МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

МЕХАНИКО–МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

Кафедра дифференциальных уравнений и системного анализа

Титульный лист

Дипломная работа

Реентовича Евгения
Васильевича

студента 4 курса, специальность 1-31 03 09 Компьютерная математика и системный анализ

Научный руководитель:
В.А. Липницкий

Минск, 2018

Содержание

Реферат

В дипломной работе 50 страниц, 1 рисунок, 4 таблицы, 13 источников, 1 приложение.

КВАДРАТИЧНО-ВЫЧЕТНЫЕ КОДЫ, ТЕОРИЯ НОРМ СИНДРОМОВ, КОДЫ БОУЗА-ЧОУДХУРИ-ХОКВИНГЕМА, КОДЫ ХЕММИНГА, МИНИМАЛЬНОЕ РАССТОЯНИЕ КОДА, НОРМА СИНДРОМА, ПОЛИНОМИАЛЬНАЯ НОРМА, ЦИКЛИЧЕСКИЙ СДВИГ, ЦИКЛОТОМИЧЕСКАЯ ПОДСТАНОВКА, Г-ОРБИТА, G-ОРБИТА.

Целью дипломной работы является применение для квадратично-вычетных кодов теории норм синдромов, а так же реализация алгоритма исправления ошибок квадратично-вычетными кодами при помощи теории норм синдромов.

Объектом исследования дипломной работы являются квадратично-вычетные коды над полем характеристика 2, в частности, коды с блоковой длиной 31 и 73.

В дипломной работе для квадратично-вычетных кодов длины 31 и 73 над полем характеристики 2 были получены следующие результаты:

- реализован алгоритм построения Г-орбит и созданы файлы, содержащие все Г-орбиты и соответствующие нормы синдромов;
- реализован алгоритм построения G-орбит и созданы файлы, содержащие все G-орбиты и соответствующие полиномиальные нормы;
- реализован алгоритм исправления случайных ошибок.

Дипломный проект носит практический характер. Разработанный алгоритм можно использовать для безопасной передачи информации через каналы связи с шумом.

Рэферат

У дыпломнай працы 50 старонак, 1 малюнак, 4 табліцы, 13 крыніц, 1 прыкладанне.

КВАДРАТЫЧНА-ВЫЛІЧНЫЯ КОДЫ, ТЭОРЫЯ НОРМ СІНДРОМАЎ, КОДЫ БОЎЗА-ЧОЎДХУРЫ-ХОКВІНГЕМА, КОДЫ ХЭММІНГА, МІНІМАЛЬНАЯ АДЛЕГЛАСЦЬ КОДА, НОРМА СІНДРОМУ, ПАЛІНАМІАЛЬНАЯ НОРМА, ЦЫКЛІЧНЫ ЗРУХ, ЦЫКЛАТАМІНАЯ ПАДСТАНОЎКА, Г-АРБІТА, G-АРБІТА.

Мэтай дыпломнай працы з'яўляецца прымяненне для квадратычна-вылічных кодаў тэорыі норм сіндромаў, а так жа рэалізацыя алгарытму выпраўлення памылак квадратычна-вылічнымі кодамі пры дапамозе тэорыі норм сіндромаў.

Аб'ектам даследавання дыпломнай працы з'яўляюцца квадратычна-вылічныя коды над полем характарыстыцы 2, у прыватнасці, коды з блочнай даўжынёй 31 і 73.

У дыпломнай працы для квадратычна-вылічных кодаў даўжыні 31 і 73 над полем характарыстыцы 2 былі атрыманы наступныя вынікі:

- рэалізаваны алгарытм пабудовы Г-арбіт і створаны файлы, якія змяшчаюць усе Г-арбіты і адпаведныя нормы сіндромаў;
- рэалізаваны алгарытм пабудовы G-арбіт і створаны файлы, якія змяшчаюць усе G-арбіт і адпаведныя палінаміальныя нормы;
- рэалізаваны алгарытм выпраўлення выпадковых памылак.

Дыпломны праект носіць практычны характар. Распрацаваны алгарытм можна выкарыстоўваць для бяспечнай перадачы інфармацыі праз каналы сувязі з шумам.

Abstract

In the diploma 50 pages, 1 drawing, 4 tables, 13 sources, 1 appendix.

QUADRATIC RESIDUE CODES, THEORY OF NORMS OF SYNDROMES, BOSE-CHAUDHURI-HOCQUENGHEM CODES, HAMMING CODES, MINIMUM HAMMING DISTANCE, NORM OF SYNDROMES, POLYNOMIAL NORM, CYCLIC SHIFT, CYCLOTHYMIC PERMUTATION, Γ-ORBIT, G-ORBIT.

The purpose of the diploma is to apply the theory of norms of syndromes for quadratic residue codes, as well as the implementation of the algorithm for correcting errors by quadratic residue codes using the theory of norms of syndromes.

The subject of the research work is the quadratic residue codes over the field characteristic 2, in particular, codes with block lengths of 31 and 73.

In the diploma for the quadratic residue codes of length 31 and 73 over the field of characteristic 2, the following results were obtained:

- implemented an algorithm for constructing the *Gamma*-orbits and created files, containing all the *Gamma*-orbits and the corresponding norms of syndromes;
- implemented an algorithm for constructing the G-orbits and created files, containing all the G-orbits and the corresponding polynomial norms;
- implemented an algorithm for correcting random errors.

The diploma is of a practical nature. The developed algorithm can be used for secure transmission of information through communication channels with noise.

Введение

Современное общество находится на этапе своего развития, который называется информационным или постиндустриальным. Под этим обычно понимается, что знания и информация стали основной движущей силой общества. Таким образом, передача, хранение, обработка и использование информации стали основными задачами для успешного развития человека и общества. Остановим свой взгляд на первой из перечисленных задач. Если лишить информацию возможности переходить из одного хранилища в другое, то она практически полностью теряет свой смысл. Так же немаловажным требованием к передаче данных является надежность, так как случайные ошибки в информации могут привести к огромным издержкам в обществе.

Одним из основных способов защиты информации при передаче по каналу связи с шумом является помехоустойчивое кодирование. Как наука, теория помехоустойчивого кодирования возникла в конце 40-х годов с появлением работ Шеннона, Хемминга, Голея, и в настоящее время получила широкое распространение в различных областях информатики и радиоэлектроники: системы и сети телекоммуникаций, системы радионавигации, радиолокации, цифровое телевидение, вычислительные машины и системы, компьютерные сети. Теория кодирования тесно связана с алгеброй: теорией чисел, теорией групп и колец, теорией полей и полей Галуа.

Наиболее популярными с практической точки зрения являются линейные циклические коды, такие как коды Хемминга, коды Боуза-Чоудхури-Хоквингема, коды Рида-Соломона. Без них не мыслимы системы сотовой связи, защита компьютерной памяти, диспетчерские компьютерные службы и т.д. Созданная белорусской школой теория норм синдромов (ТНС) на рубеже XX-XXI веков, позволила ускорить на порядок синдромные методы коррекции ошибок БЧХ-кодами, до предела расширить спектры корректируемых ошибок.

Данная работа посвящена расширению ТНС на класс квадратично-вычетных кодов. КВ-коды обладают высокой скоростью, большим минимальным расстоянием и с ростом длины характеристики КВ-кодов становятся только лучше, чем не могут похвастаться многие другие известные коды, что привлекло внимание большого числа исследователей. Однако, не смотря на богатую алгебраическую структуру, данные коды достаточно тяжело поддаются исследованию, до настоящего времени не известно общего и эффективного метода исправления ошибок квадратично-вычетными кодами.

В ходе выполнения работы предлагается решение следующих задач:

- изучение основ линейных кодов. циклических кодов, кодов Хемминга и кодов Боуза-Чоудхури-Хоквингема;
- изучение квадратично-вычетных кодов и их основных свойств;

- расчет точного минимального расстояния для КВ-кодов;
- развитие норменно-полиномиального метода для исправления ошибок квадратично-вычетными кодами.

1 Основы теории линейных кодов

1.1 Понятие линейного кода

Процесс помехоустойчивого кодирования заключается в том, что источник информации или отправитель выбирает блок информации определенного размера (блочное кодирование) и добавляет к информационным символам набор избыточных символов, чьи значение и количество зависят от выбранного кода. Далее закодированный сигнал проходит по каналу связи, в котором с некоторой вероятностью в кодовом слове происходят случайные ошибки. На приемном конце, используя избыточные символы, получатель пытается определить в каких местах произошла ошибка и исправить ее.

Одной из важнейших теорем помехоустойчивого кодирования является теорема Клода Шеннона, выражающей главную цель и назначение помехоустойчивых кодов.

Теорема 1. [1] *Введением избыточности в передаваемую в зашумленном канале связи информацию можно добиться исправления возникающих в процессе передачи этой информации сколь угодно сложных ошибок.*

В качестве информационных символов выступают элементы некоторого конечного множества, которое практически всегда является конечным полем. Количество информационных символов обозначают символом k. Таким образом, сообщение имеет вид $K = \{(x_1, x_2, \ldots, x_k), x_i \in P\}$, где P – конечное поле.

Одним из способов кодирования является линейное кодирование. Понятие линейных кодов – одно из первичных, базовых понятий теории и практики помехоустойчивого кодирования. Сформировалось в теории информации к середине XX века. В линейном коде множество всех сообщений k образует линейное пространство P_k. В ходе кодирования каждый символ кодового слова вычисляется как линейная функция от информационных символов, или другими словами – линейное отображение линейного пространства P_k в P_n, где $n > k$, n – длина кодового слова, то есть сумма количества информационных и избыточных символов.

Поясним этот момент сведениями из линейной алгебры. Если оговорено или из контекста ясно, с какими базисами этих пространств мы имеем дело, то линейный оператор $\varphi : P_k \to P_n$ однозначно определяется матрицей A_φ этого оператора в данных базисах. Пусть $u_1, u_2, \ldots, u_k$ – базис пространства P_k, а $\nu_1, \nu_2, \ldots, \nu_n$ – базис пространства P_n. Полученные координаты векторов $\varphi(u_1), \varphi(u_2), \ldots, \varphi(u_k)$ составляют столбцы матрицы A_φ. Как известно, для всякого вектора $x = (x_1, x_2, \ldots, x_k) \in P_k$ вектор-столбец $A_\varphi(x) = (y)$, где (x) – столбец из координат вектора x, состоит из координат вектора $\varphi(x)$ в базисе $\nu_1, \nu_2, \ldots, \nu_n$ пространства P_n. Протранспонировав равенство $A_\varphi(x) = (y)$, получим соотношение: $\phi(x) = x \cdot A_\varphi^T$. Таким образом, из сказанного выше следует, что матрица $G = A_\varphi^T$. Матрицу G в дальнейшем будем называть порождающей матрицей, она имеет размерность $k \times n$ Векторы из

P_n будем называть кодовыми словами.

Определение 1. [2] Линейным (n, k)-кодом над полем P называется произвольное k-мерное подпространство линейного пространства P_n. Параметр n называется длиной кода, а k – размерностью кода. Линейный (n, k)-код называется высокоскоростным, если отношение k/n близко к 1, и низкоскоростным, если отношение $k/n \ll 1$ – близко к нулю.

Пример 1. *Над полем характеристики 2 пусть $n = k + 1$ и все n символов удовлетворяют уравнению*

$$x_1 + x_2 + \ldots + x_n = 0.$$

Тогда полученный код будет называться кодом проверки на четность, потому что, как видно из уравнения, сумма всех чисел четна (или равна нулю по модулю характеристики). В случае, если на приемном конце получили нечетное количество единиц (сумма чисел равна 1), то можно с уверенностью сказать, что в во время передачи произошла ошибка.

В дальнейшем, если не оговорено заранее, характеристикой поля всегда будет являться число 2.

1.2 Порождающая и проверочная матрицы линейного кода

В предыдущем разделе было дано определение порождающей матрицы G.

Из линейной алгебры известно, что множество решений однородной системы уравнений образует подпространство размерности $r = n - k$, где n – размерность пространства, k – ранг системы уравнений.Базис данного подпространства называется ядром и обозначается Ker.

Определение 2. Проверочной матрицей кода называется матрица $H = KerG$.

Как было сказано ранее, любое кодовое слово u вычисляется по формуле $u = x.G$, где x – сообщение. Следовательно, каждое слово линейно выражается через строки порождающей матрицы, то есть является элементом подпространства, чьим базисом являются строки матрицы G. Из этого следует, что

$$H.u^T = 0 \tag{1}$$

для любого кодового слова u, благодаря чему становится очевидным, почему проверочная матрица так называется: если кодовое слово пришло без ошибок, то $H \cdot u^T = 0$, если же в ходе передачи данных по каналу связи кодовое слово было изменено, то $H \cdot u^T \neq 0$.

Проверочная матрица имеет размерность $(n-k) \times n$. Из определения следует, что матрицы G и H связаны соотношением

$$GH^T = 0 \qquad HG^T = 0.$$

Достаточно удобным является способ записи проверочной матрицы в виде[3]

$$H = [A|I_{n-k}]$$

так как тогда легко выразить порождающую матрицу

$$G = [I_k| - A^T].$$

Так же удобство заключается и в том, что матрица G в таком виде порождает код, в котором первые k элементов являются в точности сообщением. Такие коды называются систематическими.

Предложение 1. [2] *Пусть H – проверочная матрица кода C. Тогда для всякой невырожденной квадратной матрицы A порядка $m = n - k$ матрица $A\dot{H}$ также является проверочной для кода C.*

Предложение 2. [2] *Пусть H и H_1 – две проверочные матрицы линейного кода (n, k)-кода C. Тогда существует квадратная $m \times m$ матрица A для $m = n - k$, такая, что $AH = H_1$.*

Из предположений 1 и 2 следует критерий того, что две матрицы являются проверочными матрицами одного и того же линейного кода.

Теорема 2. [2] *Пусть H – проверочная $m \times n$-матрица линейного (n, k)-кода C. Матрица H^* порядка $m \times n$ является проверочной матрицей этого же кода тогда и только тогда, когда найдется такая невырожденная $m \times m$-матрица A, что $H^* = AH$.*

Следствие 1. [2] *Если у проверочной матрицы H кода C столбцы $h_{i1}, h_{i2}, \ldots, h_i m$ образуют ненулевой (нулевой) минор, то и все остальные проверочные матрицы кода C обладают тем же свойством.*

Следствие 2. [2] *Количество различных проверочных матриц линейного (n, k)-кода над конечным полем P совпадает с количеством различных невырожденных квадратных матриц порядка m в поле P, то есть с порядком группы $GL_m(P)$ невырожденных квадратных матриц порядка m над полем P.*

Пример 2. *Приведем пример линейного кода*

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Так как $-1 \equiv 1(mod\ 2)$ *то проверочная матрица будет иметь вид*

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Код, соответствующий этим матрицам состоит из следующих кодовых слов:

$$C = \begin{Bmatrix} (0 & 0 & 0 & 0 & 0 & 0 & 0) \\ (1 & 1 & 0 & 0 & 0 & 0 & 1) \\ (1 & 0 & 1 & 0 & 0 & 1 & 0) \\ (0 & 1 & 1 & 0 & 0 & 1 & 1) \\ (0 & 1 & 1 & 0 & 1 & 0 & 0) \\ (1 & 0 & 1 & 0 & 1 & 0 & 1) \\ (1 & 1 & 0 & 0 & 1 & 1 & 0) \\ (0 & 0 & 0 & 0 & 1 & 1 & 1) \\ (1 & 1 & 1 & 1 & 0 & 0 & 0) \\ (0 & 0 & 1 & 1 & 0 & 0 & 1) \\ (0 & 1 & 0 & 1 & 0 & 1 & 0) \\ (1 & 0 & 0 & 1 & 0 & 1 & 1) \\ (1 & 0 & 0 & 1 & 1 & 0 & 0) \\ (0 & 1 & 0 & 1 & 1 & 0 & 1) \\ (0 & 0 & 1 & 1 & 1 & 1 & 0) \\ (1 & 1 & 1 & 1 & 1 & 1 & 1) \end{Bmatrix}.$$

Можно заметить, что сумма любых двух кодовых слов является кодовым словом, что соответствует определению линейного кода.

