Министерство образования и науки Украины
Севастопольский национальный
технический университет

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к практическим занятиям по дисциплине
«Теория информации и кодирование».
Часть 2 «Эффективные коды»
для студентов направления подготовки
0915 – "Компьютерная инженерия"
дневной и заочной форм обучения

Севастополь
2010

УДК 621.391.23

Методические указания к практическим занятиям по дисциплине «Теория информации и кодирование». Часть 2 «Эффективные коды» для студентов направления подготовки 0915 – "Компьютерная инженерия" дневной и заочной форм обучения/Сост. Щепин Ю.Н. – Севастополь, Изд-во Сев НТУ, 2010. – 28 с.

Целью методических указаний является оказание помощи студентам в освоении методов построения оптимальных кодов.

Методические указания рассмотрены и утверждены на заседании кафедры кибернетики и вычислительной техники, протокол № 10 от 17.06.10 г.

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Рецензент Апраксин Ю.К., профессор кафедры кибернетики и вычислительной техники, доктор технических наук.

СОДЕРЖАНИЕ

Цель работы

Цель практических занятий: ознакомление с принципами кодирования информации и приобретение навыков построения эффективных кодов.

Введение

Методические указания адресованы студентам очной и заочной форм обучения направления подготовки 0915 – «Компьютерная инженерия», изучающим дисциплину «Теория информации и кодирование».

Необходимыми условиями успешного изучения данной дисциплины являются хорошие знания по курсам «Высшая математика», «Дискретная математика», «Прикладная теория цифровых автоматов» и «Теория вероятностей и математическая статистика».

В соответствии с рабочей программой дисциплины «Теория информации и кодирование» студенты при изучении раздела «Эффективное кодирование» должны приобрести практические навыки применения основных положений этого раздела: научиться строить оптимальные коды, давать оценку эффективности построенных кодов.

Каждый раздел настоящих методических указаний содержит краткое введение, которое не исчерпывает необходимый минимум теоретических положений для успешного решения списка задач. Прежде, чем приступать к решению задач, необходимо изучить соответствующие теоретические положения, пользуясь конспектом лекций и рекомендованной литературой, а также найти ответы на все предлагаемые контрольные вопросы.

1. Основные понятия и определения

Преобразование сообщения в сигнал, удобный для передачи по данному каналу связи, называют кодированием в широком смысле. Операцию восстановления сообщения по принятому сигналу называют декодированием. Для обеспечения простоты и надежности распознавания образцовых сигналов их число целесообразно сократить до минимума. В этой связи используют операцию представления исходных знаков (букв первичного алфавита) во вторичном алфавите с меньшим числом знаков, называемых символами. При обозначении этой операции используется тот же термин «кодирование», понимаемый теперь в более узком смысле. Поскольку мощность вторичного алфавита меньше мощности первичного алфавита, то каждой букве соответствует некоторая последовательность символов, которую назовём кодовой комбинацией или кодовым словом, или просто кодом. Под кодом понимают правила (алгоритм), сопоставляющие каждой букве кодовую комбинацию символов. Число символов в кодовой комбинации называют её значностью или её длиной. Код, в котором каждое кодовое слово имеет одну и ту же длину, называют кодом с фиксированной длинной или равномерным кодом. Коды, в которых различные кодовые слова могут иметь различную длину, называются неравномерными.

В процессе кодирования могут преследоваться следующие цели:

1) Простота реализации информационных устройств, минимальное время реализации, повышение эффективности системы связи, при этом наличие помех в системе связи не учитывается. Кодирование, удовлетворяющее требованиям повышения эффективности системы связи, получило название эффективного или оптимального кодирования.

2) Обеспечение заданной помехоустойчивости. Такое кодирование получило название помехоустойчивого кодирования. Код, который может обнаруживать и (или) исправлять ошибки, называется корректирующим или помехоустойчивым.

Перечисленные цели противоречивы, так как увеличение помехоустойчивости достигается введением избыточности, что снижает эффективность кода. При кодировании принят следующий подход: вначале устраняют избыточность методами эффективного кодирования, затем в эффективный код вносят избыточность для обеспечения помехоустойчивости.

Коды можно разделить на два класса: блоковые и непрерывные (рекуррентные, древовидные). В блоковом коде последовательность букв разбивается на отрезки по k символов в каждом (на блоки). Каж-

дый блок кодируется отдельно и независимо от других. Количество n>k символов вторичного алфавита, сопоставленное блоку, называется его длиной. Код равномерный, если n=const.

