Федеральное агентство по рыболовству
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Астраханский государственный технический университет»
Система менеджмента качества в области образования, воспитания, науки и инноваций сертифицирована DQS по международному стандарту ISO 9001:2015

Институт информационных технологий и коммуникаций
Кафедра «Информационная безопасность»

Теория информации

Учебно-методическое пособие

для обучающихся по специальности
09.03.01 Информатика и вычислительная техника
09.03.04 Программная инженерия

АСТРАХАНЬ – 2021

Авторы:
к.т.н., доцент кафедры «Информационная безопасность» И.В. Сибикина

Рецензент:
к.т.н., доцент кафедры «Информационная безопасность» Давидюк Н.В.

Учебно-методиченское пособие ***«Теория информации»*** утверждено на заседании кафедры «***Информационная безопасность***»
« 25 » 11 2021 г., протокол №. 10

Содержание.

Содержание. 1
Общие требования к выполнению лабораторных работ 4
Лабораторная №1. «Энтропия. Свойства энтропии». 6
Лабораторная работа №2. «Обработка алфавита введенного сообщения» 14
Лабораторная работа №3. «Оптимальное кодирование». 20
Лабораторная работа №4. «Код Хемминга». 30
Лабораторная работа №5. «Циклические коды». 40
Лабораторная работа №6. «Коды БЧХ». 49
Приложение. 56
Список литературы 61

Общие требования к выполнению лабораторных работ

Лабораторные работы выполняются на персональной ЭВМ с использованием языка программирования высокого уровня и заключаются в составлении программ, решающих определённый класс задач. Каждая программа должна обладать достаточным интерфейсом для удобства работы пользователя. В частности, это означает, что после запуска программы на каждом шаге работы пользователю должны быть даны чёткие указания или рекомендации по возможным вариантам его действий, а также необходимые комментарии промежуточных и окончательных результатов. При этом должна быть предусмотрена защита от неверного ввода с указанием на допущенную ошибку и приглашением повторить действие.

При отчёте о выполнении лабораторной работы студент должен:

показать в действии отлаженную программу, удовлетворяющую описанным выше требованиям; уверенно ориентироваться в алгоритме и самом тексте программы на

языке высокого уровня; знать необходимый теоретический материал.

При выполнении конкретных лабораторных работ преподаватель может уточнить или дополнить требования, приведённые выше, также систему оценивания и поощрений.

Лабораторная №1. «Энтропия. Свойства энтропии».

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Определение 1.1. Вероятностной схемой X называется

X	x_1	x_2	...	x_n
P	p_1	p_2	...	p_n

где $x_1, x_2, \ldots, x_n$- полная группа попарно несовместных событий, а $p_1, p_2, \ldots, p_n$ – соответствующие вероятности.

Определение 1.2. Количеством информации, содержащимся в сообщении x, называется $h(x) = -\log p(x)$. (Основание логарифма, если не оговорено противное, принимается равным 2.)

Определение 1.3. Энтропией вероятностной схемы X, называется $H(X) = -\sum_{i=1}^{n} p_i \cdot \log p_i$.

Значение функции f(t) = t·log t при t = 0 считаем равным нулю, доопределяя её в этой точке по непрерывности. Таким образом, эта

функция определена, по крайней мере, на отрезке [0;1].

Пусть имеются две схемы X и У

X	x1	x2	...	xn
P	p1	p2	...	pn

У	y1	y2	...	ym
P	q1	q2	...	qm

Определение 1.4. Энтропией произведения вероятностных схем X и У, называется

$$H(XY) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i y_j)\cdot \log\, p(x_i y_j)$$

Если схемы X и У независимы, то энтропия произведения вероятностных схем равна сумме энтропий каждой схемы: $H(XY) = H(X) + H(Y)$.

Определение 1.5. Условной энтропией вероятностной схемы У относительно схемы X называется:

$$H(Y \mid X) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i)\, p(y_j \mid x_i) \log\, p(y_j \mid x_i),$$

где $p(y_j \mid x_i)$ – условная вероятность события y_j при условии, что получено сообщение x_i.

Энтропия произведения и условная энтропия связаны между собой соотношениями:

$$H(XY) = H(X) + H(Y \mid X) = H(Y) + H(X \mid Y).$$

ПРИМЕР

Задание. Событие A в каждом из n повторных независимых испытаний происходит с вероятностью p. Найти энтропию числа появлений события A. Составить соответствующую вероятностную схему. Выяснить характер изменения энтропии в зависимости от изменения p на промежутке [0;1] при значении n = 1, построив график соответствующей функции H(p). Определить её наименьшее и наибольшее значение.

Рассмотрим энтропию числа появлений события A в серии из n испытаний.

Если n=1 и X- число появлений события A в серии из n испытаний, то

X	0	1
P	q	p

где q=1-p.

По определению 1.3, функция $H(p) = -p \cdot \log p - (1-p)\log(1-p)$. Построим график H(p) (рис.1):

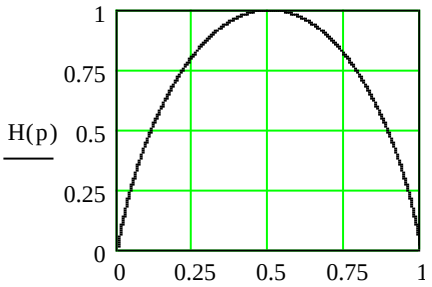


Рис.1 График функции H(p)

При p=0,5 функция H(p) достигает максимума H(0,5)=1, при p=0 или p=1 функция H(p) достигает минимума H(0)=H(1)=0. Функция возрастает на промежутке [0;0,5] и убывает на отрезке [0,5;1].

Таким образом, наименьшее значение, равное нулю, энтропия рассматриваемой вероятностной схемы принимает при p=0 и при p=1, то есть в тех случаях, когда исход опыта с вероятностной схемой X однозначно определён до его проведения. Наибольшее же значение, равное одному биту, энтропия данной схемы принимает только при p=0,5 , то есть в том случае, когда с равными вероятностями можно предполагать, что в результате испытания произойдёт или не произойдёт событие A, что соответствует наибольшей неопределённости исхода опыта с вероятностной схемой X до его проведения. При приближении p к 0,5 , то есть

с увеличением неопределённости, энтропия возрастает, а при приближении p к концам отрезка [0;1], то есть с уменьшением неопределённости, энтропия убывает. Следовательно, приведённые выше рассуждения подтверждают тезис о том, что энтропия является мерой неопределённости вероятностной схемы до проведения испытаний с ней.

ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Событие A в каждом из n повторных независимых испытаний происходит с вероятностью p. Найти энтропию числа появлений события A. Составить соответствующую вероятностную схему. Выяснить характер изменения энтропии в зависимости от изменения p на промежутке [0;1] при фиксированном значении n, построив график соответствующей функции H(p). Определить её наименьшее и наибольшее значение.(Значения параметра n задаются преподавателем.)

2. Событие A в каждом из независимых испытаний происходит с

вероятностью p. Найти энтропию числа испытаний до первого появления события A. Составить соответствующую вероятностную схему. Выяснить характер изменения энтропии в зависимости от изменения p на промежутке (0;1], построив график соответствующей функции H(p). Определить её наименьшее и наибольшее значение.

3. В партии из n изделий имеется k ($k \leq n$) стандартных. Наудачу отобраны m изделий ($m \leq n$). Найти энтропию числа стандартных изделий среди отобранных. Выяснить характер изменения энтропии в зависимости от изменения k на промежутке [0; n] при фиксированных значениях n и m, построив график соответствующей функции H(k). Для этого при каждом значении k составить необходимую вероятностную схему. Определить наименьшее и наибольшее значение H(k). (Значения параметров n и m задаются преподавателем.)

4. Интенсивность простейшего потока событий равна λ. Найти энтропию числа событий из этого потока,

появившихся за промежуток времени длительности t. Составить соответствующую вероятностную схему. Выяснить характер изменения энтропии в зависимости от изменения t на промежутке $[0;5\lambda]$ при фиксированном значении λ, построив график соответствующей функции $H(t)$. Определить её наименьшее и наибольшее значение.(Значения параметра λ задаются преподавателем.)

5. Интенсивность простейшего потока событий равна λ. Найти энтропию числа событий из этого потока, появившихся за промежуток времени длительности t. Составить соответствующую вероятностную схему. Выяснить характер изменения энтропии в зависимости от изменения λ на промежутке $[0;3t]$ при фиксированном значении t, построив график соответствующей функции $H(\lambda)$. Определить её наименьшее и наибольшее значение.(Значения параметра t задаются преподавателем.)

При выводе графика на экран должна быть тщательно прорисована система координат с обозначением и разметкой осей; показаны координаты экстремальных точек. Также аккуратно должны быть оформлены таблицы с вероятностными схемами.

ВОПРОСЫ

1. Количество информации в сообщении; основные свойства.
2. Количество информации в сообщении относительно другого сообщения; основные свойства.
3. Энтропия, условная энтропия; основные свойства.
4. Взаимная информация вероятностных схем; основные свойства.

Лабораторная работа №2. «Обработка алфавита введенного сообщения»

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.

Так как информацию можно рассматривать как неопределённость, снимаемую при получении сообщения, то можно дать следующее определение.

Определение 2.1. Пусть проводится k независимых испытаний с вероятностной схемой X. Тогда количеством информации, которое несёт в себе сообщение о результатах этой серии опытов, называется $I = k \cdot H(X)$.

В частном, но с практической точки зрения очень важном, случае, когда вероятностная схема X указывает вероятности появления символов алфавита от некоторого стохастического источника сообщений, причём буквы появляются независимо друг от друга, k интерпретируется как длина сообщения, полученного от данного источника, $H(X)$ – среднее количество информации, которое несёт в себе одна буква достаточно длинного сообщения, I - количество информации, которое несёт в себе сообщение из k символов.

Для случая равновероятных и взаимно независимых m символов $I = k \cdot \log m$.

Если схемы X и У статистически зависимы, то возможно измерение количества информации о системе X, которое дает наблюдение за системой У.

Определение 2.2. Количеством информации, которое несет схема У относительно схемы X называется:

$$I(Y, X) = H(Y) - H(Y \mid X)$$

Определение 2.3. Информационной избыточностью называется величина

$$D = 1 - \frac{H}{H_{\max}}$$

Частные виды избыточности.

1. Избыточность, обусловленная неравномерным распределением символов сообщения: $D_p = 1 - \frac{-\sum_i p_i \cdot \log p_i}{\log m}$

2. Избыточность, обусловленная статистической связью между символами сообщения:

$$D_s = 1 - \frac{-\sum_i \sum_j p(x_i) \cdot p(y_j \mid x_i) \cdot \log(y_j \mid x_i)}{\sum_i p_i \cdot \log p_i}$$

3. Полная информационная избыточность: $D = D_p + D_s - D_p D_s$.

ПРИМЕР

Задание. Произвести статистическую обработку данного сообщения, считая, что источник сообщений периодически, достаточно долго выдаёт следующую последовательность символов 12342334551233. Определить энтропию, приходящуюся в среднем на одну букву и на одно двухбуквенное сочетание, количество информации, которое несёт в себе сообщение о получении первой буквы относительно второй. Найти длину кода при равномерном кодировании и избыточность.

Пусть имеется сообщение:

1234233455123312342334551233123423345512331234233455123312342334551233... .

Составим схему появления однобуквенных сочетаний:

X	1	2	3	4	5	∑
n	2	3	5	2	2	14
w	$\frac{2}{14}$	$\frac{3}{14}$	$\frac{5}{14}$	$\frac{2}{14}$	$\frac{2}{14}$	1

Энтропия схемы X равна

$$H(X) = -\left[3 \cdot \frac{2}{14} \cdot \log \frac{2}{14} + \frac{5}{14} \cdot \log \frac{5}{14} + \frac{3}{14} \cdot \log \frac{3}{14}\right] = 2{,}21$$

Составим схему $\overline{XY}$ появления двухбуквенных сочетаний

ХУ	12	23	31	34	33	42	45	51	55	∑
n	2	3	1	2	2	1	1	1	1	14
w	$\frac{2}{14}$	$\frac{3}{14}$	$\frac{1}{14}$	$\frac{2}{14}$	$\frac{2}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	$\frac{1}{14}$	1