1.3 Метрика Хемминга

Определение 3. Метрикой или расстоянием на множестве X называется определенная на декартовом произведении $X \times X$ функция ρ с неотрицательными действительными значениями, удовлетворяющая любых $x, y \in X$ условиями:

1. $\rho(x, y) = 0$ тогда и только тогда, когда $x = y$(аксиома тождества);
2. $\rho(x, y) \leq \rho(x, z) + \rho(y, z)$(аксиома треугольника);
3. $\rho(x, y) = \rho(y, x)$(аксиома симметрии).

В теории помехоустойчивого кодирования наиболее распространенной метрикой является метрика Хемминга, чье определение представлено ниже.

Определение 4. [3] Расстоянием Хемминга между векторами $x, y \in P_n$ называется количество $dist(x, y)$ несовпадающих координат этих векторов.

Весом $wt(x)$ вектора $x \in P_n$ называется количество ненулевых координат этого вектора.

Очевидными являются следующие утверждения:

- $dist(x, y) = wt(x - y)$;
- $wt(x + y) \leq wt(x) + wt(y)$.

Лемма 1. *Расстояние Хемминга обладает всеми свойствами обычного расстояния:*

1. $dist(x, y) = 0$ *тогда и только тогда, когда* $x = y$;
2. $dist(x, z) + dist(z, y) \geq dist(x, y)$;
3. $dist(y, x) = dist(x, y)$.

Определение 5. [2] t-окрестностью вектора $x \in P_n$ назовем совокупность всех векторов $y \in P_n$, для которых $dist(x, y) \leq t$.

Лемма 2. [2] *Если* $dist(x, z) > 2t$, *то* t-*окрестности векторов* $x, z \in P_n$ *не пересекаются.*

1.4 Минимальное расстояние кода

Определение 6. [2] Минимальным или кодовым расстоянием кода C называется наименьшее из расстояний между попарно различными векторами кода C.

Из равенства $dist(x, y) = wt(x - y)$ следует, что минимальное расстояние линейного кода равно наименьшему из весов ненулевых векторов этого кода.

Значение кодового расстояния определяет следующая – фундаментальная в помехоустойчивом кодировании

Теорема 3. [2] *Если минимальное расстояние кода* C *равно* $d = 2t + 1$ *или* $2t + 2$, *то код* C *может обнаружить до* $d - 1$ *ошибок и исправить до* t *ошибок в каждом принятом векторе-слове длиной* n.

Доказательство. Возможность обнаружения до $d - 1$ ошибок. Если к кодовому слову $c \in C$ прибавить любой вектор $e \in P_n$, у которого менее d ненулевых координат, то полученный вектор $x = c + e$ не принадлежит, очевидно,

подпространству C (в противном случае $x - c = e \in C$, что противоречит минимальности d). Как мы уже знаем, принадлежность и не принадлежность данного вектора коду C определяется результатом умножения на проверочную матрицу H этого кода. Результат $H \cdot x \neq 0$ означает, что принятое сообщение $x \notin C$ и, следовательно, в отличие от переданного слова c, содержит ошибки.

О возможности декодирования до t ошибок. Если к кодовому слову $c \in C$ прибавить любой вектор $e \in P_n$, у которого $\tau \leq t$ ненулевых координат, то полученный вектор $x = c + e$ находится на расстоянии τ от вектора $c \in C$. Если c_i – другой вектор подпространства C, то в силу леммы (2) $dist(x, c_i) > t$. Предположение о противоположном неравенстве приводит к противоречию: если предположить, что найдется вектор $c_j \in C$, для которого $dist(x, c_j) \leq t$, то в силу неравенства треугольника $d \leq dist(c, c_j) \leq dist(c, x) + dist(x, c_j) \leq 2t < d$.

Таким образом, существует единственный вектор $c \in C$, находящийся на расстоянии $\rho \leq t$ от принятого вектора-сообщения x. Этот однозначно определенный вектор c естественно взять в качестве правильного, не искаженного переданного сообщения. Теорема полностью доказана.

Замечание 1. [2] *Если вектор ошибок e содержит $\tau \geq d$ ненулевых координат, то он может оказаться кодовым словом, то есть принадлежать подпространству C. Тогда в силу замкнутости C относительно операции сложения вектор $x = c + e \in C$. В таком случае мы не можем заметить ошибки в принятом сообщении $x = c + e$.*

Замечание 2. [2] *Предложенный в доказательстве теоремы метод декодирования носит название "декодирование в ближайшее кодовое слово" или "декодирование по максимуму правдоподобия" (см.* [4]*).*

Следующая теорема служит критерием для определения минимального расстояния кода.

Теорема 4. [2] *Пусть H – проверочная матрица двоичного кода C. Минимальное расстояние этого кода равно d тогда и только тогда, когда любые $d-1$ столбцов матрицы H линейно независимы, но найдутся d линейно зависимых столбцов.*

Доказательство теоремы вытекает из следующей леммы.

Лемма 3. [2] *Пусть $[i]$ – i-й столбец проверочной матрицы H линейного кода C над полем P. Вектор $c \in P_n$ весом ω с ненулевыми координатами на позициях $i_1, i_2, \ldots, i_\omega$ принадлежит коду C тогда и только тогда, когда система столбцов $[i_1], [i_2], \ldots, [i_\omega]$ линейно зависима.*

Доказательство. По условию $c = 0, \ldots, 0, c_{i_1}, 0, \ldots, 0, c_{i_2}, 0, \ldots, 0, c_{i_\omega}$ с единственными ненулевыми координатами $c_{i_j} \in P$, $1 \leq j \leq \omega$. Вектор c принадлежит коду C тогда и только тогда, когда $H \cdot c^T = 0$. По свойствам матричного умножения $H \cdot c^T = c_{i_1}[i_1] + c_{i_2}[i_2] + \ldots + c_{i_\omega}[i_\omega]$. Равенство $c_{i_1}[i_1] + c_{i_2}[i_2] + \ldots + c_{i_\omega}[i_\omega] = 0$ означает линейную зависимость столбцов $[i_1], [i_2], \ldots, [i_\omega]$. Лемма доказана.

1.5 Коды Хемминга

В разделе 1.2 мы строили конструкцию (1), говоря, что u является кодовым словом, если выражение равняется нулевому вектору.

Определение 7. Синдромом ошибок в векторе сообщений u в данном коде C, заданным проверочной матрице H, называется вектор $S(u) = H \cdot u^T$.

Другими словами, синдром – это сумма столбцов матрицы H номера которых совпадают с номерами разрядов в которых произошла ошибка. Действительно, пусть нам пришло слово $u = x + e$, где x – кодовое слово, e – ошибка, тогда $Hu = H(x + e) = Hx + He = He$.

Было бы здорово составить код, способный легко исправлять одиночные ошибки. Например, пускай синдром ошибки указывает на разряд в котором произошла ошибка. Для этого нужно задать проверочную матрицу так, чтобы каждый столбец являлся двоичным представлением своего номера в матрице.

Пример 3.

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Возьмем кодовое слово (1 0 1 0 1 0 1) *и допустим ошибку в третьем разряде* $u = (1\ 0\ 0\ 0\ 1\ 0\ 1)$

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}^T = 110_2 = 3_{10}.$$

Определение 8. [3] Двоичный код Хемминга H_χ длины $n = 2^m - 1$ имеет проверочную матрицу H, столбцы которой состоят из всех ненулевых двоичных векторов длины m, причем каждый вектор встречается в матрице один раз. Код H_χ имеет параметры $n = 2^m - 1; k = 2^m - 1 - m; d = 3$.

Можно заметить, что код из примера (3) похож на код из примера (2). Действительно, данные коды являются эквивалентными, так как их проверочные матрицы отличаются только перестановкой столбцов.

Определение 9. Два кода называются эквивалентными, если они отличаются только перестановкой символов (столбцов/строк матрицы).

Теорема 5. *Пусть код C имеет минимальное расстояние равное $2t + 1$, тогда синдромы всех ошибок, веса меньше либо равного t, различны.*

Доказательство. Возьмем два вектора ошибок e_1 и e_2 таких, что $wt(e_1) < t$, $wt(e_2) \leq t$ и $S(e_1) = S(e_2)$. Тогда вектор $e = e_1 + e_2$ имеет синдром $S(e) = H \cdot (e_1 + e_2)^T = H \cdot e_1^T + H \cdot e_2^T = S(e_1) + S(e_2) = 0$, из чего следует, что e является кодовым словом, однако его вес $wt(e) < 2t + 1$. Получили противоречие.

2 Исследование теории квадратично-вычетных кодов

2.1 Код, исправляющий две ошибки

Возьмем в качестве примера следующий код Хемминга, исправляющий одну ошибку.

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

С проверочной матрицей гораздо удобнее работать если представить ее столбцы в виде степей многочленов конечного поля, например, переставим столбцы нашей матрицы следующим образом:

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

или

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} \end{pmatrix}$$

где все $\alpha^i \in GF(2^4)$ [5]с неприводимым многочленом $x^4 + x + 1$.

Попробуем добавить к этой матрице еще одну строку, например таким способом:

$$H' = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & \alpha^{10} & \dots \\ f[1] & f[\alpha] & f[\alpha^2] & f[\alpha^3] & f[\alpha^4] & f[\alpha^5] & f[\alpha^6] & f[\alpha^7] & f[\alpha^8] & f[\alpha^9] & f[\alpha^{10}] & \dots \end{pmatrix}$$

где значением функции $f(i)$ так же соответствует многочлен из $GF(2^4)$. Но как выбрать функцию $f(i)$? Предположим, что произошли две ошибки на позициях столбцов i и j. Тогда синдром равен

$$S = H_i + H_j = \begin{pmatrix} i + j \\ f(i) + f(j) \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}.$$

Необходимо выбрать $f(i)$ так, чтобы декодер по S мог найти i и j, то есть чтобы он мог одновременно решить уравнения

$$\left.\begin{array}{l} i+j=z_1; \\ f(i)+f(j)=z_2; \end{array}\right\}$$

относительно i и j при заданных z_1 и z_2. Пусть, например $f(i)=ci$, где c – постоянная, тогда:

$$\left.\begin{array}{l} i+j=z_1; \\ c(i+j)=z_2. \end{array}\right\} \Longrightarrow cz_1=z_2,$$

следовательно система не может быть решена. Тогда попробуем $f(i)=i^2$:

$$\left.\begin{array}{l} i+j=z_1; \\ i^2+j^2=z_2. \end{array}\right\} = \left.\begin{array}{l} i+j=z_1; \\ (i+j)^2=z_2. \end{array}\right\} \Longrightarrow z_1^2=z_2$$

и данная система так же не имеет решений. $f(i)=i^3$ – вариант, при котором система будет иметь решения:

$$\left.\begin{array}{l} i+j=z_1\neq 0; \\ i^3+j^3=z_2. \end{array}\right\} \Longrightarrow z_2=(i+j)(i^2+ij+j^2)=z_1(z_1^2+ij).$$

откуда

$$ij=z_2/z_1+z_1^2.$$

И получаем уравнение

$$x^2+z_1x+(z_2/z_1+z_1^2)=0(z_1\neq 0). \tag{2}$$

Таким образом, код C, заданный проверочной матрицей

$$H=\begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} \end{pmatrix}$$

способен исправлять до двух ошибок, которые можно найти решив квадратное уравнение (2).

2.2 Циклические и БЧХ-коды

Определение 10. [3] Код называется циклическим если он линеен и любой циклический сдвиг влево/вправо его кодового слова так же является кодовым словом.

Циклические коды хороши тем, что они хорошо описываются при помощи алгебры. Представим каждое кодовое слово в виде многочлена $c = (c_0, c_1, \ldots, c_{n-1}) = c_0 + c_1 x + c_2 x^2 + \ldots + c_{n-1} x^{n-1}$. Тогда циклический сдвиг эквивалентен умножению многочлена на x по модулю $1 + x^n$. Рассмотрим кольцо $R_n = F[x]/(x^n + 1)$, где $F[x]$ – кольцо всех многочленов, R_n – кольцо многочленов степень меньше чем n (кольцо многочленов по модулю $x^n + 1$).

Определение 11. [3] Циклическим кодом длины n называется идеал кольца R_n.

Определение 12. [3] Порождающим многочленом $g(x)$ называется многочлен, порождающий идеал $< g(x) >$ кольца R_n.

Например, порождающим многочленом кода из примера 2 является многочлен $x^3 + x + 1$. Порождающим многочленом кода и раздела 3.1 является многочлен $1 + x^4 + x^6 + x^7 + x^8$. Если умножить порождающий многочлен на сообщение, представленное в виде многочлена, по полученный многочлен, степени меньше чем n, будет являться кодовым словом.

Определение 13. [3] Проверочным многочленом $h(x)$ называется многочлен, который вычисляется по формуле $h(x) = (x^n + 1)/g(x)$.

У проверочного многочлена такой же смысл как у проверочной матрицы: при умножение кодового слова на проверочный многочлен получим многочлен равный нулю.

Теорема 6. [3] *Пусть C – циклический код с таким порождающим многочленом $g(x)$, что для некоторых целых чисел $b \geq 0$ и $\delta \geq 1$ выполняется равенство*

$$g(\alpha^b) = g(\alpha^{b+1}) = \ldots = g(\alpha^{b+\delta-2}) = 0.$$

Иными словами, точно $\delta - 1$ последовательных степеней α являются нулями кода. тогда минимальное расстояние кода равно по меньшей мере δ.

Определение 14. [3] Циклический код длины n над $GF(q)$ называется кодом БЧХ с конструктивным расстоянием δ, если для некоторого целого числа $b \geq 0$ его порождающий многочлен $g(x)$ является многочленом наименьшей степени что элементы $\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+\delta-2}$ являются его корнями.

Если $b = 1$, то код называется кодом БЧХ в узком смысле, если $n = q^m - 1$, код называется примитивным кодом БЧХ.