Непрерывными называются такие коды, в которых информационная последовательность подвергается обработке без предварительного разбиения на блоки. Наиболее часто из непрерывных кодов используются свёрточные коды, т.к. они более просто реализуются.

2. Показатели качества эффективных (оптимальных) кодов

Рассматривается случай независимых букв, который базируется на теореме Шеннона о передаче информации по каналу без шума.

Код является однозначно декодируемым, если последовательности кодовых символов, соответствующие различным последовательностям источника, различны. Для равномерных кодов

$$m^n \geq \log M,$$

где n – длина реализации;

m – количество символов вторичного алфавита;

M – длина блока.

Неравномерные коды характеризуются средней длиной кодового слова $\overline{n}$:

$$\overline{n} = \log_2 m \sum_{k=1}^{k} n_k p(a_k), \qquad (1.1)$$

где K – число букв первичного алфавита;

n_k – длина k-го кодового слова;

$p(a_k)$ – вероятность появления k-го кодового слова.

Для неравномерных кодов необходимо между кодовыми комбинациями вводить специальные символы, что эквивалентно расширению первичного алфавита, в качестве неравномерных кодов используются префиксные коды.

Для двоичных кодов (m=2) средняя длина кодового слова составляет

$$\overline{n} = \sum_{k=1}^{k} n_k p(a_k). \qquad (1.2)$$

Верхняя и нижняя границы $\overline{n}$ определяются из неравенства:

$$\frac{H(A)}{\log m} \leq \overline{n} \leq \frac{H(A)}{\log m} + 1,$$

где H(A) – энтропия первичного алфавита;

m – число качественных признаков вторичного алфавита;

n_k – длина k-го кодового слова;

$p(a_k)$ – вероятность появления k-го кодового слова.

Оптимальные (эффективные) коды – коды с практически нулевой избыточностью. Коды с неравномерным распределением символов, имеющие минимальную среднюю длину кодового слова, называют оптимальными неравномерными кодами (ОНК).

Кроме средней длины кодового слова $\overline{n}$ к показателям качества эффективных кодов относятся следующие коэффициенты:

- коэффициент относительной эффективности

$$K_{оэ} = \frac{H(A)}{\overline{n}} = \frac{-\sum_{k=1}^{K} p(a_k)\log_2 p(a_k)}{\log_2 m \sum_{k=1}^{K} n_k p(a_k)}, \qquad (1.3)$$

который показывает, насколько используется статистическая избыточность передаваемого сообщения;

- коэффициент статистического сжатия

$$K_{сс} = \frac{H_{max}(A)}{\overline{n}} = \frac{\log_2 K}{\log_2 m \sum_{k=1}^{K} n_k p(a_k)}, \qquad (1.4)$$

который характеризует уменьшение количества двоичных знаков на букву первичного алфавита при применении ОНК по сравнению с применением методов нестатистического кодирования.

Напомним, что энтропия системы А вычисляется по формуле

$$H(A) = \sum_{k=1}^{K} p(a_k)\log P(a_k). \qquad (1.5)$$

3. Префиксные коды

Договоримся всюду для эффективных кодов считать нумерацию разрядов каждой кодовой последовательности символов слева направо.

Любая последовательность, являющаяся начальной частью последовательности X_k, называется её префиксом.

Например, у последовательности

Cod X_k=010011

префиксами являются следующие последовательности:

0;
01;
010;
0100;
01001.

Код, обладающий свойством префикса (префиксный код), определяется как код, в котором никакое кодовое слово не является префиксом никакого другого кодового слова.

В качестве примера в таблице 1 представлены четыре кода.

Таблица 1. Примеры неравномерных кодов

Буква a_k	Коды				Вероятность $p(a_k)$
	1	2	3	4	
a_1	0	0	0	0	0,5
a_2	0	1	10	01	0,25
a_3	1	00	110	011	0,125
a_4	10	11	111	0111	0,125

Код 1 не декодируется однозначно, так как различным буквам (a_1 и a_2) приписан одинаковый код (0). Код 2 не является однозначно декодируемым, т. к. например, последовательность символов вторичного алфавита 0011 эквивалентна различным последовательностям букв первичного алфавита, например:

$a_1\ a_1\ a_2\ a_2$;
$a_1\ a_1\ a_4$;
$a_3\ a_4\ a_4$;
$a_3\ a_4$.

Коды 3 и 4 однозначно декодируются, причём код 3 обладает свойством префикса, а код 4 – нет.

Таким образом, свойство префикса является достаточным, но не необходимым свойством однозначно декодируемых кодов.