Энтропия, приходящаяся на одно двухбуквенное сочетание, составляет

$$H(\overline{XY}) = -\left[3 \cdot \frac{2}{14} \cdot \log \frac{2}{14} + 5 \cdot \frac{1}{14} \cdot \log \frac{1}{14} + \frac{3}{14} \cdot \log \frac{3}{14}\right] = 3{,}039$$

Количество информации, которое несет появление первой буквы о второй, найдем по определению 2.3:

$$H(Y \mid X) = -\sum_{i=1}^{n} \sum_{j=1}^{m} p(x_i)\, p(y_j \mid x_i) \log p(y_j \mid x_i) =$$

$$= H(XY) - H(X) = 0{,}829$$

$$I(Y, X) = 2{,}21 - 0{,}829 = 1{,}381$$

Найдем длину кода при равномерном кодировании однобуквенных сочетаний[1]:

m=5, $l = \lceil \log 5 \rceil = 3$

При этом возникает избыточность округления $\mathrm{D}_0 = 1 - \frac{\log 5}{3} = 0{,}226$

Подсчитаем информационную избыточность:

$$\mathrm{D}_\mathrm{p} = 1 - \frac{2{,}21}{\log 5} = 0{,}048\,,\ D_s = 1 - \frac{0{,}698}{2{,}21} = 0{,}684\,,$$

$$D = 0{,}048 + 0{,}684 - 0{,}048 \cdot 0{,}684 = 0{,}699$$

ПРАКТИЧЕСКАЯ ЧАСТЬ

Составить программу, позволяющую вводить сообщение произвольной длины из файла и с клавиатуры с последующей статистической обработкой введённого текста. Статистическая обработка текста включает в себя: выделение букв (включая пробелы и знаки препинания) алфавита данного сообщения; подсчёт и выведение на экран частоты и относительной частоты появления этих букв и указанных их сочетаний в порядке убывания вероятности. Определить энтропию,

[1] $\lceil x \rceil$ – округление в большую сторону.

приходящуюся в среднем на одну букву и на одно двухбуквенное сочетание, количество информации, которое несёт в себе сообщение о получении первой буквы относительно второй. Найти длину кода при равномерном кодировании и избыточность.

При выводе на экран в соответствующих таблицах должны присутствовать столбцы: номер по порядку; символ; частота; относительная частота.

ВОПРОСЫ

5. Вероятностная схема; произведение вероятностных схем.
6. Количество информации в сообщении; основные свойства.
7. Количество информации в сообщении относительно другого сообщения; основные свойства.
8. Энтропия, условная энтропия; основные свойства.
9. Взаимная информация вероятностных схем; основные свойства.

Лабораторная работа №3. «Оптимальное кодирование».

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.

Определение 3.1. m-значным кодированием сообщений α алфавита A, в кодовом алфавите B, называется отображение $F : S \to B^*$, где S – множество сообщений, B^* - множество всех слов в алфавите B, содержащем m символов. F (α) называется кодом сообщения α .

Определение 3.2. Кодирование называется алфавитным, если оно сохраняет произведения слов. Для алфавитного кодирования коды однобуквенных сообщений называются элементарными.

Определение 3.3. Соответствие между буквами алфавита A и их элементарными кодами при алфавитном кодировании называется схемой кодирования.

Определение 3.4. Схема кодирования называется префиксной, если никакой

элементарный код не является началом другого элементарного кода.

Определение 3.5. Средней длиной элементарного кода называется $\bar{l} = \sum_{i=1}^{n} p_i \cdot l_i$, где $l_i = l(\beta_i)$ - длина элементарного кода β_i.

Определение 3.6. Коэффициентом относительной эффективности кодирования называется величина

$$\eta = \frac{H(X)}{\bar{l}}$$

Определение 3.7. Оптимальным для данного стохастического источника сообщений называется такое алфавитное кодирование, для которого достигается минимальная средняя длина элементарного кода.

ОСНОВНЫЕ ТЕОРЕМЫ КОДИРОВАНИЯ

Теорема 3.1. Неравенство Крафта.

Неравенство $\sum_{j=1}^{M} D^{-k_j} \le 1$ является необходимым и достаточным условием

существования кодовых слов, соответствующих концевым узлам дерева с длинами, равными k_j, где D - основание системы счисления.

Теорема 3.2. Средняя длина кода $\bar{k}$ меньшая, чем $\frac{H}{\log D}$ является недостижимой ни при каком кодировании.

Теорема 3.3. Можно указать такой способ кодирования равно распределенных независимых сообщений, что средняя длина кода будет удовлетворять следующим требованиям: $\bar{k} < \frac{H}{\log D} + 1$

Теорема 4.3. Существуют такие способы кодирования для достаточно длинного сообщения x1, x2, … , что средняя длина кодового слова может быть сделана сколь угодно близкой к $\frac{H}{\log D}$

При построении оптимальных кодов можно использовать алгоритмы Шеннона-Фано или Хаффмана.

Алгоритм Шеннона-Фано.

1. Множество сообщений данной вероятностной схемы располагается в порядке убывания вероятностей.

2. Множество сообщений разбивается на части, приблизительно равные по суммарной вероятности. Первой части присваивается ноль, второй единица.

3. К каждой из частей применяются действия пункта 2.

Условием окончания работы алгоритма является наличие одного символа в каждой из подгрупп.

Алгоритм Хаффмана.

1. Последовательность сообщений данной вероятностной схемы располагается в порядке убывания вероятностей.

2. Последние два символа объединяются в один с вероятностью, равной сумме вероятностей объединенных символов.

3. С полученной последовательностью произвести действия пунктов 1 и 2, до образования последовательности из одного символа с суммарной вероятностью равной 1.

4. Строится кодовое дерево, в корне которого стоит символ с вероятностью 1.

ПРИМЕР

Задание. Произвести статистическую обработку данного сообщения, считая, что источник сообщений периодически, достаточно долго выдаёт следующую последовательность символов 12342334551233. Определить энтропию, приходящуюся в среднем на одну букву, длину кода при равномерном кодировании и избыточность. Построить схемы алфавитного кодирования методами Фано и Хаффмана. Найти среднюю длину элементарного кода, эффективность сжатия. Предусмотреть возможность кодирования короткого сообщения в данном алфавите, введённого с клавиатуры, по каждой из схем.