В общем случае проверочная матрица БЧХ кода выглядит следующим образом:

$$H = \begin{pmatrix} 1 & \alpha^{b} & \alpha^{2b} & \cdots & \alpha^{(n-1)b} \\ 1 & \alpha^{b+1} & \alpha^{2(b+1)} & \cdots & \alpha^{(n-1)(b+1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{b+\delta-2} & \alpha^{2(b+\delta-2)} & \cdots & \alpha^{(n-1)(b+\delta-2)} \end{pmatrix} \tag{3}$$

Так же, зная порождающий/проверочный многочлен, можно легко построить порождающую/проверочную матрицу[3]. Пусть $g(x) = g_0 + g_1 x + \cdots + g_r x^r$, тогда

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & \cdots & g_r & & 0 \\ & g_0 & g_1 & \cdots & g_{r-1} & g_r & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & & g_0 & g_1 & \cdots & g_{r-1} & g_r \end{pmatrix}$$

$$H = \begin{pmatrix} 0 & & h_k & \cdots & h_2 & h_1 & h_0 \\ & h_k & h_{k-1} & \cdots & h_1 & h_0 & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ h_k & h_{k-1} & \cdots & h_1 & h_0 & & 0 \end{pmatrix}$$

2.3 Квадратично-вычетные коды

Определение 15. Число a называется квадратичным вычетом по модулю простого числа p если имеет решение сравнение

$$x^2 \equiv a (mod\ p).$$

Если сравнение решений не имеет, что число a называют квадратичным невычетом по модулю p.

Для того чтобы найти все квадратичные вычеты по модулю p, достаточно рассмотреть квадраты чисел от 1 до $(p-1)/2$. Следовательно $(p-1)/2$ чисел будут являться квадратичными вычетами, остальные $(p-1)/2$ чисел будут являться квадратичными невычетами.

Возьмем два простых числа p и l, причем l является квадратичным вычетом по модулю p. p будет являться длинной кода, l характеристикой поля. Так как мы исследуем коды над полями характеристики 2, то , согласно теореме [Мак-Вильямс, 463], число p будет простым числом вида $8m \pm 1$.

Обозначим через Q множество всех квадратичных вычетов по модулю p, а через N – всех квадратичных невычетов. Пусть α – примитивный корень степени p из единицы в некотором поле, содержащем поле $GF(l)$. Зададим два многочлена

$$q(x) = \prod_{r \in Q} (x - \alpha^r) \qquad n(x) = \prod_{n \in N} (x - \alpha^n)$$

коэффициенты которых лежат в поле $GF(l)$. При этом

$$x^p - 1 = (x-1)q(x)n(x).$$

Определение 16. [3] Квадратично-вычетными кодами $\mathcal{L}, \overline{\mathcal{L}}, \mathcal{N}, \overline{\mathcal{N}}$ называются циклические коды кольца R, порожденные соответственно многочленами

$$q(x), (x-1)q(x), n(x), (x-1)n(x).$$

В зависимости от выбора неприводимого многочлена код $\mathcal{L}$ ($\overline{\mathcal{L}}$) может порождать любой из многочленов $q(x)$ или $n(x)$ ($(x-1)q(x)$ или $(x-1)n(x)$). То же самое можно сказать и про код $\mathcal{N}$ ($\overline{\mathcal{N}}$).

Ясно, что $\mathcal{L} \subset \overline{\mathcal{L}}$ и $\mathcal{N} \subset \overline{\mathcal{N}}$. В двоичном случае $\overline{\mathcal{L}}$ представляет собой подкод кода $\mathcal{L}$, состоящий из всех кодовых слов четного веса, а $\overline{\mathcal{N}}$ – подкод кода $\mathcal{N}$, состоящий из всех кодовых слов четного веса. Размерность кодов $\mathcal{L}$ и $\mathcal{N}$ равна $(p+1)/2$, для кодов $\overline{\mathcal{L}}$ и $\overline{\mathcal{N}}$ – $(p-1)/2$.

Подстановка на множестве координат кольца R, индуцируемая отображением $x \to x^n$ при некотором фиксированном невычете n, переводит коды $\mathcal{L}$ и $\mathcal{N}$ (а также $\overline{\mathcal{L}}$ и $\overline{\mathcal{N}}$) друг в друга[3], так что эти коды эквивалентны. Минимальное расстояние КВ-кодов $d \geq \sqrt{p}$.

В двух последних абзацах видно, что все четыре вида кодов имеют одинаковые характеристики и легко переходят друг в друга, поэтому, если не оговорено заранее, под квадратично-вычетным кодом будет пониматься код $\mathcal{L}$.

Пример 4. *Приведем несколько примеров квадратично вычетных кодов.*

Код Хемминга длинной 7 *является квадратично вычетным кодом с порождающим многочленом* $q(x) = x^3 + x + 1$ *(*$q(x) = 1 + x^2 + x^3$*).*

Коды Голея длины 23 *над полем* $GF(2)$ *и длины* 11 *над полем* $GF(3)$ *так же являются квадратично-вычетными кодами.*

$$x^{23} - 1 = (1+x)(1+x+x^5+x^6+x^7+x^9+x^{11})(1+x^2+x^4+x^5+x^6+x^{10}+x^{11}).$$

$$x^{11} - 1 = (x-1)(-1-x+x^2-x^3+x^5)(-1+x^2-x^3+x^4+x^5).$$

Квадратично-вычетный код длины 31 может быть представлен многочленом $q(x) = 1 + x + x^2 + x^6 + x^7 + x^{12} + x^{15}$*. Код длины* 73 *– многочленом* $q(x) = 1 + x + {} + x^5 + x^6 + x^7 + x^8 + x^{11} + x^{15} + x^{17} + x^{18} + x^{19} + x^{21} + x^{25} + x^{28} + x^{29} + x^{30} + x^{31} + x^{35} + x^{36}$.

2.4 Проверочная матрица квадратично-вычетного кода

Определение 17. Циклотомический класс по модулю p над полем $GF(q)$ определяется как множество:

$$C_s = \{s, sq, sq^2, \cdots, sq^{m_s-1}\},$$

где $sq^{m_s} \equiv s(mod\ p)$.

В качестве s может быть выбран любой член циклотомического класса. Множество всех целых чисел по модулю n разбивается на непересекающиеся циклотомические классы:

$$\{0, 1, 2, \cdots, p-1\} = \bigcup_s C_s,$$

где s пробегает множество представителей классов по модулю p.

Отдельно стоит выделить циклотомический класс, состоящий только из нулевого элемента. В дальнейшем, чтобы упростить вычисления, будем считать что данного циклотомического класса не существует.

Предложение 3. *Если p – простое число, то количество членов каждого циклотомического класса будет одинаковым.*

Для любого простого числа p, для которого 2 является квадратичным вычетом, количество циклотомических классов над полем $GF(2)$ является четным числом, причем половина циклотомических классов состоит из всех квадратичных вычетов по модулю p, другая половина классов – из квадратичных невычетов по модулю p. В данном случае количество элементов в каждом циклотомическом классе равняется числу m, которое является наименьшим числом, удовлетворяющим следующему выражению:

$$2^m - 1 \equiv 0(mod\ p). \tag{4}$$

Следовательно, количество самих циклотомических классов равно $n = \frac{p-1}{m}$.

Предложение 4. [6] *Всякий двоичный КВ-код длиной p, определенный над полем $GF(2^m)$ – минимальным расширением поля $GF(2)$, содержащим все корни p-й степени из 1, является обобщенным не примитивным БЧХ-кодом, то есть с проверочной матрицей вида*

$$H = [\beta^{q_1}, \beta^{q_2}, \cdots, \beta^{q_{(n/2)}}]^T, \tag{5}$$

и с конструктивным расстоянием $\delta = 2t + 1$, где $(p-1)/2 = mt$.

Здесь q_i – представитель циклотомического класса, состоящего из квадратичных вычетов, причем все q_i принадлежат разным циклотомическим классам. $\beta^{q_i} = (1, \alpha^{q_i}, \alpha^{2q_i}, \ldots, \alpha^{(p-1)q_i})$ – строка проверочной матрицы, представленной в виде (3).

2.5 Идемпотенты

Определение 18. [3] Многочлен $E(x)$ кольца R_n называется идемпотентом, если $E(x) = E^2(x) = E(x^2)$.

Например, $x + x^2 + x^4$ – идемпотент в R_7, так как $(x + x^2 + x^4)^2 = x +^2 +x^4$. Многочлены 1 и $x^3 + x^6 + x^5$ также являются идемпотентами. В общем случае

$$E(x) = \sum_{i=0}^{n-1} \varepsilon_i x^i$$

– идемпотент тогда и только тогда, когда $\varepsilon_i = \varepsilon_{2i}$ (индексы берутся по модулю n).

Теорема 7. [3] *Циклический код, или идеал $C =< g(x) >$, содержит единственный идемпотент $E(x)$ такой, что $C =< E(x) >$. Кроме того, $E(x) = p(x)g(x)$ для некоторого многочлена $p(x)$ и $E(\alpha^i) = 0$ тогда и только тогда, когда $c(x)E(x) = 0$. $c(x) \in C$ тогда и только тогда, когда $c(x)E(x) = c(x)$.*

Теорема 8. [3] *Для квадратично-вычетного кода с характеристиками $l = 2$ и $p = 4k - 1$ порождающим идемпотенты кодов $\mathcal{L}, \overline{\mathcal{L}}, \mathcal{N}, \overline{\mathcal{N}}$ будут соответственно равны:*

$$E_q(x) = \sum_{r \in Q} x^r; \quad F_q(x) = 1 + \sum_{n \in N} x^n;$$

$$E_n(x) = \sum_{n \in N} x^n; \quad F_n(x) = 1 + \sum_{r \in Q} x^r.$$

При помощи многочлена этих многочленов можно построить порождающие матрицы КВ-кодов. Порождающая матрица кода $\overline{\mathcal{L}}$ равна

$$\overline{G} = \begin{pmatrix} f_0 & f_1 & \cdots & f_{p-1} \\ f_{p-1} & f_0 & \cdots & f_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ f_1 & f_2 & \cdots & f_0 \end{pmatrix}$$

где f_i – коэффициенты многочлена F_q.

Порождающая матрица кода $\mathcal{L}$ равна

$$G = \left(\begin{array}{cccc} f_0 & f_1 & \cdots & f_{p-1} \\ f_{p-1} & f_0 & \cdots & f_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ f_1 & f_2 & \cdots & f_0 \\ \hline 1 & 1 & \cdots & 1 \end{array}\right)$$

Аналогичные равенства имеют место и для кодов $\mathcal{N}$ и $\overline{\mathcal{N}}$.

2.6 Расширенные квадратично-вычетные коды

Каждая из матриц G и H состоит из набора линейно независимых векторов, то есть каждая из них может порождать линейное подпространством а как следствие,

может порождать линейный код. Так же, обратив внимание на выражение этих матриц на странице 5, можно сделать вывод, что эти матрицы могут заменить друг друга, породив при этом код, который называется дуальным.

Определение 19. [3] Если C – линейный (n, k)-код над полем $GF(l)$, то дуальный или ортогональный к нему код $C^{\perp}$ определяется как множество всех векторов, ортогональных всем кодовым словам кода C:

$$C^{\perp} = \{u | u \cdot v = 0, \forall v \in C\}.$$

Здесь ортогональность векторов определяется как $u \cdot v = 0$, где $(\cdot)$ – скалярное произведение.

КВ-код можно расширить путем добавления общей проверки на четность таким образом, что

$$\left.\begin{array}{ll} \hat{\mathcal{L}}^{\perp}=\hat{\mathcal{L}};\ \hat{\mathcal{N}}^{\perp}=\hat{\mathcal{N}}, & p = 4k-1; \\ \hat{\mathcal{L}}^{\perp}=\hat{\mathcal{N}}, & p = 4k+1, \end{array}\right\}$$

где " $\wedge$ " обозначает расширенный код.

Если $a = (a_0, \ldots, a_{p-1})$ – слово кода $\mathcal{L}$ (или $\mathcal{N}$) и $p = 4k - 1$, то расширенный код получается путем дописывания позиции

$$a_{\infty} = -y \sum_{i=0}^{p-1} a_i,$$

где $1 + y^2 p = 0$.

2.7 Группа автоморфизмов квадратично-вычетных кодов

Определение 20. [7] Пусть p – простое число вида $8m \pm 1$. Совокупность всех подстановок на множестве $0, 1, 2, \ldots, p-1, \infty$ вида

$$y \to \frac{ay + b}{cy + d},$$

где $a, b, c, d \in GF(p)$ и $ad - bc = 1$, образует группу, которая называется проективной специальной линейной группой $PSL_2(p)$ или дробно-линейной группой.

Теорема 9. [7] *Группа $PSL_2(p)$ порождается тремя подстановками:*

$$\left.\begin{array}{l} S : y \to y + 1; \\ V : y \to \rho^2 y; \\ T : y \to -\left(\frac{1}{y}\right), \end{array}\right\}$$

где ρ – примитивный элемент поля $GF(p)$.

Теорема 10. *Группа $PSL_2(p)$ состоит из $\frac{p(p^2-1)}{2}$ подстановок вида:*

$$V^i S^j : y \to \rho^{2i} y + j;$$

$$V^i S^j T S^k : y \to k - \left(\rho^{2i} y + j\right).$$

Теорема 11. *(Глизон и Прэндж)*[8] *Если $l = 2$ и $p = 8m \pm 1$, то расширенный квадратично-вычетный код $\hat{\mathcal{L}}$ инвариантен относительно группы $PSL_2(p)$.*

Под словом "инвариантен" понимается, что перестановка символов любого кодового слова кода $\hat{\mathcal{L}}$ согласно подстановке из группы $PSL_2(p)$, так же будет являться кодовым словом кода $\hat{\mathcal{L}}$.

Из теоремы (11) следует тот факт, что группа автоморфизмов расширенного КВ-кода ($Aut(\hat{\mathcal{L}})$) содержит $PSL_2(p)$. Известны три случая, когда $Aut(\hat{\mathcal{L}})$ на самом деле больше группы $PSL_2(p)$, а именно при $l = 2$ и $p = 7$ код равен $(8, 4)$-расширенному коду Хемминга. Два других случая получаются при $l = 2$ и $p = 23$ а так же при $l = 3$ и $p = 11$, то есть в случае расширенного кода Голея. Но представляется очень вероятным, что для остальных значений l и p группа $Aut(\hat{\mathcal{L}})$ равна группе $PSL_2(p)$. Это предположение выполняется во многих случаях, например, имеет место следующий результат.

Теорема 12. *(Ассмус и Мэттсон)*[3] *Если $(p-1)/2$ – простое число и $5 < p \leq 4079$, то, за исключением трех случаев, группа $Aut(\hat{\mathcal{L}})$ равна (изоморфна) группе $PSL_2(p)$.*

3 Исследование способов определения точного минимального расстояния квадратично-вычетных кодов

Как уже было сказано во введении, минимальное расстояние кода является главной характеристикой кода в помехоустойчивом кодировании. В зависимости от минимального расстояния вычисляется количество ошибок, которое код способен обнаружить или исправить, вероятность определения ошибки, оптимальность кода и другие. При сравнении двух кодов двух кодов в первую очередь сравнивают именно их минимальное расстояние, так как с ростом этого показателя повышается общая эффективность кода, понижается вероятность не обнаружить ошибку.