Для декодирования префиксных кодов можно предложить следующий алгоритм:

п.1. Начиная с начала последовательности кодовых слов выделить первую последовательность символов, совпадающую с кодовым словом, и декодировать это слово.

п.2. Повторить п.1, считая началом последовательности первый символ, следующий за только что декодированной последовательностью.

Например, для префиксного кода 3 (таблица 1) последовательность 10111011000111 будет декодирована следующим образом:

Кодовое слово:	10	111	0	110	0	0	111
Буква:	a_2	a_4	a_1	a_3	a_1	a_1	a_4

Оптимальные коды – коды с практически нулевой избыточностью. Оптимальные коды имеют минимальную среднюю длину кодового слова $\overline{n}$. Верхняя и нижняя границы $\overline{n}$ определяются из неравенства:

$$\frac{H(A)}{\log m} \le \overline{n} \le \frac{H(A)}{\log m} + 1,$$

где H(A) – энтропия первичного алфавита;

m – число качественных признаков вторичного алфавита.

Максимально эффективными будут те ОНК, у которых средняя длина кодовых слов равна энтропии источника:

$$\overline{n} = \log_2 m \sum_{k=1}^{K} n_k p(a_k) = H(A).$$

Согласно теореме Шеннона (основной теореме кодирования) средняя длина кодового слова $\overline{n}$ приближается к энтропии источника сообщений H(A).

Коды, обладающие свойством префикса, выгодно отличаются от других кодов тем, что конец кодового слова всегда может быть опознан так, что декодирование может быть выполнено без задержки наблюдаемой последовательности кодовых слов. Поэтому такие коды также называют мгновенными.

Удобно представлять префиксные коды букв первичного алфавита концевыми узлами дерева (листьями). Для кода 3 (таблица 1) кодовое дерево представлено на рисунке 1. Дерево строится следующим образом: множество рёбер, инцидентных корневой вершине, сопоставляется первому разряду кодового слова (ярус с номером I). Множество рёбер, инцидентных каждой вершине первого яруса, сопоставляется второму разряду кодового слова (ярус с номером II) и так далее.

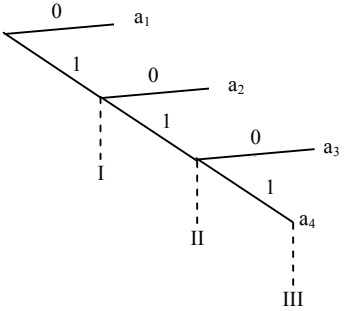

Рисунок 1 – Кодовое дерево (код 3, таблица 1)

Путь от корневой вершины к концевой вершине можно сопоставить коду буквы, соответствующей этой концевой вершине. При этом код представляет последовательность отметок дуг, составляющих этот путь. Очевидно, таким образом с помощью дерева можно описать любой код. Однако если код не обладает свойством префикса, то кодовым словам будут соответствовать не только концевые вершины.

На интуитивном уровне покажем, что минимальному значению средней длины кодового слова (1.2) соответствует такой код, при котором все рёбра, исходящие из одного узла, будут равновероятными. При этом каждый символ будет нести информацию, равную $\log_2 m$.

В нашем примере (код 3, таблица 1) при распределении вероятностей, заданных в таблице 1, каждый символ имеет 1 бит информации. Воспользуемся формулами (1.2) и (1.5).

$\overline{n} = 0{,}5 \times 1 + 0{,}25 \times 2 + 0{,}25 \times 3 = 1{,}75.$

$H(A) = -0{,}5 \log_2 0{,}5 - 0{,}25 \log_2 0{,}25 - 2 \times 0{,}125 \log_2 0{,}125 =$
$= 0{,}5 + 0{,}5 + 0{,}75 = 1{,}75$ (бит/символ).

Таким образом, $\overline{n} = H(A)$, следовательно, код является оптимальным.

Отметим, что для любого основания m всегда с помощью дерева можно выбрать префиксный код.

3.1. Код Шеннона-Фано

Рассмотрим процедуру побуквенного кодирования при отсутствии статистической зависимости между буквами (это нужно только для оценки качества кода).

Цель – для данного первичного алфавита построить оптимальный неравномерный код (ОНК) с минимальным значением средней длины кодового слова $\overline{n}$. Знаки первичного алфавита будем называть буквами, знаки вторичного алфавита – символами.

Алгоритм построения двоичного кода Шеннона-Фано

п.1. Буквы располагаются в порядке убывания вероятностей.

п.2. Буквы разделяются на две группы так, чтобы суммы вероятностей букв в каждой группе были по возможности одинаковыми.