Статистическая обработка приведённого сообщения, была выполнена в предыдущем примере, где и была получена вероятностная схема

X	1	2	3	4	5	∑
n	2	3	5	2	2	14
w	$\frac{2}{14}$	$\frac{3}{14}$	$\frac{5}{14}$	$\frac{2}{14}$	$\frac{2}{14}$	1

Построим схему кодирования по алгоритму Шеннона-Фано.

символ	P				код
3	$\frac{5}{14}$	0	0		00
2	$\frac{3}{14}$		1		01
1	$\frac{2}{14}$	1	0		10
4	$\frac{2}{14}$		1	0	110
5	$\frac{2}{14}$			1	111

Кодовые комбинации полученные при кодировании методом Шенноно-Фано обладают свойством префикса, то есть ни одна более короткая кодовая комбинация не является началом более длинной.

Данное свойство является обязательным для оптимальных кодов и позволяет декодировать полученное сообщение, даже если нет разделителей между символами.

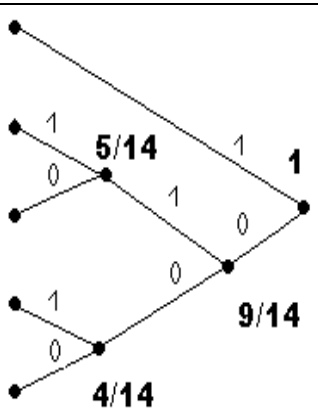
Средняя длина кодового слова равна

$$\bar{l} = \frac{5}{14} \cdot 2 + \frac{3}{14} \cdot 2 + \frac{2}{14} \cdot 3 \cdot 2 = 2.29$$

Коэффициент эффективности равен

$$\eta = \frac{2.21}{2.29} = 0.97$$

Построим схему кодирования по алгоритму Хаффмена.

символ	P		код
3	$\frac{5}{14}$		1
2	$\frac{3}{14}$		011
1	$\frac{2}{14}$		010
4	$\frac{2}{14}$		001
5	$\frac{2}{14}$		000

Средняя длина кодового слова равна

$$\bar{l} = \frac{5}{14} + \frac{3}{14} \cdot 3 + \frac{2}{14} \cdot 3 \cdot 3 = 2.29$$

Коэффициент эффективности равен

$$\eta = \frac{2.21}{2.29} = 0.97$$

БЛОЧНОЕ КОДИРОВАНИЕ

Пусть имеются две буквы алфавита A и B. Как возможно закодировать данные буквы, видимо только по одному символу.

A	0.9	0
B	0.1	1

Средняя длина будет равна 1 биту
$\bar{k} = 1 \cdot 0.9 + 1 \cdot 0.1 = 1$ бит/буква

А энтропия равна
$H = -0,9 \cdot \log 0,9 - 0,1 \cdot \log 0,1 = 0,47$. То есть, избыточность составляет 53%. Как же быть? Попробуем закодировать двухбуквенные сочетания. В этом случае уже можно воспользоваться эффективным кодированием.

AA	0.81	0	------	------	0
AB	0.09		0	------	10
BA	0.09	1	1	0	110
BB	0.01			1	111

Тогда средняя длина на блок из двух букв будет $\bar{k}_{2б} = 1,29$. А на одну букву $\bar{k}$ =0.645 бит/буква. Избыточность в этом случае будет уже составлять примерно 17%. Если мы возьмем сочетания из трех букв, то получим еще лучший результат и т.д. Увеличивая длину блоков можно как угодно близко приблизиться к оптимальному значению $H / \log 2$.

Блочное кодирование удобно применять и для устранения избыточности при кодировании десятичных цифр. При передаче десятичных цифр двоичным кодом

максимально загруженными бывают только те символы вторичного алфавита, которые передают значения, являющиеся целочисленными степенями двойки. Это 4, 8, 16, … В других случаях тремя разрядами можно передать и 5 и 8. Так для передачи цифры 5 необходимо $k = \log 5 / \log 2 = 2{,}32 бит$. Однако эту цифру необходимо округлить до ближайшего целого числа 3. Избыточность от округления будет составлять

$$L = \frac{k_ч - k}{k_w} = 1 - \frac{k}{k_w} = 1 - \frac{2{,}32}{3} = 0{,}227$$

Избыточность от не равновероятного появления символов и избыточность от округления можно устранить за счет кодирования блоками.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Составить программу, позволяющую вводить сообщение произвольной длины из файла и с клавиатуры с последующей статистической обработкой введённого текста. Определить энтропию, приходящуюся в среднем на одну букву, длину кода при

равномерном кодировании и избыточность. Построить схемы алфавитного кодирования методами Фано и Хаффмана. Найти среднюю длину элементарного кода, эффективность сжатия. Предусмотреть возможность кодирования короткого сообщения в данном алфавите, введённого с клавиатуры, по каждой из схем.

При выводе на экран в соответствующих таблицах должны присутствовать столбцы: номер по порядку; символ; относительная частота; элементарный код.

ВОПРОСЫ

10. Кодирование. Алфавитное кодирование. Основные понятия.
11. Префиксные схемы алфавитного кодирования.
12. Неравенство Крафта-Макмиллана.
13. Стохастические источники сообщений. Основные понятия. Теоремы Шеннона.
14. Экономное кодирование. Определение, основные свойства.
15. Методы кодирования Фано и Хаффмана.

Лабораторная работа №4. «Код Хемминга».

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.

Корректирующими называются коды позволяющие обнаруживать и исправлять ошибки. Идею представления корректирующих кодов можно представить с помощью N-мерного пространства. Возьмем трехмерный куб (рис, 2.) длина ребер, в котором равна одной единице. Вершины такого куба отображают двоичные коды. Минимальное расстояние между вершинами определяется минимальным количеством ребер, находящихся между вершинами. Это расстояние называется кодовым (или хэмминговым) и обозначается буквой d.