Следуя определению (6), можно составить следующий алгоритм вычисления минимального расстояния:

1. строим код (вычисляем все кодовые слова);
2. сравниваем все кодовые слова друг с другом и вычисляем для них расстояние Хемминга;
3. из всех вычисленных расстояний выбираем наименьшее.

Данный способ является наиболее тривиальным, и, что очевидно, наиболее неэффективным. С его помощью можно вычислять минимальные расстояния кодов только с очень маленькой длинной. В случае, если мы имеем дело с линейным кодом, данный алгоритм можно заметно упростить:

1. строим код (вычисляем все кодовый слова);
2. вычисляем вес для всех кодов слов, кроме нулевого;
3. выбираем наименьшее вычисленный вес.

Данный алгоритм напрямую следует из определения (1), так как любой линейное пространство замкнуто относительно сложения, то есть сумма любых двух векторов линейного пространства является также принадлежит этому пространству.

Этот алгоритм так же подходит только для кодов малой длинны, так как количество кодовых слов растет с экспоненциальной скоростью относительно длины кода и, например, для квадратично-выетного кода длины 73 составляет 2^{37} кодовых слов, что примерно равняется 137 миллиардам.

3.1 Использование группы автоморфизмов для вычисления минимального расстояния

Миккельтвей, Лэм, Мак-Элис предложили мощный метод нахождения весового распределения квадратично-вычетных кодов (под весовым распределением понимается характеристика кода, которая показывает количество кодовых слов, которые имеют данный вес). В его основе лежит следующее утверждение.

Лемма 4. [9] *Пусть H – некоторая подгруппа группы автоморфизмов $Aut(C)$ кода C. Если A_i – число слов веса i в коде C, а $A_i(H)$ – число слов веса i в коде C, инвариантных относительно некоторого элемента из подгруппы H, то*

$$A_i \equiv A_i(H)(mod\ |H|).$$

Обозначим через S_q максимальную подгруппу в группе $Aut(C)$, порядок которой равен степени простого числа q, делящего $|Aut(C)|$. (S_q называется силовской подгруппой группы $Aut(C)$; в случае $Aut(C) = PSL_2(p)$ эти подгруппы имеют очень простую структуру.) Заставляя H пробегать по подгруппам Силова S_q, по лемме (4) можно определить все величины $A_i(mod\ q^a)$ и, следовательно, по китайской теореме об остатках, все величины $A_i(mod\ |Aut(C)|)$. При больших значениях $|Aut(C)|$ это часто позволяет точно найти величины A_i.

Следовательно, алгоритм будет выглядеть следующим образом:

1. строим группу автоморфизмов $Aut(C) = PSL_2(p)$;
2. находим все подгруппы Силова (S_q) данной группы автоморфизмов;
3. находим все слова, инвариантные относительно некоторого элемента из S_q;
4. считаем $A_i(S_q)(mod\ |S_q|)$;
5. считаем $A_i(Aut(C))(mod\ |Aut(C)|)$;
6. считаем A_i.

Данный алгоритм уже успешно применялся в [9], [10], [11].

Следуя теореме 10, первый пункт выполняется достаточно легко и быстро.

Что бы найти подгруппы Силова, разбиваем $|PSL_2(p)|$ на степени простых чисел: $|PSL_2(p)| = q_1^{i_1} \cdot q_2^{i_2} \cdot \ldots \cdot q_k^{i_k}$. Затем для каждого q_i находим свой образующий элемент и строим подгруппу Силова (S_q).

В третьем пункте[9] необходимо выделить не единичный элемент из подгруппы Силова (g) и найти все слова, инвариантные относительно данной перестановки. Далее необходимо составить систему уравнений, состоящую из уравнения проверки на четность и $p+1$-го уравнения $c_u = c_{g \cdot u}$, $u = 0, 1, \ldots, p = 1, \infty$, где $g \cdot u$ – действие

перестановки g на элемент u из вектора c. Затем необходимо решить составленную систему.

В четвертом пункте, после нахождения всех решений системы уравнений, считаем количество слов каждого веса.

Пятый пункт легко вычисляется при помощи китайской теоремы об остатках.

В шестом пункте нам известно $A_i(Aut(C))$, а так же известна формула $A_i \equiv A_i(Aut(C))(mod\ |Aut(C)|)$, то есть $A_i = A_i(C) + |Aut(C)| \cdot n$, где n – некоторое натуральное число или 0. Для нахождения этого n составляется весовой многочлен квадратично-вычетного кода и используется алгоритм из линейного программирования.

Данный алгоритм достаточно успешно используется для нахождения весового спектра квадратично-вычетного кода (см. [9], [11]), однако, попробуем упростить его таким образом, чтобы он давал ответ только о минимальном расстоянии кода:

1. строим группу автоморфизмов $Aut(C) = PSL_2(p)$;
2. находим все подгруппы Силова (S_q) данной группы автоморфизмов;
3. находим все слова, инвариантные относительно некоторого элемента из S_q;
4. считаем вес каждого слова и выбираем наименьшее число, кроме нуля.

В этом алгоритме есть недостаток, который заключается в том, что третий пункт может не дать нам слова минимального веса, то есть $A_i(Aut(C)) = 0$, где i – минимальный вес, следовательно $A_i = A_i(Aut(C)) + |Aut(C)| \cdot n = |Aut(C)| \cdot n$. И, как следствие, n здесь не будет равно нулю. Следуя из этого, можно только утверждать, что алгоритм дает лишь верхнюю границу для минимального расстояния кода. Однако, обратив внимание на примеры, можно увидеть, что $A_i(Aut(C))$, где i – минимальный вес, практически никогда не равно нулю, из чего можно предположить, что алгоритм всегда будет выдавать верный ответ.

В ходе выполнения дипломной работы алгоритм был частично реализован. Была построена функция для вычисления $PSL_2(p)$. За счет преобразования вычислений, а так же хранения промежуточных результатов, в настоящий момента алгоритм работает приметно в 30 раз быстрее, чем в изначальном варианте. Для построения группы $PSL_2(32)$ уходит примерно 1 секунда и примерно 13 мегабайт памяти, для построения группы $PSL_2(74)$ затрачивается примерно 30 секунд и примерно 365 мегабайт памяти.

Построение подгрупп Силова на данный момент вызывает наибольшее количество вопросов, так как эта задача является весьма нетривиальной. К сожалению, данный пункт алгоритма на данный момент реализован не был.

Из-за отсутствия реализации второго пункта дальнейшая разработка алгоритма вызывает некоторые затруднения, и, возможно, даже бессмысленность. Однако,

выбирая в качестве примера случайные элементы из $PSL_2(p)$, можно сделать вывод, что третий пункт алгоритма вполне реализуем и работает достаточно быстро (для $p = 74$ находит решение системы уравнений примерно за 0.01 секунды), а так же действительно выдает кодовые слова, в том числе и слова минимального веса.

Так как данный алгоритм не был реализован, достаточно трудно говорить о его преимуществах или недостатках. Однако, обратив внимание на текущую реализацию, можно сказать, что алгоритм обладает достаточно высокой скоростью работы. К недостаткам стоит отнести вероятность получения неправильного ответа. Окончательная эффективность алгоритма зависит от того, будет ли найден алгоритм для эффективного автоматического вычисления подгрупп Силова. В случае, если алгоритм найден не будет, то, для обнаружения слов минимального веса, придется выполнять предварительную ручную работу по вычислению подгрупп Силова. Если алгоритм будет найден, то итоговая эффективность скорее всего будет зависеть от него, так как реализованные на данный момент этапы не являются слишком трудными для вычислительной машины.

3.2 Использование теории норм синдромов для вычисления минимального расстояния

Полностью теория норм синдромов будет раскрыта в следующей главе. В данном разделе мы лишь частично затронем данную теорию, чтобы описать достаточно эффективный способ обнаружения слов минимального веса.

Как уже говорилось в теореме (5), синдромы всех ошибок, которые линейный код способен исправить – различны. Из чего следует следующий алгоритм нахождения минимального веса:

1. $i = 0$;
2. $i++$;
3. перебираем все ошибки весом i и считаем их синдромы ($s = He$);
4. если все вычисленные за время работы алгоритма синдромы различные то, возвращаемся в пункт 2;
5. если нашелся синдром, который уже был вычислен, то количество ошибок, которые способен исправить код равняется $i - 1$.

Однако, в теории норм синдромов есть результат, который заключается в том, что все ошибки можно собрать в непересекающиеся Г-орбиты. Так мы рассматриваем квадратично-вычетные коды, то длина вектора ошибки простая, и, следовательно, в каждой Г-орбите будет находится p векторов. Оставив в каждой орбите только образующий элемент, мы уменьшим количество векторов ошибок в p раз.

Так же каждой Г-орбите соответствует свой вектор, который называется нормой синдрома. Данный вектор имеет весьма простой алгоритм для вычисления. Для примитивных БЧХ-кодов все нормы синдромов, вычисленные от ошибок, которые код способен исправить, – уникальны. Поэтому возникает следующий алгоритм вычисления минимального расстояния кода:

1. $i = 0$;
2. $i++$;
3. перебираем все Г-орбиты весом i и считаем их нормы синдромов;
4. если все вычисленные за время работы алгоритма нормы синдромов различные то, возвращаемся в пункт 2;
5. если нашлась норма синдрома, которая уже был вычислена, то количество ошибок, которые способен исправить код равняется $i - 1$.

Однако, в случае, если квадратично-вычетный код имеет длину $p \neq 2^k - 1$, то может возникнуть ситуация, что нормы синдромов, вычисленные от разных ошибок, которые код способен исправить, будут совпадать. В таком случае необходимо сравнивать синдромы ошибок (данный этап не вызовет увеличения времени работы алгоритма, потому что синдромы уже вычислены, так как для вычисления нормы синдрома необходимо знать зам синдром). В силу данного замечания алгоритм преобразуется следующим образом:

1. $i = 0$;
2. $i++$;
3. перебираем все Г-орбиты весом i и считаем их нормы синдромов;
4. если все вычисленные за время работы алгоритма нормы синдромов различные то, возвращаемся в пункт 2;
5. если нашлась норма синдрома, которая уже был вычислена, то сравниваем синдромы ошибок. Если синдромы различны, то возвращаемся в пункт 2;
6. если синдромы ошибок совпадают, то количество ошибок, которые код способен исправить равняется $i - 1$.

К недостаткам алгоритма стоит отнести быстрое увеличение объемов работы в зависимости от длины кода. Например, для квадратично-вычетного кода длины p общее количество Г-орбит составляет 161, для квадратично-вычетного кода длины 73 – 2,5 миллиона. На такой быстрый рост влияет не только увеличение длины кода, но так же и увеличение минимального расстояния, потому что количество

ошибок веса $i+1$ (C_p^{i+1}/p) гораздо больше чем количество ошибок веса i (C_p^i/p) (для $i < p/2$).

Так же стоит отметить, что объем вычислений неизвестен. Если мы не знаем минимального расстояния кода, ты мы и не знаем когда алгоритм закончит свою работу, при этом каждый следующий шаг алгоритма будет работать в разы дольше чем предыдущий.

Преимуществом алгоритма является простота реализации. Так же вычисленные во время работы Г-орбиты необходимы для реализации алгоритма декодирования квадратично-вычетного кода.

4 Исследование норменных и полиномиальных инвариантов орбит ошибок квадратично-вычетных кодов

4.1 Многообразие ошибок линейных кодов, их веса и диаметры

Определение 21. [12] (Циклическим) пакетом ошибок длиной b называется вектор e, все ненулевые координаты которого расположены среди b последовательных (по циклу) координат, первая и последняя из которых отличны от нуля. Пакет ошибок называется сплошным, если вес ошибки совпадает с длиной пакета.

Пример 5. *Построим все возможные пакеты ошибок длиной 3 в пространстве* P_6, *где* $P = GF(2)$.

1	1	1	0	0	0	0	1	1	1	0	0	0	0	1	1	1	0
0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	1
1	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0
0	0	0	1	0	1	1	0	0	0	1	0	0	1	0	0	0	1

Определение 22. [12] Модулем ошибок длиной b (b делит n) называется такой вектор ошибок $e = (e_1, e_2, \dots e_n)$, у которого $kb + 1$-ая и $kb + b$-ая координаты отличны от нуля для некоторого целого k, $0 \leq k \leq n/b$, а так же возможно отличны от нуля некоторые координаты e_i, $kb+2 \leq i \leq kb+b-1$, а все отсальные координаты раны нулю.

Из определения следует, что, если координаты модульного вектора-ошибки разбиты последовательно на $k = n/b$ блоков длиной b, то ненудевыми могут быть лишь координаты в одном из таких подблоков.

Пример 6. *Изобразим модули ошибок длиной 4 в пространстве* P_8 *над полем* $GF(2)$.

1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1
1	1	0	1	0	0	0	0	0	0	0	0	1	1	0	1
1	0	1	1	0	0	0	0	0	0	0	0	1	0	1	1
1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1

Определение 23. Всякую вектор-ошибку e весом $wt(e) > 1$ можно интерпретировать различным образом как пакетную ошибку с соответствующими значениями длины b. Наименьшую из длин b при всех таких интерпретациях вектора-ошибки e назовем диаметром D этой ошибки.

Пример 7. *Вектор-ошибка* $e = (100010)$ *– есть ошибка веса* $wt(e) = 2$ *в кодовом слове длиной* $n = 6$*, его можно рассматривать как пакет ошибок длиной 5 или пакет длиной 3. таким образом, диаметр* D *этого вектора-ошибки равен 3.*

Предложение 5. [12] *Диаметр вектора-ошибки* e *весом* $wt(e) > 1$ *с ненулевыми координатами на позициях* $i_1, i_2, \ldots, i_\omega$ *вычисляется по формуле*

$$D(e) = min\{i_\omega - i_1 + 1, n + i_1 - i_2 + 1, n + i_2 - i_3 + 1, \ldots, n + i_{\omega-1} - i_\omega + 1\}.$$

В частности, при $wt(e) = 2$

$$D(e) = min\{i_2 - i_1 + 1, n + i_1 - i_2 + 1\};$$

при $wt(e) = 3$

$$D(e) = min\{i_3 - i_1 + 1, n + i_1 - i_2 + 1, n + i_2 - i_3 + 1\}.$$

Следствие 3. *Диаметры векторов-ошибок весом 2 в точности принадлежат отрезку* $[2; \lfloor n/2 \rfloor + 1]$.