п.3. Всем буквам первой группы присваивается очередной кодовый символ «0», а остальным буквам – символ «1».

п.4. Для каждой из групп повторяются пункты 2 и 3 с целью получения кодового слова до тех пор, пока в каждой группе останется одна буква.

Методика просто распространяется на случай m>2. Для m=2 лучшие результаты получаются, когда вероятности появления букв первичного алфавита равны целым отрицательным степеням двойки.

Пример 1

Построить ОНК по методу Шеннона-Фано, если символы источника сообщений появляются с вероятностями, заданными таблицей 2.

Таблица 2 – Исходные данные к примеру 1

Буква	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
Вероятность	0,04	0,10	0,20	0,11	0,16	0,22	0,02	0,15

Решение

Реализуем алгоритм построения кода Шеннона-Фано по пунктам. Проиллюстрируем работу алгоритма построения кода Шеннона-Фано с помощью таблицы 3.

п.1. В столбце 1 (таблица 3) буквы первичного алфавита отсортированы в порядке убывания вероятностей их появления.

п.2. Разобьем множество букв на две группы:

– первая группа – (a_6, a_3, a_5) с суммарной вероятностью
$$p(a_6)+p(a_3)+p(a_5)=0,22+0,20+0,16=0,58;$$
– вторая группа – (a_8, a_4, a_2, a_1, a_7) с суммарной вероятностью
$$p(a_8)+p(a_4)+p(a_2)+p(a_1)+p(a_7)=0,15+0,11+0,10+0,04+0,02=0,42.$$

Таблица 3 – Построение кода Шеннона Фано (пример 1)

Буква	$p(a_k)$	Код 1	n_k	$n_k\, p(n_k)$	$-p(a_k)\log p(a_k)$
1	**2**	**3**	**4**	**5**	**6**
a_6	0,22	00	2	0,44	0,4806
a_3	0,20	010	3	0,60	0,4644
a_5	0,16	011	3	0,48	0,4230
a_8	0,15	10	2	0,30	0,4105
a_4	0,11	110	3	0,33	0,3503
a_2	0,10	1110	4	0,40	0,3322
a_1	0,04	11110	5	0,20	0,1858
a_7	0,02	11111	5	0,10	0,1129
			$\overline{n}_{Ш-Ф}=2,85$		H(A)=2,7597

п.3. Первым символам кодовых слов букв a_6, a_3, a_5 присваиваем символ «0», а первым символам кодовых слов букв a_8, a_4, a_2, a_1, a_7 присваиваем символ «1» (столбец 3, таблица 3).

п.4. Повторяется **п.2** для первой группы, которая разбивается на две подгруппы: (a_6) и (a_3, a_5). В соответствии с **п.**3 второму символу буквы a_6 присваивается символ «0», и на этом процесс кодирования буквы a_6 заканчивается:

$$\text{Cod } a_6 = 00.$$

Вторым символам группы (a_3, a_5) присваивается символ «1».

Повторяется **п.2** для группы (a_3, a_5), она разбивается на буквы a_3 и a_5.

В соответствии с **п.**3 третьему символу буквы a_3 присваивается символ «0», а третьему символу буквы a_5 – символ «1», и на этом процесс кодирования букв a_3 и a_5 заканчивается:

$$\text{Cod } a_3 = 010,$$
$$\text{Cod } a_5 = 011.$$

Аналогично проводится кодирование остальных символов букв a_8, a_4, a_2, a_1, a_7, результаты которого представлены в столбце 3, таблица 3.

Оценим эффективность построенного кода.

Среднюю длину кодового слова вычислим по формуле (1.2), для чего для каждой буквы первичного алфавита воспользуемся данными из столбцов 4 и 2 таблицы 3, поместим полученные произведения в соответствующие ячейки столбца 5 и просуммируем их ($\overline{n}_{Ш-Ф} = 2{,}85$).

Вычислим энтропию первичного алфавита по известной формуле (1.5). В столбце 6 таблицы 3 произведены соответствующие вычисления:

$$H(A) = 2{,}7597 \text{ бит/символ}.$$

Вычислим коэффициент относительной эффективности по формуле (1.3)

$$K_{оэ}^{Ш-Ф} = \frac{2{,}7597}{2{,}85} = 0{,}9683$$

и коэффициент статистического сжатия – по формуле (1.4)

$$K_{сс}^{Ш-Ф} = \frac{\text{Log}_2 8}{2{,}85} = \frac{3{,}0}{2{,}85} = 1{,}0526.$$

Если для алфавита, заданного таблицей 2, будет построен другой код, для него надо вычислить значения коэффициента относительной эффективности и коэффициента статистического сжатия и провести сравнения. Более эффективным окажется тот код, для которого значения коэффициента относительной эффективности и коэффициента статистического сжатия окажутся большими.

