по модулю соответствующего диагонального элемента в той же строке



Определение 7.15.

Предложение 7.15 (Генерация ключей)

Секретный ключ определяется коротким вектором

$$v = (x_1 p, \dots, x_n p, y_1, \dots, y_n) \quad x_i, y_i \in \{-1, 0, 1\}$$

Свяжем с этим вектором бициклическую q -модулярную решётку по Лемме 7.15.

Порождающим множеством этой решётки являются бицеркулянты вида $\text{rot}_2^k(v)$ для всех $k = 0, \dots, n-1$ и множество векторов вида $q e_k, k = 1, \dots, 2n$.

Открытый ключ определяется, как эрмитов нормальный базис бициклической q -модулярной решётки, определяемой вектором v .



Предложение 7.16. Генерация ключей

Предложение 7.16 (Шифрование/Дешифрование)

Рассмотрим вектор $(m, -r)$. При приведении этого вектора по модулю эрмитова нормального базиса H получим шифротекст $(t, 0)$, где t – многочлен из определения шифрования с помощью многочленов.

Алгоритм дешифрования не имеет геометрической интерпретации и выполняется по ранее описанным формулам.



Предложение 7.17. Шифрование/Дешифрование