4.2 Циклическая классификация векторов-ошибок

Рассмотрим действие на пространстве ошибок P_n подгруппы симметрической группы, порожденной подстановкой

$$\sigma = (1, 2, \ldots, n) = \begin{pmatrix} 1 & 2 & 3 & \ldots & n-1 & n \\ 2 & 3 & 4 & \ldots & n & 1 \end{pmatrix}.$$

Минимальная подгруппа $\Gamma =< \sigma >$ симметрической группы S_n, содержащая σ, состоит из степеней этой подстановки: $\Gamma = \{\sigma, \sigma^2, \ldots, \sigma^n = id\}$, где id – тождественная подстановка. Следовательно, это циклическая группа порядка $|\Gamma| = n$.

Действие циклической подстановки $\sigma = (1, 2, \ldots, n)$ на произвольный вектор $e = (e_1, e_2, ..., e_n)$ пространства P_n приводит к вектору ошибке $\sigma(e) = (e_n, e_1, e_2, \ldots, e_{n-1})$, который получается из e циклическим сдвигом всех координат вправо на одну позицию. Отсюда следует, что k-кратное действие σ на e, $1 \leq k \leq n-1$, есть вектор-ошибка $\sigma^k(e) = (e_{n-k+1}, e_{n-k+2}, \ldots, e_n, e_1, e_2, \ldots, e_{n-k})$. Очевидно, $\sigma^n(e) = e$. Может случится, что $\sigma^\lambda(e) = e$ для некоторого целого $\lambda, 1 < \lambda < n$.

Определение 24. [12] Совокупность всех попарно различных векторов-ошибок $\sigma^k(e)$,
$0 \leq k < n$, для данного фиксированного вектора e называется Γ-орбитой или σ-орбитой вектора0ошибки e при действии Γ на множестве P_n и обозначается через $< e >$.

Теорема 13. [12] *Для произвольного фиксированного вектора* e *из пространства ошибок* P_n *его* Γ*-орбита* $< e >$ *состоит из* λ *элементов, где* $\lambda = n$ *или* λ *делит* n.

При этом λ – наименьшее натуральное число с условием $\sigma^{\lambda}(e) = (e)$ и Г-орбита $< e >$ имеет следующую структуру:

$$< e >= \{e, \sigma(e), \ldots, \sigma^{\lambda-1}(e)\}.$$

Все векторы-ошибки из $< e >$ имеют одинаковый диаметр.

Следствие 4. [12] *пусть в условиях теоремы вектор e имеет вес $\omega, 1 \leq \omega \leq n-1$, а $\lambda = n$. Тогда Г-орбита $< e >$ содержит в точности ω векторов с отличной от нуля i-ой координатой для каждого фиксированного целого i, $i \leq i \leq n$.*

Предложение 6. [12] *Для любых двух векторов-ошибок e и e' из P_n их Г-орбиты $< e >$ и $< e' >$ либо совпадают, либо не имеют общих элементов.*

Пример 8. *Построим Г-орбиту в пространстве P_{15}, порожденной вектором-ошибки веса 6.*

1	1	0	0	0	1	1	0	0	0	1	1	0	0	0
0	1	1	0	0	0	1	1	0	0	0	1	1	0	0
0	0	1	1	0	0	0	1	1	0	0	0	1	1	0
0	0	0	1	1	0	0	0	1	1	0	0	0	1	1
1	0	0	0	1	1	0	0	0	1	1	0	0	0	1

Из теоремы (13) и предложения (6) следует, что под действием группы Г циклических сдвигов пространство P_n разбивается на непересекающиеся классы – Г-орбиты. Всякое разбиение множества на непересекающиеся классы определяет отношение эквивалентности на нем. Множество всех Г-орбит пространства P_n будем обозначать через P_n/Γ.

Таким образом, приходим к следующей естественной классификации двоичных векторов-ошибок: два вектора из P_n называются эквивалентными, принадлежащими одному классу эквивалентности, если они переходят друг в друга под действием циклических сдвигов.

Пример 9. *Построим все Г-орбиты на двоичном 4-мерном пространстве векторов-ошибок.*

$$\begin{array}{lcl} e_0 = (0000) & \Rightarrow & < e_0 >= \{(0000)\} \\ e_1 = (1000) & \Rightarrow & < e_1 >= \{(1000, 0100, 0010, 0001)\} \\ e_2 = (1100) & \Rightarrow & < e_2 >= \{(1100, 0110, 0011, 1001)\} \\ e_3 = (1010) & \Rightarrow & < e_3 >= \{(1010, 0101)\} \\ e_4 = (1110) & \Rightarrow & < e_4 >= \{(1110, 0111, 1011, 1101)\} \\ e_5 = (1111) & \Rightarrow & < e_5 >= \{(1111)\} \end{array}$$

Г-орбиты пространства P_n, содержащие по n элементов называются полными, а остальные неполные. В примере (9) три Г-орбиты являются полными, и три неполными, причем две из них $(e0, e5)$ имеют мощность 1.

Предложение 7. [12] *В любом двоичном пространстве P_n имеются лишь две Γ-орбиты мощности 1 – это $< 0 >$ и $< 1 >$. Если $n = p$ – простое число, то все остальные Γ-орбиты являются полными (то есть содержат по n векторов). Всего в пространстве P_p имеется $\Pi_p = \frac{2^p-2}{p}$ полных Γ-орбит.*

Заметим, что согласно малой теореме Ферма число 2^{p-1} сравнимо в 1 по модулю p для простого числа $p > 2$, поэтому величина $\frac{2(2^{p-1}-1)}{p} = \Pi_p$ есть целое число.

Теорема 14. *Вектор-ошибка $e = (e_1, e_2, \ldots, e_n)$ из пространства P_n называется b-периодическим для делителя b числа n, если его координаты удовлетворяют условию:*

$$e_{ib+j} = e_j$$

для произвольных целых i и j, где $1 \leq j \leq b, 1 \leq i \leq r-1, rb = n$.

Лемма 5. *Пусть e – b-периодическая вектор-ошибка, m – вектор из P_n у которого первые b координат совпадают с координатами вектора e, а остальные равны нулю. тогда*

$$e = m + \sigma^b(m) + \sigma^{2b}(m) + \ldots + \sigma^{(r-1)b}(m).$$

Теорема 15. [12] *Если вектор-ошибка g не является периодическим, то Γ-орбита $< g >$ является полной. В частности, если вес ω вектора g взаимно прост с длиной n, то Γ-орбита $< g >$ – полная.*

4.3 Действие циклотомических подстановок на пространстве ошибок двоичных кодов

Определение 25. [12] Преобразование $\varphi : T \to T$, где $T = \{1, 2, \ldots, n\}$ задается следующим образом:

$$\varphi(i) = \begin{cases} 2i-1, & 2i-1 \leq n \\ 2i-1-n, & 2i-1 > n \end{cases}, \ \forall i \in T.$$

Лемма 6. [12] *Отображение φ является биекцией множества T тогда и только тогда, когда n нечетно.*

Найдем порядок циклической подгруппы $< \varphi >$, порожденной степенями подстановки φ в симметрической группе S_n. Из определения действия $< \varphi >$ следует, что для всякого целого $k \geq 1$ $\varphi^k(1) = 1$, а для каждого $i \in T, i > 1$, справедливы следующие соотношения:

$$\begin{aligned} &\varphi^2(i) = \overline{2(2i-1)-1} = \overline{2^2 i - 2 - 1}; \\ &\varphi^3(i) = \overline{2(2^2 i - 2 - 1) - 1} = \overline{2^3 i - 2^2 - 2 - 1}; \\ &\vdots \\ &\varphi^k(i) = \overline{2^k i - 2^{k-1} - \ldots - 1} = \overline{2^k i - (2^k - 1)} = \overline{(2^k - 1)(i-1) + i} \end{aligned} \tag{6}$$

Здесь $\overline{a} \in T$, то есть $\overline{a} = a - nq$ для подходящего целого q. Согласно теореме Эйлера $2^{\varphi(n)} \equiv 1(mod\ n)$, где $\varphi(n)$ – количество натуральных чисел, меньших n и взаимно простых с n. Следовательно, существует наименьшее натуральное число m с условием: $2^m - 1 = nq$, то есть n делит $2^m - 1$. Тогда из формулы (6) вытекает, что $\varphi^m(i) = i$ и, следовательно, $| < \varphi > | \leq m$.

Предложение 8. [12] *Циклическая группа* Φ*, порожденная степенями* φ *конечна и имеет порядок* m*.*

Группа Φ действует на пространстве ошибок P_n любого двоичного линейного кода, переставляя координаты веторов=-ошибок в соответствии с действием на их номера, образующие множество T. Вообще говоря, здесь уместно было бы нумеровать координаты не с 1-й по n-ую, а с 0-й по $n-1$-ую. Тогда правила действия φ и ее степеней на i-ую координату, $0 \leq i \leq n-1$, выглядят проще: $\varphi(i) = 2i(mod\ n)$. Соответственно, $\varphi^k(i) = 2^k i(mod\ n)$. При этом заметим, что числа $i, 2i, 2^2 i, \ldots, 2^{m-1} i$ образуют циклотомический класс по модулю n. поэтому подстановки $\varphi, \varphi^2, \ldots, \varphi^m = id$ – называются циклотомическими и, соответственно, группа Φ – циклотомической.

Определение 26. [12] Совокупность всех различных векторов $\varphi^s(e), 0 \leq s \leq m-1$, для данного фиксированного вектора $e \in P_n$ называется Φ-орбитой или циклотомической орбитой вектора e при действии группы Φ на пространстве P_n и обозначается через $< e >_\Phi$.

Предложение 9. [12] *Для произвольного фиксированного вектора* $e \in P_n$ *его* Φ*-орбита* $< e >_\Phi$ *состоит из* μ *элементов,* $\mu = m$ *или* $\mu | m$*. При этом* μ *– наименьшее натуральное число с условием* $\varphi^\mu(e) = e$*, а* Φ*-орбита* $< e >_\Phi$ *имеет следующую структуру:* $< e >_\Phi = \{e, \varphi(e), \ldots, \varphi^{\mu-1}(e)\}$*. Все векторы из* $< e >_\Phi$ *имеют одинаковый вес.*

Естественно Φ-орбиты пространства P_n называть полными, если они содержат по m элементов, а остальные – неполными.

Пример 10. *Выпишем* Φ*-орбиты векторов весом 1 в пространстве* P_{15} *(через* (i) *будем обозначать вектор с единицей на* i*-том месте).*

$$< (1) >_\Phi = \{(1)\};$$
$$< (2) >_\Phi = \{(2), (3), (5), (9)\};$$
$$< (4) >_\Phi = \{(4), (7), (13), (10)\};$$
$$< (6) >_\Phi = \{(6), (11)\};$$
$$< (8) >_\Phi = \{(8), (15), (14), (12)\}.$$

Лемма 7. *Для произвольного* $e \in P_n$ $\varphi(\sigma(e)) = \sigma^2(\varphi(e))$.

Предложение 10. [12] *Пусть* $J = \{e, \sigma(e), \ldots, \sigma^{l-1}(e)\}$*, где* $l = n$ *или* $l | n$ *– Г-орбита векторов-ошибок из* P_n*, а* $\varphi(J) = \{\varphi(e), \varphi(\sigma(e)), \ldots, \varphi(\sigma^{l-1}(e))\}$*. Тогда* $\varphi(J)$ *– также Г-орбита векторов из* P_n*.*

Следствие 5. *Группа* $\Phi = \{\varphi, \varphi^2, \ldots, \varphi^m = e\}$ *является группой подстановок на множестве* P_n/Γ *всех* Γ*-орбит пространства* P_n.

Предложение 11. *Циклическая и циклотомическая подстановки* σ *и* φ, *порождают некоммутативную группу* G *подстановок на координатах пространства* P_n *нечетной размерности* n. *Она состоит из подстановок* $\varphi^j\sigma^i, 0 \le i \le n-1$, $0 \le j \le m-1$. *Умножение подчинено тождествам*

$$\varphi\sigma = \sigma^2\varphi, \sigma^n = e, \varphi^m = e.$$

Порядок этой группы равен $|G| = mn$.

Определение 27. Два вектора f и g из P_n называются G-эквивалентными, если найдется подстановка $\tau = \varphi^j\sigma^i \in G$, такая что $g = \tau(f)$.

Определение 28. G-орбитой называется совокупность всех попарно G-эквивалентных между собой векторов-ошибок из P_n.

Предложение 12. [12] *Пусть* $< e >$ – Γ*-орбита, порожденная вектором* $e \in P_n$. *Тогда* G*-орбита* $< e >_G$ *состоит из всех векторов, принадлежащих всем* Γ*-орбитам из* Φ*-орбиты* $< e >_G = \{< e >, \varphi < e >, \varphi^2 < e >, \ldots, \varphi^{\mu-1} < e >\}$.

Пример 11. *Девять полных* Γ*-орбит ошибок веса* $1-3$ *из пространства* P_7 *делится на* 5 G*-орбит:*

$$\begin{array}{ll} < (1) >_G & = \ < (1) >; \\ < (1,2) >_G & = \{< (1,2) >, < (1,3) >, < (1,4) >\}; \\ < (1,2,3) >_G & = \{< (1,2,3) >, < (1,3,5) >, < (1,2,5) >\}; \\ < (1,2,4) >_G & = \ < (1,2,4) >; \\ < (1,3,4) >_G & = \ < (1,3,4) > . \end{array}$$

Таким образом, можно сделать вывод, что циклотомические подстановки переводят друг в друга Γ-орбиты ошибок одинакового веса, образуя G-орбиту. Таким образом, можно полностью восстановить G-орбиту, имея только одного представителя орбиты, используя mn подстановок из предложения (11).

4.4 Спектры синдромов и синдромные признаки полных Γ-орбит

Предложение 13. [12] *Пусть* e – *произвольный вектор ошибок в БЧХ-коде* C *с проверочной матрицей (3). Пусть* $S(e) = (s_1, s_2, \ldots, s_{\delta-1})$*-синдром вектора* e. *Тогда*

$$S(\sigma(e)) = (\alpha^b s_1, \alpha^{b+1} s_2, \ldots, \alpha^{b+i-1} s_i, \ldots, \alpha^{b+\delta-2} s_{\delta-1}). \qquad (7)$$

Следствие 6. [12] *в условиях предложения (13) для произвольного целого* λ *синдром*

$$S(\sigma^\lambda(e)) = (\alpha^{\lambda b} s_1, \alpha^{\lambda(b+1)} s_2, \ldots, \alpha^{\lambda(b+i-1)} s_i, \ldots, \alpha^{\lambda(b+\delta-2)s_{\delta-1}}).$$

Следствие 7. *Если какая-либо из компонент s_i, $1 \le i \le \delta - 1$, синдрома $S(e)$ равна 0 (отлично от нуля), то и у всех векторов-ошибок из Γ-орбиты $< e >$ соответствующая компонента синдрома равна нулю (отлична от нуля).*

Следствие 8. *Если дан КВ-код с проверочной матрицей (5), то*

$$S(\sigma(e)) = (\beta^{q_1} s_1, \beta^{q_2} s_2, \ldots, \beta^{q_{(n/2)}} s_{(n/2)}).$$

Определение 29. [12] Спектром синдромов в коде C Γ-орбиты J назовем множество синдромов всех векторов-ошибок из J в этом коде и обозначим через $S(J)$. Спектр $S(J)$ будем называть полным, если его мощность совпадает с мощностью Γ-орбиты J: $|S(J) = |J|$, в противном случае $S(J)$ будем называть неполным.