Пример 2

Рассмотренная методика Шеннона-Фано не всегда приводит к однозначному построению кода, так как при разбиении на подгруппы можно сделать большей по сумме вероятностей как верхнюю, так и нижнюю подгруппы.

Например, множество букв, приведенных в таблице 3 (пример 1), можно разбить следующим образом:

- первая группа – (a_6, a_3) с суммарной вероятностью
$$p(a_6)+p(a_3)=0,22+0,20=0,42;$$
- вторая группа – (a_5, a_8, a_4, a_2, a_1, a_7) с суммарной вероятностью
$$p(a_5)+p(a_8)+p(a_4)+p(a_2)+p(a_1)+p(a_7)=$$
$$=0,16+0,15+0,11+0,10+0,04+0,02=0,58.$$

Постройте код Шеннона-Фано для этого варианта разбиения множества букв на подгруппы, вычислите значения коэффициента относительной эффективности и коэффициента статистического сжатия, проведите сравнительный анализ результатов решения примера 2 с результатами решения примера 1.

3.2. Двоичный код Хаффмена

Теоретической основой построения двоичного кода Хаффмена являются две теоремы Хаффмена.

Теорема Хаффмена об упорядочении букв

Для любого источника с числом букв $K \geq 2$ существует оптимальный префиксный двоичный код, в котором два наименее вероятных кодовых слова a_K и a_{K-1} имеют одну и ту же длину и отличаются лишь последними символами: код буквы a_K оканчивается на «1», а код буквы a_{K-1} – на «0».

Теорема сводит задачу построения оптимального кода для ансамбля $A=\{a_1, a_2,\ldots,a_K\}$ к задаче построения оптимального кода для редуцированного ансамбля $A' = \{a'_1, a'_2, \ldots, a'_{K-1}\}$, в котором вероятности появления букв $p(a'_i)$ равны:

$$p(a'_i) = \begin{cases} p(a_i), & \text{для } i \leq K-2; \\ p(a_{K-1}) + p(a_K) & \text{для } i = K-1. \end{cases}$$

Теорема Хаффмена об оптимальности кода

Если A' есть редуцированный ансамбль А и если некоторый префиксный код для A' является оптимальным, то соответствующий ему префиксный код для А также является оптимальным.

Метод всегда приводит к получению оптимального множества слов в том смысле, что никакое другое множество не имеет меньшего среднего числа символов на сообщение.

Алгоритм построения двоичного кода Хаффмена

При построении алгоритма на очередном шаге будем нумеровать К-м и (К–1)-м номерами буквы с двумя наименьшими вероятностями $p(a_{K-1})$ и $p(a_K)$, причем $p(a_{K-1}) \geq p(a_K)$

п.1. Выбираем две буквы a_K и a_{K-1} с минимальными вероятностями $p(a_{K-1})$ и $p(a_K)$.

п.2. Полагаем, что последний символ буквы a_K равен «1», а последний символ буквы a_{K-1} – «0».

п.3. Редуцируя ансамбль, объединяем буквы a_K и a_{K-1} в одну вспомогательную букву, которой приписываем суммарную вероятность $p(a_K)+p(a_{K-1})$.

п.4. Для редуцированных ансамблей повторяем пункты 1-3, определяя очередной символ кода двух наименее вероятных обобщённых букв до тех пор, пока не получим редуцированный ансамбль из одной буквы, которой соответствует единичная вероятность.

п.5. Для каждой буквы первичного алфавита строим двоичное кодовое слово, которому соответствует путь на кодовом дереве от корневой вершины к соответствующей концевой вершине по отметкам дуг из множества {0, 1}.

Пример 3

Построить двоичный код Хаффмена для последовательности букв алфавита, заданного таблицей 2 (пример 1). Провести сравнительный анализ результатов построения кода Хаффмена и кода Шеннона-Фано.

Р е ш е н и е

Шаг 1: п.1. Выбираем две буквы с наименьшими вероятностями $a_{K-1}=a_1$ и $a_K=a_7$:

$$p(a_1)=0{,}04;$$
$$p(a_7)=0{,}02.$$

п.2. Последнему символу кода буквы a_1 присваиваем значение «0» (на рисунке 2 – дуга, инцидентная концевой вершине a_1, получила отметку «0»), а последнему символу буквы a_7 – значение «1» (дуга, инцидентная концевой вершине a_7, получила отметку «1»).