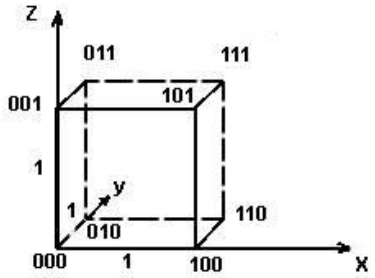


Рис.2. Представление двоичных кодов с помощью куба

Иначе, кодовое расстояние – это то минимальное число элементов, в которых одна кодовая комбинация отличается от другой. Для определения кодового расстояния достаточно сравнить две кодовые комбинации по модулю 2. Так, сложив две комбинации

10110101101
11001010101
01111111000

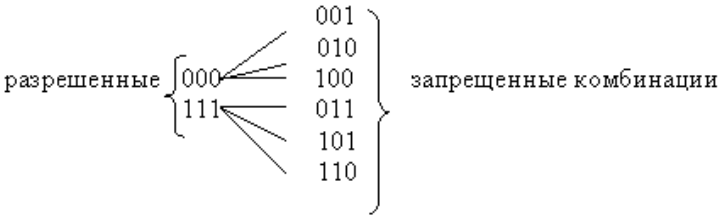
определим, что расстояние между ними d=7.

Для кода с N=3 восемь кодовых комбинаций размещаются на вершинах трехмерного куба. Такой код имеет кодовое расстояние d=1, и для передачи используются все восемь кодовых комбинаций 000,001,..,111. Такой код является не помехоустойчивым, он не в состоянии обнаружить ошибку.

Если выберем комбинации с кодовым расстоянием d=2, например, 000,110,101,011, то такой код позволит обнаруживать однократные ошибки. Назовем эти комбинации

разрешенными, предназначенными для передачи информации. Все остальные 001,010,100,111 - запрещенные.

Любая одиночная ошибка приводит к тому, что разрешенная комбинация переходит в ближайшую, запрещенную комбинацию (см. рис.5.3). Получив запрещенную комбинацию, мы обнаружим ошибку. Выберем далее вершины с кодовым расстоянием d=3



Такой код может исправить одну одиночную ошибку или обнаружить две ошибки. Таким образом, увеличивая кодовое расстояние можно увеличить помехоустойчивость кода. В общем случае кодовое расстояние определяется по формуле

$$d=t+l+1$$

где t - число исправляемых ошибок , l - число обнаруживаемых ошибок. Обычно $l>t$.

Большинство корректирующих кодов являются линейными кодами. Линейные коды - это такие коды, у которых контрольные символы образуются путем линейной комбинации информационных символов. Кроме того, корректирующие коды являются групповыми кодами. Групповые коды (G_n) - это такие коды, которые имеют одну основную операцию. При этом, должно соблюдаться условие замкнутости (то есть, при сложении двух элементов группы получается элемент принадлежащий этой же группе). Число разрядов в группе не должно увеличиваться. Этому условию удовлетворяет операция поразрядного сложения по модулю 2. В группе, кроме того, должен быть нулевой элемент.

Ниже приведены кодовые комбинации, являющиеся группой или нет.

1) 1101 1110 0111 1011 – не группа, так как нет нулевого элемента

2) 0000 1101 1110 0111 – не группа, так как не соблюдается условие замкнутости (1101+1110=0011)

3) 000 001 010 011 100 101 110 111 - группа

4) 000 001 010 111 - подгруппа

Большинство корректирующих кодов образуются путем добавления к исходной k – комбинации r – контрольных символов. В итоге в линию передаются n=k+r символов. При этом корректирующие коды называются (n,k) кодами. Для построения кода способного обнаруживать и исправлять одиночную ошибку необходимое число контрольных разрядов будет составлять

$$n-k \geq \log(n+1).$$

Это равносильно известной задаче о минимуме числа контрольных вопросов, на которые могут быть даны ответы вида “да” или “нет”, для однозначного определения одного из элементов конечного множества.

Если необходимо исправить две ошибки, то число различных исходов будет составлять C_n^2. Тогда $n-k \geq \log\left(1+C_n^1+C_n^2\right)$, в этом случае обнаруживаются однократные и двукратные

ошибки. В общем случае, число контрольных символов должно быть не меньше

$$n-k \geq \log\left(1+C_n^1+C_n^2+\ldots+C_n^t\right)=\log\sum_{i=0}^{t}C_n^i$$

Эта формула называется неравенством Хэмминга, или нижней границей Хэмминга для числа контрольных символов.

КОД ХЭММИНГА

Для исправления одной ошибки кодовое расстояние должно быть не менее 3 ($d_0 = 2s+1 \geq 3$).

Для того чтобы в принятом сообщении можно было исправлять ошибки, кодовая комбинация должна обладать некоторой избыточностью, которая достигается за счет добавления контрольных разрядов. Число корректирующих разрядов должно удовлетворять следующим условиям.

Пусть r– число корректирующих символов, k– количество информационных разрядов, n– длина кода, тогда

$\log(n+1)+1 > r \geq \log(n+1)$.

Код Хемминга является типичным примером систематического кода и может строиться на основе производящей матрицы.

Порождающая матрица имеет k строк и n столбцов.

Порождающая матрица G может быть представлена двумя матрицами, единичной и добавочной. При выборе добавочной матрицы учитывают, что вес (весом двоичного вектора называется величина расстояния Хемминга от него до нулевого вектора) каждой строки не должен быть менее $d_0 - 1$.

Кодирование реализуется при помощи умножения информационной комбинации α на порождающую матрицу

$$\beta = \alpha \cdot G$$

Проверочная матрица H при двоичном кодировании представляет собой транспонированную добавочную матрицу, дополненную единичной. Проверочная матрица имеет r строк и n столбцов. Причем столбцы представляют собой значения синдрома для разряда, соответствующего номеру этого столбца.

Для определения синдрома необходимо умножить кодовую комбинацию на транспонированную проверочную матрицу

$$S = \overline{\beta} \cdot H^T$$

ПРИМЕР

Задание. Методом Хемминга закодировать комбинацию α=1101, построив порождающую проверочную матрицы. Внести ошибку в один из разрядов кодового вектора; найти синдром; найти и исправить ошибку.

Нетрудно видеть, что число информационных разрядов k=4, определим r, n.

Для расчета r можем использовать эмпирическую формулу $r = \lceil \log((k+1) + \lceil \log(k+1) \rceil) \rceil$. Получим r=3, n=7.

Имеем (7,4)– кодирование. Порождающая матрица G имеет размерность 4×7, а проверочная – 3×7.