Предложение 14. [12] *Пусть в БЧХ-коде C с проверочной матрицей (3) вектор-ошибка e имеет синдром $S(e) = (s_1, s_2, \ldots, s_{\delta-2})$. Спектр синдромов Γ-орбиты $J =< e >$ состоит из всех попарно различных векторов пространства $S(P_n)$ вида*

$$(\alpha^{\lambda b} s_1, \alpha^{\lambda(b+1)} s_2, \ldots, \alpha^{\lambda(b+j-1)} s_j, \ldots, \alpha^{\lambda(b+\delta-2)} s_{\delta-1}), 0 \le \lambda \le n-1. \quad (8)$$

Пусть синдром $S(e)$ имеет компоненту $s_j \neq 0$ для целого j, $1 \le j \le \delta - 1$, что НОД$(b + j - 1, n) = 1$. Тогда $S(J)$ содержит n попарно различных синдромов и, следовательно, $|S(J)| = |J| = n$.

Следствие 9. *Пусть J – неполная Γ-орбита векторов-ошибок в коде C. Тогда у синдромов всех векторов этой Γ-орбиты координаты $s_i = 0$ для каждого целого i, $0 \le i \le \delta - 1$, с условием: $b + i - 1$ взаимно просто с n.*

Замечание 3. *Предложение (14) утверждает, что, как и векторы каждой Γ-орбиты J, спектр синдромов $S(J)$ произвольной Γ-орбиты J векторов0ошибок можно сконструировать по формуле (8) из синдрома $S(e)$ любого вектора $e \in J$. Этот факт можно выразить следующим равенством: $S(< e >) =< S(e) >$.*

4.5 Определение нормы синдрома ошибок в произвольном БЧХ-коде

Определение 30. [12] Нормой синдрома $S(e) = (s_1, s_2, \ldots, s_\delta)$ вектора ошибок e в коде C называется вектор $N(S(e)) = (N_{12}, N_{13}, \ldots, N_{1(\delta-1)}, N_{23}, \ldots, N_{(\delta-2)(\delta-1)})$ с $C^2_{\delta-1}$ координатами N_{ij}, $1 \le i < j \le \delta - 1$, которые вычисляются по формулам:

$$N_{ij} = \frac{s_j^{(b+i-1)/h_{ij}}}{s_i^{(b+j-1)/h_{ij}}}; \quad (9)$$

$N_{ij} = \infty$, если $s_j \neq 0$, $s_i = 0$;

$N_{ij} = -$ (не существует), если $s_i = s_j = 0$;

где h_{ij} =НОД$(b + i - 1, b + j - 1)$.

Из определения следует, что координаты N_{ij} нормы $N(S)$ могут принимать $q^m + 2$ значения: либо какой-нибудь из q^m элементов поля $GF(q^m)$, либо ∞, либо $-$ (не существует).

Предложение 15. [12] *Пусть у БЧХ-кода C_δ параметр $b = 1$, а синдром $S = (s_1, s_2, \ldots, s_{\delta_1})$ имеет компоненту $s_1 \neq 0$. Тогда все координаты нормы $N(S)$, начиная с N_{23}, однозначно определяются первыми $\delta - 2$ координатами этой нормы по формулам:*

$$N_{ij} = \frac{N_{1j}^{i/h_{ij}}}{N_{1i}^{j/hij}}. \tag{10}$$

4.6 Инвариантность нормы синдрома относительно действия группы Γ циклических сдвигов

Теорема 16. [12] *Для всякого вектора ошибок e БЧХ-кода C и его синдрома $S(e)$ справедливо равенство $N(S(\sigma(e))) = N(S(e))$.*

Доказательство. Пусть $S(e) = (s_1, s_2, \ldots, s_{\delta-1})$. Пусть $s_i \neq 0, 1 \leq i < \delta-1$. Тогда i-ая компонента синдрома $S(\sigma(e))$ равна $s'_i = \alpha^{b+i-1} s_i$ согласно формуле (7) и, следовательно, $s'_i \neq 0$. Тогда для каждого j, $i < j \leq \delta - 1$, координаты N'_{ij} нормы $N(S(\sigma(e)))$ вычисляются по формуле:

$$N'_{ij} = \frac{s'^{(b+i-1)/h_{ij}}_j}{s'^{(b+j-1)/h_{ij}}_i} = \frac{\left(\alpha^{b+j-1} s_j\right)^{(b+i-1)/h_{ij}}}{\left(\alpha^{b+i-1} s_i\right)^{(b+j-1)/h_{ij}}} = \frac{s_j^{(b+i-1)/h_{ij}}}{s_i^{(b+j-1)/h_{ij}}} = N_{ij}. \tag{11}$$

Пусть $s_i = 0$, тогда и $s'_i = 0$. Если $s_j \neq 0$, то и $s'_j \neq 0$. Тогда $N_{ij} = \infty$ и $N'_{ij} = \infty$. Если же $s_j = 0$, то и $s'_j = 0$ и, следовательно N_{ij} и N'_{ij} одновременно не существуют. Теорема полностью доказана.

Следствие 10. *Если нормы двух векторов-ошибок в коде C различны, то данные векторы принадлежат различным* Γ*-орбитам.*

Определение 31. Нормой Γ-орбиты J в коде C называется норма синдрома любого вектора-ошибки из этой Γ-орбиты и обозначается через $N(J)$.

Предложение 16. [12] *Пусть J_1, J_2 – две* Γ*-орбиты векторов-ошибок в коде C с различными нормами: $N(J_1) \neq N(J_2)$. Тогда для любых векторов $g \in J_1$ и $f \in J_2$ их синдромы $S(f)$ и $S(g)$ различны. Другими словами, спектры синдромов таких* Γ*-орбит не пересекаются.*

4.7 Нормы синдромов циклотомически связанных векторов ошибок

Определение 32. Спектром норм G-орбиты $< e >_G$ называется множество норм составляющих ее Γ-орбит ($\{N_1, N_2, \ldots, N_\mu\}$) и обозначается через $N(< e >_G)$.

Спектр норм G-орбиты называется полным, если его мощность совпадает с количеством составляющих Г-орбит.

Предложение 17. [12] *Пусть* $N(< e >_G)$ *– спектр норм* G*-орбиты* $< e >_G$*. Тогда множество* $\{N_i\}$ *значений* i*-ой координаты,* $i = 1, 2, \ldots, \mu$ *в ее спектре норм* $N(< e >_G)$ *либо совпадает с одним из одноэлементных множеств* $\{0\}$, $\{\infty\}$, $\{-\}$, *либо множество показателей* $\deg N_i$, *составляет один циклотомический класс по модулю* $n = 2^m - 1$.

Определение 33. Полиномиальной нормой назовем вектор, который вычисляется по формуле:

$$P(N(S(e))) = P(\{N_{12}, \ldots, N_{(\delta-2)(\delta-1)}\}) = \\ = \left\{(x - N_{ij})(x - N_{ij})^2(x - N_{ij}^{2^2}) \ldots (x - N_{ij}^{2^{\mu-1}})\right\}^{1 \le i < j \le \delta - 1} \tag{12}$$

если $P = 0$ если $N_{ij} = \infty$.

Следствие 11. *Из предложения (17) следует, что для всех норм синдромов из спектра синдрома* $N(< e >_G)$ *полиномиальная норма будет одинаковой.*

Таким образом, в качестве элемента, характеризующего Г-орбиту можно использовать норму синдрома. Для G-орбиты такой характеристикой будет являться полиномиальная норма.

5 Разработка алгоритмов коррекции ошибок КВ-кодами на основе норменно-полиномиальных методов

5.1 Применение норм синдромов для КВ-кодов

В данном разделе пересмотрим теорию норм синдромов, введенную в предыдущем разделе, применительно к квадратично-вычетным кодам.

Для квадратично-вычетного кода с проверочной матрицей вида (5) синдром будем иметь следующий вид: $S(e) = (s_{q_1}, s_{q_2}, \ldots, s_{q_u})$, где $u = n/2$.

Норма синдрома будет считаться по следующей формуле:

$$N_{ij} = \frac{s_{q_j}^{q_i/h_{ij}}}{s_{q_i}^{q_j/h_{ij}}}; \tag{13}$$

$N_{ij} = \infty$, если $s_{q_j} \neq 0$, $s_{q_i} = 0$;

$N_{ij} = -$ (не существует), если $s_{q_i} = s_{q_j} = 0$;

где h_{ij} =НОД(q_i, q_j), $1 \leq i < j \leq u$.

Предложение 18. *Пусть $l = 2$, $p = 8l \pm 1$. Все циклотомические классы по модулю p над поле $GF(p)$, которые состоят из квадратичных вычетов по модулю числа p, имеют хотя бы одно простое число или 1.*

Данное предположение было опытным путем проверено для $7 \leq p \leq 503$.

Таким образом, в качестве представителя циклотомического класса q_i гораздо удобнее брать простое число или 1, так как пропадает необходимость в вычислении h_{ij} =НОД(q_i, q_j). Следовательно, формула (13) преобразуется к следующему виду:

$$N_{ij} = \frac{s_{q_j}^{q_i}}{s_{q_i}^{q_j}}; \tag{14}$$

$N_{ij} = \infty$, если $s_{q_j} \neq 0$, $s_{q_i} = 0$;

$N_{ij} = -$ (не существует), если $s_{q_i} = s_{q_j} = 0$;

где $1 \leq i < j \leq u$.

Как видно из формулы, для вычисления нормы синдромы нам необходимы как минимум два различных числа q_i. Однако далеко не всегда количество циклотомических классов по модулю простого числа вида $p = 8k \pm 1$ больше двух. Возьмем, например, число $p = 41$. Наименьшим число m, удовлетворяющим формуле (4)

является число $m = 20$, и оно совпадает с количеством квадратичных вычетов числа p. Таким образом, по модулю p имеется только два циклотомических класса и только один из них состоит из квадратичных вычетов. Следовательно, можно выбрать только один q_i, и как результат, мы не в состоянии вычислить ному синдрома квадратично-вычетного кода длиной 41. Наименьшими КВ-кодами, для которых можно вычислить норму синдрома являются КВ-коды длиной 31 и 73.

При разработке метода декодирования квадратично-вычетного кода так же можно избавиться от необходимости в вычислении полиномиальной нормы по формуле (12).

Определение 34. [3] Минимальным многочленом элемента β над полем $GF(p)$ называется нормированный многочлен $\mu(x)$ с коэффициентами из $GF(p)$ наименьшей возможной степени такой, что $M(\beta) = 0$.

Теорема 17. *Если i лежит в C_s, то* $M^{(i)}(x) = \prod\limits_{j \in C_S} (x - \alpha^j)$.

Как видно из теоремы, вид минимального многочлена не зависит от выбора элемента из циклотомического класса, следовательно, $M^i(x) = M^j(x) = M(x)$ для любых i, $j \in C_s$. Следовательно, минимальный многочлен может выступать в качестве характеристики для своего циклотомического класса. Обратив внимание на предложение (17) и на формулу (12) можно заметить, что полиномиальная норма есть набор минимальных многочленов, которых соответствуют циклотомическим классам по модулю $n = 2^m - 1$. Таким образом, если нам известно поле $GF(2^m)$, мы можем заранее разбить его на классы многочленов: в каждом классе у всех многочленов будет одинаковый многочлен. Таким образом, вместо вычисления многочлена по формуле (12), можно просто выбирать заранее вычисленный многочлен из поля $GF(2^m)$.

Так же очевидным является факт, что количество минимальных многочленов (циклотомических классов) поля $GF(2^m)$ меньше, чем общее количество многочленов этого класса. Поэтому вместо хранения полиномиальной нормы в виде набора минимальных многочленов можно хранить полиномиальную норму в виде порядковых номеров минимальных многочленов (циклотомических классов), что в будущем поможет немного сократить объем занимаемой памяти.

5.2 Предварительные расчеты для квадратично-вычетных кодов длины 31 и 73

Пусть $l = 2$ и $p = 31$. По формуле (4) $m = 5$. $k = 2^m - 1$ совпадает с числом p, то есть данный квадратично-вычетный код является примитивным БЧХ-кодом. $n = \frac{p-1}{m} = 6$, то есть всего имеется имеется 6 циклотомических классов. Выпишем их все:

$C_1 = \{1, 2, 4, 8, 16\}$;
$C_3 = \{3, 6, 12, 24, 17\}$;
$C_5 = \{5, 10, 20, 9, 18\}$;
$C_7 = \{7, 14, 28, 25, 19\}$;
$C_{11} = \{11, 22, 13, 26, 21\}$;
$C_{15} = \{15, 30, 29, 27, 23\}$.

По числам 1, 9 и 25, видно, что циклотомическими классами, состоящими из квадратичных вычетов, являются классы C_1, C_5 и C_7. Так как 1, 5 и 7 являются взаимно простыми числами, их можно взять в качестве q_1, q_2 и q_3.

В качестве неприводимого многочлена возьмем многочлен $g = 1 + x^2 + x^3 + x^4 + x^5$. Проверочная матрица квадратично-вычетного кода длины 31 выглядит следующим образом:

$$H_{31} = \begin{pmatrix} \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \cdots & \alpha^{26} & \alpha^{27} & \alpha^{28} & \alpha^{29} & \alpha^{30} \\ \alpha^0 & \alpha^5 & \alpha^{10} & \alpha^{15} & \alpha^{20} & \alpha^{25} & \cdots & \alpha^6 & \alpha^{11} & \alpha^{16} & \alpha^{21} & \alpha^{26} \\ \alpha^0 & \alpha^7 & \alpha^{14} & \alpha^{21} & \alpha^{28} & \alpha^4 & \cdots & \alpha^{27} & \alpha^3 & \alpha^{10} & \alpha^{17} & \alpha^{24} \end{pmatrix};$$

где α – корень неприводимого многочлена.

Пусть $l = 2$ и $p = 73$. По формуле (4) $m = 9$. $k = 2^m - 1 > p$, то есть данный квадратично-вычетный код не является примитивным БЧХ-кодом. $n = \frac{p-1}{m} = 8$, то есть всего имеется имеется 8 циклотомических классов. Выпишем их все:

$C_1 = \{1, 2, 4, 8, 16, 32, 64, 55, 37\}$;
$C_3 = \{3, 6, 12, 24, 48, 23, 46, 19, 38\}$;
$C_5 = \{5, 10, 20, 40, 7, 14, 28, 56, 39\}$;
$C_9 = \{9, 18, 36, 72, 71, 69, 65, 57, 41\}$;
$C_{11} = \{11, 22, 44, 15, 30, 60, 47, 21, 42\}$;
$C_{13} = \{13, 26, 52, 31, 62, 51, 29, 58, 43\}$;
$C_{17} = \{17, 34, 68, 63, 53, 33, 66, 59, 45\}$;
$C_{25} = \{25, 50, 27, 54, 35, 70, 67, 61, 49\}$.