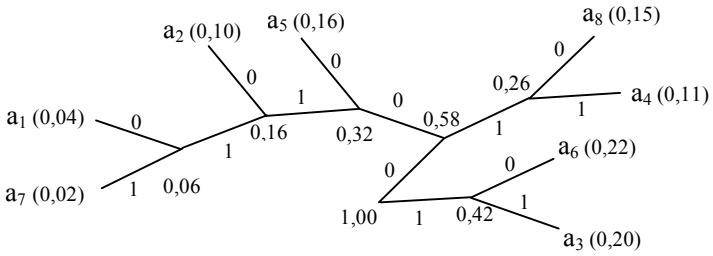

Рисунок 2 – Кодовое дерево (пример 3)

п.3. Объединяем буквы a_1 и a_7 в одну букву с вероятностью

$$p(a_1)+p(a_7)=0,02+0,04=0,06,$$

редуцируя исходный ансамбль A.

Шаг 2: п.4. Повторяем **пункты 1,2,3**, редуцируя ансамбль A': объединяем буквы с наименьшими вероятностями (объединенные на **шаге 1** буквы a_1 и a_7 и букву a_2):

$$p(a_1, a_7)+p(a_2)=0,06+0,10=0,16.$$

Соответствующие дуги помечаем символами «0» (дуга с весом 0,10) и «1» (дуга с весом 0,06).

Шаг 3: Повторяем **пункты 1,2,3**, редуцируя ансамбль A', объединяем буквы a_4 и a_8 с наименьшими вероятностями:

$$p(a_4)+p(a_8)=0,11+0,15=0,26$$

и делаем соответствующие отметки «1» и «0» на дугах, инцидентных объединяемым вершинам (далее последние действия производим без комментариев).

Шаг 4: объединяем буквы с наименьшими вероятностями (объединенные на **шаге 2** буквы a_1, a_7, a_2 и букву a_5):

$$p(a_1, a_7, a_2)+p(a_5)=0,16+0,16=0,32.$$

Шаг 5: Объединяем буквы с наименьшими вероятностями (объединенные на **шаге** 3 буквы a_4 и a_8 с объединенными на **шаге 4** буквами (a_1, a_7, a_2, a_5)

$$p(a_4, a_8)+p(a_1, a_7, a_2, a_5)=0,26+0,32=0,58.$$

Шаг 6: Объединяем буквы с наименьшими вероятностями a_3, a_6:

$$p(a_3)+p(a_6)=0,20+0,22=0,42.$$

Шаг 7: Воспользуемся результатами объединения букв на **шаге 6** (буквы a_3 и a_6) и на **шаге 5** (буквы a_1, a_7, a_2, a_5, a_4, a_8) для получения последней буквы с вероятностью 1,0:

$$p(a_3, a_6)+p(a_1, a_7, a_2, a_5, a_4, a_8)=0,42+0,58=1,00.$$

На этом редуцирование исходного ансамбля А завершено (на рисунке 2 получена корневая вершина кодового дерева с отметкой с суммарной вероятностью, равной 1,0).

Завершает построение двоичного кода Хаффмена реализация **п.5**:

Cod a_1=00010;
Cod a_2=0000;
Cod a_3=11;
Cod a_4=011;
Cod a_5=001;
Cod a_6=10;
Cod a_7=00011;
Cod a_8=010.

Анализ результатов

Построенный код является префиксным. В этом легко убедиться, сравнив попарно кодовые слова, – ни одно кодовое слово не является префиксом никакого другого кодового слова.

Коды букв a_7 и a_1 с наименьшими вероятностями имеют одинаковую длину ($n_7=n_1=5$) и отличаются лишь последним символом.

Согласно формуле (1.2) средняя длина кодового слова составляет

$$n_{Хаф}=5\times 0,04+4\times 0,10+2\times 0,20+3\times 0,11+3\times 0,16+2\times 0,22+5\times 0,02+$$
$$+3\times 0,15=0,2+0,4+0,40+0,33+0,48+0,44+0,1+0,45=2,80.$$