Построим проверочную матрицу H, так чтобы ее столбцы были различны и не содержали нулевую комбинацию:

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}, \; d_0 \geq 3$$

Строим порождающую матрицу G:

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

Кодовая комбинация β имеет вид $\beta=\alpha G=1101010$,

Внесем ошибку в третий разряд $\overline{\beta}=1111010$, вычислим синдром $S=\overline{\beta}\cdot H^T=101$, что соответствует ошибке в третьем разряде. Исправленная кодовая комбинация $\beta_{исп}=1101010$.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Методом Хемминга закодировать информационные комбинации предложенные преподавателем, построив порождающую проверочную матрицы. В программном продукте предусмотреть поле для ввода комбинаций различной длины, Внести ошибку в один из разрядов кодового вектора; найти синдром; найти и исправить ошибку.

ВОПРОСЫ

1. Линейное кодирование. Основные понятия.
2. Порождающая и проверочная матрицы; синдром.

3. Помехоустойчивое кодирование. Основные понятия. Расстояние Хемминга; кодовое расстояние.

4. Метод кодирования Хемминга.

Лабораторная работа №5. «Циклические коды».

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Определение 5.1. Блочное (n, k)-кодирование называется циклическим, если при любой циклической перестановке символов кодовой комбинации получается также кодовая комбинация.

Теорема 5.1. Блочное (n, k)-кодирование является циклическим тогда и только тогда, когда оно порождено многочленом степени $r = n - k$, являющимся делителем двучлена $x^n + 1$.

Определение 5.2. Частное от деления двучлена $x^n + 1$ на порождающий многочлен называется проверочным многочленом.

Таким образом, для того, чтобы задать циклическое кодирование необходимо и достаточно определить соответствующий порождающий многочлен. Неприводимые делители двучленов $x^n + 1$ табулированы (см., например, таблицу 1), а прочие многочлены, порождающие циклические коды

могут быть представлены как наименьшие общие кратные неприводимых.

Теорема 5.2. Для того, чтобы циклическое кодирование позволяло исправлять не менее одной ошибки необходимо и достаточно, чтобы остатки от деления одночленов x^i (i = 0, 1, …,n – 1) на соответствующий порождающий многочлен были различны.

Построение и декодирование циклических кодов, исправляющих одиночную ошибку, осуществляется следующим образом.

1) Производится расчет количества контрольных символов. Если задано число информационных разрядов, то можем воспользоваться эмпирической формулой $r = \lceil \log((k+1) + \lceil \log(k+1) \rceil) \rceil$

2) Выбор образующего многочлена производится по таблице неприводимых многочленов (Таблица 1).Образующий многочлен следует выбирать как можно более коротким, но степень его должна быть не менее числа контрольных разрядов, а число ненулевых членов– не меньше кодового расстояния.

3) Производится умножение многочлена, соответствующей информационной комбинации на одночлен той же степени, что и образующий многочлен.

4) Значения корректирующих разрядов находятся в результате деления многочлена, полученного в пункте 3) на образующий многочлен. Остаток от деления складывается по модулю 2 с многочленом, полученным в пункте 3).

Обнаружение и исправление ошибки также производится с помощью остатка от деления полученной комбинации на образующий многочлен.

Если принятая комбинация делится на образующий многочлен без остатка, то принят правильный код.

Если остаток не равен нулю, то в коде присутствует ошибка. Для исправления ошибки необходимо выполнить ряд действий.

1) Подсчитать вес (весом двоичного вектора называется величина расстояния Хемминга от него до нулевого вектора) остатка. Если он не больше корректирующей способности кода, то принятую комбинацию

складывают с по модулю 2 с полученным остатком. Результат дает исправленную комбинацию.

2) Если вес остатка больше корректирующей способности кода, то необходимо циклически сдвинуть кодовую комбинацию на один разряд влево и результат поделить на образующий многочлен. Если вес остатка не больше корректирующей способности кода, то делимое складывают с остатком, а затем производят циклический сдвиг вправо на один разряд. Полученная комбинация является исправленной.

3) Если вес остатка больше корректирующей способности кода, то необходимо циклически сдвигать кодовую комбинацию влево, пока остаток не станет меньше корректирующей способности кода. В этом случае, для восстановления исправленной комбинации, результат сложения последнего делимого с его остатком сдвигают на такое количество разрядов вправо, сколько было совершено сдвигов влево.

ПРИМЕР

Задание. Закодировать комбинацию α=1101, построив порождающий и проверочный многочлен. Внести ошибку в один из разрядов кодового многочлена; проверить полученное сообщение; найти и исправить ошибку.

Очевидно, что число информационных разрядов k=4, определим кодовое расстояние r и порождающий многочлен.

Так как d_0=3, то для расчета r можем использовать формулу $r = \lceil \log((k+1) + \lceil \log(k+1) \rceil) \rceil$, r=3. По таблице 1 найдем образующий многочлен – $x^3 + x + 1$ или 1011 (в дальнейшем все многочлены мы записываем, как последовательность коэффициентов в порядке убывания степеней).

Умножив его на одночлен 1000, получим 1101000.

Найдем остаток от деления полученной комбинации на образующий многочлен:

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

Сложим комбинацию 1101000 с остатком 001:

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

Кодовая комбинация β имеет вид β =1101001,

Внесем ошибку во второй разряд $\overline{\beta}$=1001001. Для обнаружения и исправления ошибки произведем деление:

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

Так как вес остатка больше одного, то производим циклический сдвиг на один разряд влево с последующим делением на образующий многочлен:

```
0  0  1  0  0  1  1 | 1  0  1  1
   1  0  1  1       | 1
   ----------
      1  0  1
```

Так как вес остатка больше одного, то производим циклический сдвиг на один разряд влево с последующим делением на образующий многочлен:

```
0  1  0  0  1  1  0 | 1  0  1  1
   1  0  1  1       | 1  0  1
   ----------
      1  0  1  0
      1  0  1  1
      ----------
               1
```

Так как вес остатка не больше корректирующей способности кода, то производим суммирование по модулю 2, и для получения исправленной комбинации производим циклический сдвиг на 2 разряда вправо:

```
0  1  0  0  1  1  0
               1
-----------------
0  1  0  0  1  1  1
```

Исправленная комбинация $\beta_{исп}$ =1101001.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Закодировать указанные информационные комбинации, построив порождающий и проверочный многочлен. Внести ошибку в один из разрядов кодового многочлена; проверить полученное сообщение; найти и исправить ошибку.