В данном случае циклотомическими классами, состоящими из квадратичных вычетов, являются классы C_1, C_3, C_9 и C_{25}. Выберем в представителей циклотомических классов следующим образом $q_1 = 1, q_2 = 3, q_3 = 71, q_4 = 67$.

В качестве неприводимого многочлена возьмем многочлен $g = 1 + x^4 + x^9$. Проверочная матрица квадратично-вычетного кода длины 73 выглядит следующим образом:

$$H_{73} = \begin{pmatrix} \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \cdots & \alpha^{68} & \alpha^{69} & \alpha^{70} & \alpha^{71} & \alpha^{72} \\ \alpha^0 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} & \cdots & \alpha^{58} & \alpha^{61} & \alpha^{64} & \alpha^{67} & \alpha^{70} \\ \alpha^0 & \alpha^{71} & \alpha^{69} & \alpha^{67} & \alpha^{65} & \alpha^{63} & \cdots & \alpha^{10} & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 \\ \alpha^0 & \alpha^{67} & \alpha^{61} & \alpha^{55} & \alpha^{49} & \alpha^{43} & \cdots & \alpha^{30} & \alpha^{24} & \alpha^{18} & \alpha^{12} & \alpha^6 \end{pmatrix}.$$

5.3 Проверочная матрица, вычисление синдрома, нормы синдрома и полиномиальной нормы

Норма синдрома для квадратично-вычетного кода длиной 31 будет иметь следующий вид:

$$N_{31} = \left\{ \frac{s_2}{s_1^5}, \frac{s_3}{s_1^7}, \frac{s_3^5}{s_2^7} \right\}.$$

Для квадратично-вычетного кода длиной 73 норма синдрома вычисляется по формуле:

$$N_{73} = \left\{ \frac{s_2}{s_1^3}, \frac{s_3}{s_1^{71}}, \frac{s_4}{s_1^{67}}, \frac{s_3^3}{s_2^{71}}, \frac{s_4^3}{s_2^{67}}, \frac{s_4^{71}}{s_3^{67}} \right\}.$$

Однако, если многочлен представлен в виде коэффициентов, то операции возведения в степень и деления вычисляются очень трудно. Этот процесс можно в разы ускорить, если представить многочлены в виде логарифмов. Однако, для вычисления синдрома необходимо суммировать соответствующие столбцы проверочной матрицы, что невозможно выполнить в логарифмической форме, то есть нам так же необходимо представление многочленов в виде коэффициентов. Поэтому, для быстрой работы, мы будем хранить поле $GF(2^m)$, в которым буду хранится многочлены и соответствующие им логарифмы. Вид, в котором мы будем хранить проверочную матрицу не имеет разницы, однако для большей наглядности лучше представить проверочную матрицу с логарифмическим представлением многочленов:

$$H_{31} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & \cdots & 26 & 27 & 28 & 29 & 30 \\ 0 & 5 & 10 & 15 & 20 & 25 & \cdots & 6 & 11 & 16 & 21 & 26 \\ 0 & 7 & 14 & 21 & 28 & 4 & \cdots & 27 & 3 & 10 & 17 & 24 \end{pmatrix}.$$

$$H_{73} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & \cdots & 68 & 69 & 70 & 71 & 72 \\ 0 & 3 & 6 & 9 & 12 & 15 & \cdots & 58 & 61 & 64 & 67 & 70 \\ 0 & 71 & 69 & 67 & 65 & 63 & \cdots & 10 & 8 & 6 & 4 & 2 \\ 0 & 67 & 61 & 55 & 49 & 43 & \cdots & 30 & 24 & 18 & 12 & 6 \end{pmatrix}.$$

Таким образом, вычисление синдрома будет выполняться по следующему алгоритму:

1. получаем сигнал y и вычисляем местоположение единиц в сигнале;
2. берем соответствующие столбцы проверочной матрицы, преобразуем их к представлению в виде коэффициентов;
3. складываем все столбцы вместе;
4. полученный синдром переводим обратно к логарифмическому представлению.

Норму синдрома считаем по формуле:

$$N_{31} = \{s_2 - 5s_1, s_3 - 7s_1, 5s_3 - 7s_2\} \, (mod\ 31).$$

$$N_{73} = \{s_2 - 3s_1, s_3 - 71s_1, s_4 - 67s_1, 3s_3 - 71s_2, 3s_4 - 67s_2, 71s_4 - 67s_3\} \, (mod\ 511).$$

Однако, если мы будем считать норму синдрома N_{73} по данной формуле с соответствующей проверочной матрицей, мы заметим, что теорема (16) не работает. Это происходит из-за того, что мы построили проверочную матрицу по модулю 73, хотя работаем в поле $GF(512)$. В итоге проверочная матрица не "зацикливается" и терема перестает работать. Для устранения данной проблемы умножим проверочную матрицу на число $k/p = 7$:

$$H_{73} = \begin{pmatrix} 0 & 7 & 14 & 21 & 28 & 35 & \cdots & 476 & 483 & 490 & 497 & 504 \\ 0 & 21 & 42 & 63 & 84 & 105 & \cdots & 406 & 427 & 448 & 469 & 490 \\ 0 & 497 & 483 & 469 & 455 & 441 & \cdots & 70 & 56 & 42 & 28 & 14 \\ 0 & 469 & 427 & 385 & 343 & 301 & \cdots & 210 & 168 & 126 & 84 & 42 \end{pmatrix}.$$

Заметим, что для случая $p = 31$ $k/p = 1$ (так как код является примитивным БЧХ-кодом), поэтому подобной проблемы не возникает.

Наконец, нам остается посчитать полиномиальную норму. Для этого используем следующий простой алгоритм:

1. вычисляем норму синдрома по формуле выше;
2. для каждого элемента нормы синдрома находим циклотомический класс числа k над полем $GF(2)$, членом которого этот элемент и является.

5.4 Генерация Г-орбит и G-орбит КВ-кодов длины 31 и 73

Для работы алгоритма декодирования необходимо иметь предварительно вычисленные G-орбиты, а для построения G-орбит необходимо знать Г-орбиты. Наиболее очевидный алгоритм для нахождения Г-орбит выглядит следующим образом:

1. i=0;
2. i++;
3. строим множество E_p^i всех векторов ошибок веса i;
4. выбираем первую ошибку из E_p^i и находим все ошибки в E_n^i, которые являются циклическим смещением первой;
5. всю полученную Г-орбиту удаляем из E_p^i, один из элементов Г-орбиты сохраняем в отдельный файл вместе соответствующей с нормой синдрома;

6. если E_p^i не пусто, переходим в пункт 4;

7. если все вычисленные за время работы алгоритма нормы синдромов различные то, возвращаемся в пункт 2;

8. если нашлась норма синдрома, которая уже был вычислена, то сравниваем синдромы ошибок. Если синдромы различны, то возвращаемся в пункт 2;

9. если синдромы ошибок совпадают, то откатываем все вычисленные ошибки и нормы весом i. Конец алгоритма.

Замечание 4. *Для квадратично-вычетного кода длины 73 может случится, что нормы синдромов двух различных* Г*-орбит будут совпадать, для этого необходимо дополнительно проверять синдромы ошибок. Для примитивного БЧХ-кода все нормы синдромов будут различны, поэтому для квадратично-вычетного кода длины 31 восьмой шаг алгоритма отпадает за ненадобностью.*

Данный алгоритм обладает огромными недостатками. Во-первых во время его работы необходимо строить множество E_p^i. Например, при $p = 73$ и $i = 6$ (минимальное расстояние КВ-кода равняется 13, следовательно код способен исправить 6 ошибок) мощность множества E_p^i будет равна $C_{73}^6 = 170.230.452$, однако необходимо сохранить только 2.331.924 представителя Г-орбиты. Во-вторых сам по себе алгоритм работает очень долго.

В ходе выполнения дипломной работы был найден более простой алгоритм построения Г-орбит, не требующий вычисления множества E_p^i.

Пример 12. *Для примера, рассмотрим Г-орбиты пространства P_{13} и весом не более трех.*

Таблица 1: Г-орбиты пространства P_{13} и весом не более трех

Вес	Представитель Г-орбиты
1	(1,0,0,0,0,0,0,0,0,0,0,0,0)
2	(1,1,0,0,0,0,0,0,0,0,0,0,0) (1,0,1,0,0,0,0,0,0,0,0,0,0) (1,0,0,1,0,0,0,0,0,0,0,0,0) (1,0,0,0,1,0,0,0,0,0,0,0,0) (1,0,0,0,0,1,0,0,0,0,0,0,0) (1,0,0,0,0,0,1,0,0,0,0,0,0)

Продолжение таблицы 1

Вес	Представитель Г-орбиты
3	(1,1,1,0,0,0,0,0,0,0,0,0,0)
	(1,1,0,1,0,0,0,0,0,0,0,0,0)
	(1,1,0,0,1,0,0,0,0,0,0,0,0)
	(1,1,0,0,0,1,0,0,0,0,0,0,0)
	(1,1,0,0,0,0,1,0,0,0,0,0,0)
	(1,1,0,0,0,0,0,1,0,0,0,0,0)
	(1,1,0,0,0,0,0,0,1,0,0,0,0)
	(1,0,1,1,0,0,0,0,0,0,0,0,0)
	(1,0,1,0,1,0,0,0,0,0,0,0,0)
	(1,0,1,0,0,1,0,0,0,0,0,0,0)
	(1,0,1,0,0,0,1,0,0,0,0,0,0)
	(1,0,1,0,0,0,0,1,0,0,0,0,0)
	(1,0,1,0,0,0,0,0,1,0,0,0,0)
	(1,0,0,1,1,0,0,0,0,0,0,0,0)
	(1,0,0,1,0,1,0,0,0,0,0,0,0)
	(1,0,0,1,0,0,1,0,0,0,0,0,0)
	(1,0,0,1,0,0,0,1,0,0,0,0,0)
	(1,0,0,1,0,0,0,0,1,0,0,0,0)
	(1,0,0,0,1,1,0,0,0,0,0,0,0)
	(1,0,0,0,1,0,1,0,0,0,0,0,0)
	(1,0,0,0,1,0,0,1,0,0,0,0,0)
	(1,0,0,0,1,0,0,0,1,0,0,0,0)

Как можно увидеть в примере, все первые единицы находятся на первой позиции. Вторая единица в случае с весом 2 "доходит" до седьмой позиции. В случае с весов 3, вторая единица "доходит" позиции 5, а третья – всегда "останавливается" на позиции 9. Все эти числа можно рассчитать по формуле

$$\left\lceil \frac{i \cdot n}{p} \right\rceil \tag{15}$$

где n – длина вектора;

p – количество единиц;

i – текущая единица, не считая первой.

Таким образом, схематически алгоритм будет выглядеть таким образом:

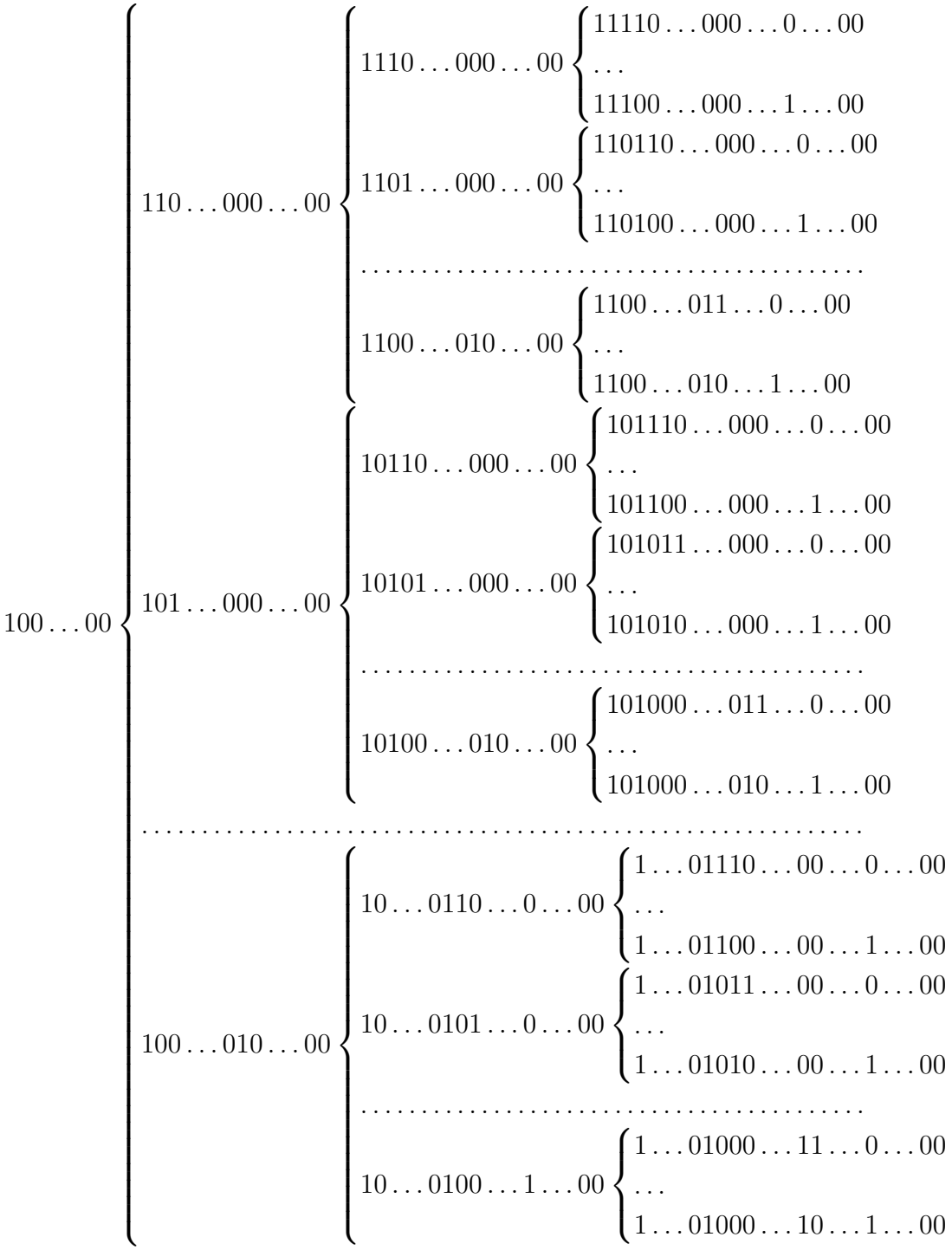

Рисунок 1: схематический алгоритм построения Г-орбиты.

Здесь последнее положение последней единицы на каждом шаге вычисляется по формуле (15).

То есть мы получаем рекурсивный алгоритм, который выглядит следующим образом:

1. строим вектор у которого на первом месте стоит единица, а все остальные элементы – нули;
2. на основании полученного вектора строим новый набор векторов: первому вектору добавляем единицу в следующий разряд сразу после последней единицы у полученного вектора, второму – через один разряд и так до разряда, вычисленного по формуле (15);
3. если в каждом векторе из нового набора векторов количество единиц меньше, чем требуется – отправляем каждый вектор в пункт 2;
4. конец алгоритма.