Распределение вероятностей букв первичного алфавита в примерах 1 и 3 одинаковые, поэтому при вычислении значения коэффициента относительной эффективности воспользуемся значением энтропии первичного алфавита, вычисленного в столбце 6 таблицы 3:

$$K_{оэ}^{Хаф}=\frac{2,7597}{2,80}=0,9856.$$

Вычислим коэффициент статистического сжатия для построенного двоичного кода Хаффмена:

$$K_{сс}^{Хаф}=\frac{Log_2 8}{2,80}=\frac{3,0}{2,80}=1,0714.$$

Сравним показатели качества кода Хаффмена (пример 3) и кода Шеннона-Фано (пример 1):

$$\overline{n}_{Хаф} = 2{,}80 < \overline{n}_{Ш-Ф} = 2{,}84;$$

$$K_{оэ}^{Хаф} = 0{,}9856 > K_{оэ}^{Ш-Ф} = 0{,}9683;$$

$$K_{сс}^{Хаф} = 1{,}0714 > K_{сс}^{Ш-Ф} = 1{,}0526.$$

Из соотношений показателей делаем вывод: построенный код Хаффмена эффективнее кода Шеннона-Фано.

3,3 Методика Хаффмена для недвоичных кодов

Преимущество метода Хаффмена сказывается при построении ОНК для вторичных алфавитов с числом качественных признаков m>2 (например, если сообщения передаются с помощью трех или более частот). Большая эффективность достигается за счет более строгого выбора числа L наименее вероятных букв первичного алфавита, объединяемых на первом этапе построения кодового дерева.

Число L этих букв должно удовлетворять условиям

$$2 \le L \le m, \tag{1.5}$$

кроме того,

$$\frac{K - L}{m - 1} = f, \tag{1.6}$$

где f – целое положительное число;

K – число букв первичного алфавита;

m – количество качественных признаков вторичного алфавита.

Хаффмен показал, что для получения минимально возможного значения средней длины кодового слова кода $\overline{n}$ необходимо и достаточно выполнения следующих условий:

- при $p(a_i)>p(a_t)$ длины i-го и t-го кодовых слов должны находиться в отношении $n_t \ge n_i$;
- L букв первичного алфавита с наименьшими вероятностями имеют кодовые слова одинаковой длины, отличающиеся друг от друга последним символом;
- любая возможная последовательность (n_K-1) кодовых символов должна либо сама быть кодовой комбинацией, либо иметь своим префиксом разрешенную кодовую комбинацию.

Алгоритм построения недвоичного кода Хаффмена

При выполнении алгоритма удобно строить недвоичное кодовое дерево и помечать дуги, входящие в очередную вершину, символами из множества {0, 1,…,m}.

п.1. Объединить

$$L=2+R_{m-1}(K-2) \qquad (1.7)$$

букв с наименьшими вероятностями в некоторую новую букву с вероятностью, равной сумме вероятностей объединяемых букв. В (1.7) через $R_{m-1}(K-2)$ обозначен остаток от деления (K–2) на (m-1).

Полагаем, что последний символ буквы a_{K-L+1} равен «0», последний символ буквы a_{K-L+2} – «1», последний символ буквы a_{K-L+3} – «2» и так далее до «L».

п.2. В редуцированном ансамбле определить m букв с наименьшими вероятностями, объединить их в обобщенную букву и присвоить очередному символу каждой из объединяемых букв значения 1, 2,…, m.

п.3. Повторять **п.2.** до тех пор, пока не получится редуцированный ансамбль из одной буквы, которой соответствует единичная вероятность.

п.4. Для каждой буквы первичного алфавита строим недвоичное кодовое слово, которому соответствует путь на кодовом дереве от корневой вершины к соответствующей концевой вершине по отметкам дуг из множества {0, 1,…, m}.

Пример 4

Построить недвоичный код Хаффмена для последовательности букв алфавита, заданного таблицей 2 (пример 1). Провести сравнительный анализ результатов построения двоичного кода Хаффмена (пример 3) и недвоичного кода Хаффмена (m=3).

Решение

Шаг 1: п.1. По формуле (1.7) вычисляем число букв, объединяемых на первом этапе:

$$L=2+R_{3-1}(8-2)=2+ R_2(6)=2+0=2.$$

Выбираем две буквы с наименьшими вероятностями $a_{K-1}=a_1$ и $a_K=a_7$:

$$p(a_1)=0,04;$$
$$p(a_7)=0,02.$$

Последнему символу кода буквы a_1 присваиваем значение «0» (на рисунке 3 дуга, инцидентная концевой вершине a_1, получила отметку «0»), а последнему символу буквы a_7 – значение «1» (дуга, инцидентная концевой вершине a_7, получила отметку «1»). Объединяем буквы a_1 и a_7 в одну букву с вероятностью

$$p(a_1)+p(a_7)=0,02+0,04=0,06,$$

редуцируя исходный ансамбль A.