Вариант	Информационные комбинации		
1	01111	101	10000001
2	111	1111000	11110
3	0011	10110	0000100
4	1111	0000110	10101
5	010111	11100011	001
6	011100	1010	1010101010
7	1110	010	010001000
8	1001	110	111001010
9	011	111111110	0100011
Вариант	Информационные комбинации		
10	001	10001010	1111
11	111100001	1010	101
12	11000000	111	00001
13	110	001001	11110111
14	10011	011	011111111
15	001001001	11111	100

ВОПРОСЫ

5. Линейное кодирование. Основные понятия.

6. Порождающая и проверочная матрицы; синдром.

7. Помехоустойчивое кодирование. Основные понятия. Расстояние Хемминга; кодовое расстояние.

8. Коды, порождённые многочленами. Основные понятия.

9. Циклические коды. Основные понятия и свойства.

Лабораторная работа №6. «Коды БЧХ».

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.

Коды Боуза-Чоудхури-Хоквингема (БЧХ) относятся к циклическим кодам с $d_0 \geq 5$, то есть позволяющим исправлять не менее двух ошибок. Методика их построения имеет отличительные особенности в выборе образующего многочлена. Выбор образующего многочлена в основном зависит от длины кода n и числа исправляемых ошибок s. Соотношения длины кода n, числа информационных символов k и количества корректирующих разрядов r приведены в таблице 2. Для, так называемых, примитивных двоичных кодов БЧХ необходимо, чтобы n удовлетворяло условию: $n = 2^h - 1$ для некоторого натурального числа h.

Если известна длина кода n, удовлетворяющая указанному выше условию, то $h = \log(n+1)$. Тогда, чтобы построить многочлен g(x), порождающий БЧХ код с исправлением s ошибок, необходимо, выбрав произвольный примитивный многочлен P(z) степени h, построить поле Галуа GF(2^h)

(очевидно, при таком построении z будет примитивным элементом поля $GF(2^h)$), найти минимальные многочлены $P_i(x)$, (i = 1, 2, …, 2s) для всех нечётных степеней z^i выбранного примитивного элемента и положить $g(x)=НОК(P_1(x),P_3(x), …, P_{2s-1}(x))$.

На практике, поскольку минимальные многочлены табулированы, можно обойтись без непосредственного построения поля Галуа, как это, например, показано ниже.

Число контрольных символов равно степени образующего многочлена g(x). Построение производится при помощи минимальных многочленов (таблица 3).

h указывает на колонку в таблице минимальных многочленов (таблица 3), из которой производится выбор многочлена P(x).

Так как для построения g(x) используются только нечетные многочлены, то их количество равно числу исправляемых ошибок.

Обнаружение и исправление ошибок производится по той же методике, что и для циклических кодов.

ПРИМЕР

Задание. Закодировать информационную комбинацию α=10011, , построив порождающий многочлен для кода, исправляющего s=3 ошибок при минимальной длине *n* кодового слова. Внести $m = 2$ ошибки в кодовую комбинацию; проверить полученное сообщение; найти и исправить ошибки.

Очевидно, что число информационных разрядов k=5.

По таблице 2 находим наименьшее значение n=15 для k=5, s=3.

Тогда h = log16 = 4, следовательно, старшая степень минимального многочлена равна 4.

i=2s-1=5, следовательно из четвертой колонки таблицы 3 выбираем $P_1(x),P_3(x),P_5(x)$.

$g(x)$ = НОК($P_1(x),P_3(x),P_5(x)$) = 10011·11111·111 = 10100110111 – образующий многочлен.

r=10, следовательно умножаем информационную комбинацию на x^{10}, а затем делим на образующий многочлен,

```
1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 | 1 0 1 0 0 1 1 0 1 1 1
1 0 1 0 0 1 1 0 1 1 1         |   1 0 1 1
      1 1 1 1 1 0 1 1 1 0 0
      1 0 1 0 0 1 1 0 1 1 1
        1 0 1 1 1 0 1 0 1 1 0
        1 0 1 0 0 1 1 0 1 1 1
            0 1 1 1 0 0 0 0 1 0
```

β =100110111000010 – кодовая комбинация. Внесем ошибку во второй и третий разряды.

$\overline{\beta}$ =111110111000010

```
1 1 1 1 1 0 1 1 1 0 0 0 0 1 0 | 1 0 1 0 0 1 1 0 1 1 1
1 0 1 0 0 1 1 0 1 1 1         |   1 1 0 0 1
  1 0 1 1 1 0 1 0 1 1 0
  1 0 1 0 0 1 1 0 1 1 1
        1 1 1 0 0 0 0 1 0 1 0
        1 0 1 0 0 1 1 0 1 1 1
            1 0 0 0 1 1 1 1 0 1
```

Так как вес остатка больше 3, то произведем циклический сдвиг на один разряд влево, и снова найдем остаток от деления на образующий многочлен:

```
1 1 1 1 0 1 1 1 0 0 0 0 1 0 1 | 1 0 1 0 0 1 1 0 1 1 1
1 0 1 0 0 1 1 0 1 1 1         |   1 1 0 0 0
  1 0 1 0 0 0 1 1 1 1 0
  1 0 1 0 0 1 1 0 1 1 1
          0 1 0 1 0 0 1 1 0 1
```

Вес остатка больше 3, следовательно сдвигаем на второй разряд влево с последующим делением на образующий многочлен:

```
1 1 1 0 1 1 1 0 0 0 0 1 0 1 1 | 1 0 1 0 0 1 1 0 1 1 1
1 0 1 0 0 1 1 0 1 1 1         |     1 1 0 1 1
-----------------------
  1 0 0 1 0 0 0 1 1 1 1
  1 0 1 0 0 1 1 0 1 1 1
  -------------------
      1 1 0 1 1 1 0 0 0 0 1
      1 0 1 0 0 1 1 0 1 1 1
      -------------------
        1 1 1 1 0 1 0 1 1 0 1
        1 0 1 0 0 1 1 0 1 1 1
        -------------------
          1 0 1 0 0 1 1 0 1 0
```