Построив все Г-орбиты, вычисляем у каждой из них норму синдрома и все записываем в файл.

Теперь необходимо найти все G-орбиты. Так как у нас уже есть все Г-орбиты, мы группируем все Г-орбиты по полиномиальной норме. В случае квадратично-вычетного кода длины 31, в каждой группе будет находится ровно одна G-орбита, поэтому можно оставить только один элемент и сохранить его в файл вместе с вычисленной полиномиальной нормой(см. приложение). В случае квадратично-вычетного кода длины 73, нормы синдромов могут повторятся, и, как следствие, будут повторятся полиномиальные нормы, поэтому в каждой группе может находится несколько G-орбит. Однако известно, что у всех Г-орбит, которые принадлежат одной и той же G орбите будет разная норма синдрома. Поэтому можно вычислить норму синдрома у первой Г-орбиты и оставить только те Г-орбиты, у которых норма будет совпадать с первой. И все эти представители G-орбит записываются в файл вместе с соответствующей полиномиальной нормой.

Для КВ-кода длины 73 получилось достаточно много G-орбит, поэтому для более быстрого обнаружения нужной G-орбиты была проведена индексация данных. В результате индексации по первому элементу полиномиальной нормы получилось 59 разделов, в каждом из них, при индексации по второму элементу полиномиальной нормы, получилось так же примерно по 59 разделов. И в каждом разделе находится примерно 80-90 элементов.

Результат вычисления Г- и G-орбит представлен в следующей таблице.

Таблица 2: Результат вычисления Г- и G-орбит

Длина кода	Количество Г-орбит	Занимаемая память	Количество G-орбит	Занимаемая память
31	161	18 КБ	33	3.54 КБ
73	2 553 481	632 МБ	281 422	67.7 МБ

Среди G-орбит квадратично-вычетного кода длины 73 вышло 279 097 уникальных норм, 2 313 полиномиальных норм, которым соответствуют 2 G-орбиты, 11 тройных норм и 1 норма с четырьмя орбитами.

5.5 Процесс исправления ошибок квадратично-вычетными кодами на основе норменно-полиномиальных методов

Алгоритм исправления ошибок выглядит следующим образом[13]:

1. на вход получаем сигнал y;
2. вычисляем синдром $s = H \cdot y$. Если синдром равен нулю, y – кодовое слово, конец алгоритма;
3. если синдром не равен нулю, вычисляем норму синдрома и полиномиальную норму;
4. находим G-орбиты с такой же полиномиальной нормой;
5. для первой Г-орбиты применяем циклотомическую подстановку и вычисляем норму синдрома до тех пор, пока норма синдрома не будет совпадать с нормой вектора y;
6. ко всем остальным Г-орбитам применяем такое же количество циклотомических подстановок, как и к первой;
7. среди полученного набора Г-орбит находим вектор, чей синдром совпадает с s;
8. конец алгоритма.

Распишем подробнее седьмой пункт алгоритма. Из предложения (14) можно сделать вывод, что, если разделим синдром $s(\sigma^{\lambda}(y))$ на синдром $s(y)$, то мы получим вектор $(\alpha^{\lambda b}, \alpha^{\lambda(b+1)}, \ldots, \alpha^{\lambda(b+j-1)}, \ldots, \alpha^{\lambda(b+\delta-2)})$, что в точности совпадает в $\lambda+1$-ым столбцом проверочной матрицы вида (3). Таким образом, вычисляя частное от деления синдрома вектора y и синдрома представителя Г-орбиты (учитывая, что храним логарифмы многочленов – разность), можно узнать величину, на которую следует циклически сдвинуть представителя Г-орбиты, чтобы получить вектор-ошибку и исправить сообщение y. В случае, если частное от деления не совпадает ни с одним столбцом проверочной матрицы, значит ошибка в сообщении y не принадлежит Г-орбите. Таким образом седьмой пункт будет звучать следующим образом:

вычисляем разность s и представителя Г-орбиты, если результат не является столбцом проверочной матрицы – проверяем другую Г-орбиту. Если разность

совпадает со столбцом матрицы под номером n – применяем $n-1$ циклическую подстановку к представителю Г-орбиты.

В случае примитивного БЧХ-кода алгоритм будет выглядеть немного проще:

1. на вход получаем сигнал y;
2. вычисляем синдром $s = H \cdot y$. Если синдром равен нулю, y – кодовое слово, конец алгоритма;
3. если синдром не равен нулю, вычисляем норму синдрома и полиномиальную норму;
4. находим G-орбиту с такой же полиномиальной нормой;
5. для Г-орбиты применяем циклотомическую подстановку и вычисляем норму синдрома до тех пор, пока норма синдрома не будет совпадать с нормой вектора y;
6. вычисляем разность s и представителя Г-орбиты, которая совпадает со столбцом проверочной матрицы под номером n – применяем $n-1$ циклическую подстановку к представителю Г-орбиты;
7. конец алгоритма.

Приведем пример исправления ошибки при помощи квадратично-вычетного кода длины 73.

Пример 13. *Пусть нам пришло сообщение y с единицами на позициях 6, 7, 8, 13, 14, 17, 20, 22, 23, 25, 26, 30, 32, 33, 36, 38, 42, 44, 49, 50, 51, 52, 53, 54, 56, 57, 60, 63, 67, 71, 72. Выбираем соответствующие столбцы матрицы (5.3), переводим их к представлению с коэффициентами, суммируем между собой, результат переводим обратно в логарифмический вид. Получим синдром* $s = (317, -\infty, 23, 69)$ *(*$-\infty$ *говорит, что данная компонента синдрома равна нулю).*

Норма синдрома $N = (0, 0, 292, \infty, \infty, 292)$.

Полиномиальная норма $P = (1, 1, 32, 0, 0, 32)$ *– номера циклотомических классов по модулю 511, где первым считается нулевой класс.*

Такую же полиномиальную норму имеют три G-орбиты с единицами на позициях $e1 = (1, 2, 5, 10, 34, 47)$, $e_2 = (1, 3, 19, 22, 25, 49)$ *и* $e_3 = (1, 8, 23, 27, 33, 44)$.

Норма синдрома первой G-орбиты равняется $(0, 0, 146, \infty, \infty, 146)$, *что не совпадает с нашей нормой, поэтому применим к ней циклотомическую подстановку:* $\varphi(e_1) = (1, 3, 24, 38, 42, 54)$. *Норма синдрома* $(0, 0, 73, \infty, \infty, 73)$ *опять не совпадает с нашей. Применим подстановку еще раз:* $\varphi^2(e_1) = (1, 2, 49, 56, 58, 64)$. *Норма*

синдрома $(0, 0, 292, \infty, \infty, 292)$ *на этот раз совпадает, поэтому применим циклическую подстановку дважды к остальным векторам:* $e_1 = (1, 2, 49, 56, 58, 64)$, $e_2 = (1, 7, 13, 38, 42, 61)$, $e_3 = (1, 9, 21, 30, 43, 44)$.

Вычислим синдромы и разности синдромов:

Таблица 3: Синдромы и разности синдромов

Ошибка	Синдром	Разность с s
$e_1 = (1, 2, 49, 56, 58, 64)$	$S(e_1) = (237, -\infty, 475, 330)$	$(431, -, 452, 261)$
$e_2 = (1, 7, 13, 38, 42, 61)$	$S(e_2) = (401, -\infty, 366, 76)$	$(84, -, 343, 7)$
$e_3 = (1, 9, 21, 30, 43, 44)$	$S(e_3) = (190, -\infty, 204, 247)$	$(364, -, 181, 178)$

Учитывая, что $-$ *может быть любым числом как неопределенность, единственная разность, которая совпадает с каким либо столбцом проверочной матрицы – (84,-,343,7). Столбец (84,252,343,7) имеет порядковый номер 13, следовательно, нам необходимо сместить вектор* $e_2 = (1, 7, 13, 38, 42, 61)$ *на 12 разрядов влево.*

Получим вектор с единицами на позициях $e = (1, 26, 30, 49, 62, 68)$, *что и является ошибкой в сообщении* y. *Если прибавить данный вектор к* y *и посчитать синдром, то мы получим нулевой вектор, что говорит о том, что* $y + e$ *– кодовое слово.*

В таблице ниже предоставлена информация о результате реализации алгоритма исправления ошибок при помощи квадратично-вычетных кодов.

Таблица 4: Результате реализации алгоритма исправления ошибок

Длина кода	Минимальное расстояние	Количество ошибок которое код способен исправить	Занимаемая память G-орбитами	Скорость исправления ошибки, сек
31	7	3	3.54 КБ	0.0078
73	13	6	67.7 МБ	0.02-0.03

Заключение

В ходе выполнения дипломной работы была изучена теория помехоустойчивого кодирования: были изучены линейные коды, циклические коды, коды Хемминга, коды Боуза-Чоудхури-Хоквингема, квадратично-вычетные коды. Так же был изучен необходимый инструментарий для работы в данном направлении: алгебра, теория чисел и многочленов, теория групп и колец. Были изучены различные характеристики линейных кодов, построены проверочные и порождающие матрицы. Была исследована теория норм синдромов применительно к кодам Боуза-Чоудхури-Хоквингема.

В результате исследования групп автоморфизмов квадратично-вычетных кодов, алгоритма определения весового распределения КВ-кодов, был предложен весьма эффективный алгоритм для обнаружения слов минимального веса КВ-кодов. Так же был изучен и реализован алгоритм определения слов минимального веса при помощи теории норм синдромов.

Как результат дипломной работы, были построены Г- и G-орбиты. Одним из главных достижений дипломной работы считается создание и реализация алгоритма для построения Г-орбит. Алгоритм исправления ошибок при помощи норм синдромов был успешно применен для квадратично-вычетных кодов. Итоговые данные о результатах реализации данного алгоритма приведены в таблицах 2 и 4. Проект был успешно апробирован на XVI Белорусско-российской научно-технической конференции "Технические средства защиты информации" [14].

Список использованной литературы

[1] Шеннон К. Работы по теории информации и кибернетике. – М.: ИЛ, 1963. – 732 с.

[2] Липницкий В.А., Чесалин Н.В. Линейные коды и кодовые последовательности: учеб.-метод. пособие для студентов мех.-мат. фак. БГУ – Минск: БГУ, 2008. – 41 с.

[3] Мак-Вильямс Ф.Дж., Слоэн Н.Дж.А. Теория кодов, исправляющих ошибки. – М.: Связь, 1979. – 744 с.

[4] Кассами Т., Токура Н., Инвадари Ё., Инагаки Я. Теория кодирования. – М.: Мир, 1976. – 574 с.

[5] Лидл Р., Нидеррайтер Г. Конечные поля: В 2-х т. Пер. с англ. – М.: Мир, 1988. – 232 с.

[6] Липницкий В.А., Кушнеров А.В., Середа Е.В. Квадратично-вычетные коды как обобщенные коды Боуза-Чоудхури-Хоквингема. – Военная академия Республики Беларусь: Минск, Беларусь.

[7] Дьедонне Ж. Геометрия классических групп. – М.:Мир, 1974. – 204 с.

[8] Берлекэмп, Э., Алгебраическая теория кодирования. – М.: Мир. 1971.–480 с.

[9] J. Mykkeltveit, C. Lam and R.J. McEliece. On the weight enumeration of quadratic residue codes, JPL Technical Report 32-1526, Vol. XII, pp. 161-166, Jet Propulsion Laboratory, Pasadena, Calif., 1974.

[10] P.Gaborit, C.-S. Nedeloaia, A. Wassermann. On the weight enumerators of duadic and quadratic residue codes. – IEEE Trans. Inform. Theory, vol. 51, pp. 402-407, 2005.

[11] C.Tjhai, M. Tomlinson, M. Ambroze, M. Ahmed. On the Weight Distribution of the Extended Quadratic Residue Code of Prime 137. – University of Plymouth: Plymouth, PL4 8AA, United Kingdom, 2008.

[12] Липницкий В.А., Конопелько В.К. Норменное декодирование помехоустойчивых кодов и алгебраические уравнения. – Мн: БГУ, 2007. – 214 с.

[13] Липницкий В.А., Олексюк А.О. Теория норм синдромов и плюс-декодирование. – Военная академия Республики Беларусь: Минск, Беларусь.

[14] Реентович Е.В., Липницкий В.А. Квадратично-вычетные коды как коды Хемминга и обобщенные коды Боуза-Чоудхури-Хоквингема// XVI Белорусско-российская науч.-техн. конф. "Технические средства защиты информации" июнь 2018 г. – Тезисы докладов. – Минск : БГУИР, 2018. – с. 80-81.

Приложение

Список G-орбит для квадратично-вычетного кода длины 31 над полем $GF(2)$ с порождающим многочленом $q(x) = 1 + x + x^2 + x^6 + x^7 + x^{12} + x^{15}$ (в первом столбце указаны номера соответствующих циклотомических классов по модулю 31 над полем $GF(2)$, (0 – нет соответствующего класса)):

Полиномиальная норма	G-орбиты
(1, 1, 1)	(10000000000000000000000000000000)
(2, 6, 7)	(11000000000000000000000000000000)
(5, 4, 4)	(10010000000000000000000000000000)
(3, 3, 2)	(10000100000000000000000000000000)
(4, 5, 5)	(11100000000000000000000000000000)
(5, 5, 6)	(11010000000000000000000000000000)
(6, 3, 5)	(11001000000000000000000000000000)
(6, 6, 4)	(11000100000000000000000000000000)
(6, 2, 3)	(11000010000000000000000000000000)
(6, 3, 7)	(11000001000000000000000000000000)
(3, 2, 4)	(11000000100000000000000000000000)
(5, 4, 7)	(11000000010000000000000000000000)
(7, 6, 6)	(11000000001000000000000000000000)
(6, 5, 2)	(11000000000100000000000000000000)
(4, 1, 1)	(11000000000010000000000000000000)
(1, 7, 0)	(11000000000001000000000000000000)
(2, 7, 7)	(11000000000000100000000000000000)
(7, 6, 3)	(11000000000000010000000000000000)
(5, 2, 6)	(11000000000000001000000000000000)
(0, 0, 5)	(11000000000000000100000000000000)
(3, 3, 5)	(10100000000100000000000000000000)
(4, 2, 4)	(10100000000001000000000000000000)
(3, 4, 2)	(10100000000000010000000000000000)
(5, 3, 3)	(10100000000000000100000000000000)
(7, 7, 5)	(10010010000000000000000000000000)
(4, 4, 4)	(10010000100000000000000000000000)
(2, 7, 3)	(10010000010000000000000000000000)
(3, 6, 7)	(10010000000100000000000000000000)
(2, 4, 3)	(10010000000010000000000000000000)
(7, 5, 6)	(10010000000000000010000000000000)
(7, 5, 2)	(10010000000000000001000000000000)
(4, 7, 6)	(10000100001000000000000000000000)
(2, 2, 2)	(10000010000100000000000000000000)