Вес остатка снова больше 3, следовательно сдвигаем еще на разряд влево с последующим делением на образующий многочлен:

```
1 1 0 1 1 1 0 0 0 0 1 0 1 1 1 | 1 0 1 0 0 1 1 0 1 1 1
1 0 1 0 0 1 1 0 1 1 1         |       1 1 1
-----------------------
  1 1 1 1 0 1 0 1 1 0 0
  1 0 1 0 0 1 1 0 1 1 1
  -------------------
    1 0 1 0 0 1 1 0 1 1 1
    1 0 1 0 0 1 1 0 1 1 1
    -------------------
                        1 1
```

Так как вес остатка не больше корректирующей способности кода, то производим суммирование по модулю 2, и для

получения исправленной комбинации производим циклический сдвиг на 3 разряда вправо:

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

Исправленная комбинация
$\beta_{исп}$ =100110111000010.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Закодировать информационные комбинации, предложенные преподавателем, построив порождающий многочлен для кода, исправляющего s ошибок при минимальной длине n кодового слова. В программном продукте предусмотреть ввод кодируемых комбинаций различной длины и определение максимального количества ошибок. Внести m ошибок ($m \leq s$) в кодовую комбинацию; проверить полученное сообщение; найти и исправить ошибки.

ВОПРОСЫ

10. Линейное кодирование. Основные понятия.

11. Порождающая и проверочная матрицы; синдром.

12. Помехоустойчивое кодирование. Основные понятия. Расстояние Хемминга; кодовое расстояние.

13. Коды, порождённые многочленами. Основные понятия.

14. Циклические коды. Основные понятия и свойства.

15. Неприводимые, примитивные и минимальные многочлены.

16. Коды Боуза-Чоудхури-Хоквингема.

Приложение.

Таблица 1.

Фрагменты таблицы образующих многочленов.

код[2]	код	код
11	111001	1101101
101	111011	1101111
111	111101	1110001
1001	111111	1110011
1011	1000001	1110101
1101	1000011	1110111
1111	1000101	1111001
10001	1000111	1111011
10011	1001001	1111101
10101	1001011	1111111
10111	1001101	10000001
11001	1001111	11100001
11011	1010001	100000001
11101	1010011	100000011
11111	1010101	1000000001
100001	1010111	1100000001
100011	1011001	10000000001
100101	1011011	100000000001
100111	1011101	100000000011

[2] Под заголовком код понимается вектор коэффициентов многочлена, например, коду 11 соответствует многочлен x+1, коду 1001 сопоставляется многочлен x^3+1.

101001	1011111	100000000101
101011	1100001	1000000000001
101101	1100011	10000000000001
101111	1100101	100000000000011
110001	1100111	100000000000101
110101	1101001	100000000000111
110111	1101011	100000000001001

Таблица 2.

Соотношение корректирующих и информационных разрядов для БЧХ кодов[3].

n	K	r	s	n	k	r	s
7	4	3	1	127	106	21	3
15	11	4	1	127	99	28	4
15	7	8	2	127	92	35	5
15	5	10	3	127	85	42	6
31	26	5	1	127	78	49	7
31	21	10	2	127	71	56	9
31	16	15	3	127	64	63	10

[3] В таблице приняты следующие обозначения: n – длина кода, k – число информационных символов, r – число корректирующих символов, s – число исправляемых ошибок.

31	11	20	5	127	57	70	11
31	6	25	7	127	50	77	13
63	57	6	1	127	43	84	14
63	51	13	2	127	36	91	15
63	45	18	3	127	29	98	21
63	39	24	4	127	22	105	23
63	36	27	5	127	15	112	27
63	30	33	6	127	8	119	31
63	24	39	7	255	247	8	1
63	18	35	10	255	239	16	2
63	16	37	11	255	231	24	3
63	10	53	13	255	223	32	4
63	7	56	15	255	215	40	5
127	120	7	1	255	207	48	6
127	113	14	2	255	199	56	7

Таблица 3.

Минимальные неприводимые многочлены в поле Галуа GF(2).

степень	2	3	4	5	6
1	111	1011	10011	100101	1000011
3		1101	11111	111101	1010111
5			111	110111	1100111
7			11001	101111	1001001
9				110111	1101
11				111011	1101101

Таблица 3 (продолжение)

степень	7	8	9	10
1	10001001	100011101	1000010001	10000001001
3	10001111	101110111	1001011001	10000001111
5	10011101	111110011	1100110001	10100001101
7	11110111	101101001	1010011001	11111111001
9	10111111	110111101	1100010011	10010101111
11	11010101	111100111	1000101101	10000110101
13	10000011	100101011	1001110111	10001101111
15		111010111	1101100001	10110101011
17		010011	1011011011	11101001101
19	11001011	101100101	1110000101	10111111011
21	11100101	110001011	1000010111	11111101011
23		101100011	1111101001	10000011011
25		100011011	1111100011	10100100011
27		100111111	1110001111	11101111011
29			101101011	10100110001
31				11000100001
33				111101
35			1100000001	11000010011
37		101011111	1001101111	11101100011
39			1111001101	10001000111
41			1101110011	10111100101
43		111000011	1111001011	10100011001
45		100111001	1001111101	11000110001
47				11001111111
49				11101010101
51		011111	1111010101	10101100111
53			1010010101	10110001111
55			1010111101	11100101011

57				11001010001
59				11100111001
67				10111000001
69				11011010011

Таблица 3(продолжение).

степень	8	9	10
71		1011	11101000111
73		1111111011	10001011111
75		1101001001	10100001011
83	111	1100010101	11110010011
85		1010110111	10111000111
87			10011001001
89			10011010111
91			11010110101
93			11111111111
99			110111

Список литературы

1. Белов, В.М. Теория информации. Курс лекций: Учебное пособие / В.М. Белов, С.Н. Новиков, О.И. Солонская. - М.: ГЛТ, 2012. - 143 с.
2. Белов, В.М. Теория информации. Курс лекций: Учебное пособие для вузов. / В.М. Белов, С.Н. Новиков, О.И. Солонская. - М.: РиС, 2016. - 143 с
3. Кузьмин, И. В. Основы теории информации и кодирования / И.В. Кузьмин, В.А. Кедрус. - М.: Вища школа, 2016. - 280 с