Шаг 2: п.2. Продолжаем редуцировать ансамбль A' : объединяем три буквы с наименьшими вероятностями (объединенные на **шаге 1** буквы a_1 и a_7 и буквы a_2 и a_4):

$$p(a_1, a_7)+p(a_2)+p(a_4)==0,06+0,10+0,11=0,27.$$

Соответствующие дуги помечаем символами «0» (дуга с весом 0,11) «1» (дуга с весом 0,10) и «2» (дуга с весом 0,06).

Шаг 3: Повторяем **пункт 2**, редуцируя ансамбль A' , объединяем буквы с наименьшими вероятностями a_8, a_5 и a_3 и делаем соответствующие отметки на дугах, инцидентных соответствующим вершинам: «0» (дуга с весом 0,20), «1» (дуга с весом 0,16) и «2» дуга с весом 0,15, инцидентных объединяемым вершинам (далее последние действия с отметками дуг производим без комментариев):

$$p(a_8)+p(a_5)+p(a_3)=0,15+0,16+0,20=0,51.$$

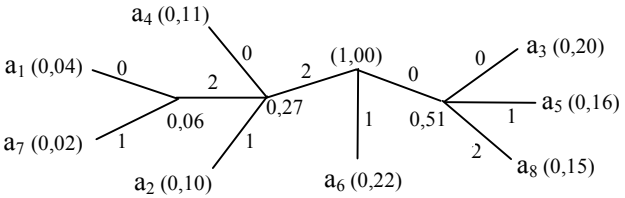

Рисунок 3 – Кодовое дерево (пример 4)

Шаг 4: Объединяем три буквы – букву a_6, букву, объединенную на шаге 2, и букву, объединенную на шаге 3:

$$p(a_6)+p(a_1, a_7, a_2, a_4)+ p(a_8, a_5 a_3)=0,22+0,27+0,51=1,00.$$

На этом редуцирование исходного ансамбля А завершено (на рисунке 3 получена корневая вершина кодового дерева с суммарной вероятностью, равной 1,0). Завершает построение недвоичного кода Хаффмена (m=3) реализация **п.4**:

Cod a_1=120;
Cod a_2=11;
Cod a_3=00;
Cod a_4=10;
Cod a_5=01;
Cod a_6=2;
Cod a_7=121;
Cod a_8=02.

Анализ результатов

Построенный код является префиксным. В этом легко убедиться, сравнив попарно кодовые слова, – ни одно кодовое слово не является префиксом никакого другого кодового слова.

Коды букв a_7 и a_1 с наименьшими вероятностями имеют одинаковую длину $n_7=n_1=3$ и отличаются лишь последним символом.

Согласно формуле (1.1) средняя длина кодового слова составляет

$$\overline{n}_{Хаф}(m=3)=\log 2\ 3\times(3\times 0{,}04+2\times 0{,}1+2\times 0{,}20+2\times 0{,}11+2\times 0{,}16+$$
$$+1\times 0{,}22+ +3\times 0{,}02+2\times 0{,}15=1.585\times(0{,}12+0{,}2+0{,}40+0{,}22+$$
$$+0{,}32+0{,}22+0{,}06+0{,}3)=1.585\times 1{,}84=2.916.$$

Распределение вероятностей букв первичного алфавита в примерах 1 и 4 одинаковые, поэтому при вычислении значения коэффициента относительной эффективности воспользуемся значением энтропии первичного алфавита, вычисленного в столбце 6 таблицы 3:

$$K_{оэ}^{Хаф}(m=3)=\frac{2{,}7597}{2{,}916}=0{,}9464.$$

Вычислим коэффициент статистического сжатия для построенного недвоичного кода Хаффмена:

$$K_{сс}^{Хаф}(m=3)=\frac{\log_2 8}{2{,}916}=\frac{3{,}0}{2{,}916}=1{,}0288.$$

Сравним показатели качества недвоичного(m= 3) и двоичного кодов Хаффмена:

$$\overline{n}_{Хаф}(m=2)=2{,}80<\overline{n}_{Хаф}(m=3)=2{,}916;$$
$$K_{оэ}^{Хаф}(m=2)=0{,}9856>K_{оэ}^{Хаф}m=3)=0{,}9464;$$
$$K_{сс}^{Хаф}(m=2)=1{,}0714>K_{сс}^{Хаф}(m=3)=1{,}0288.$$

Из соотношений показателей делаем вывод: двоичный код Хаффмена оптимальнее недвоичного кода Хаффмена, одной из причин является то обстоятельство, что на первом этапе (пример 4) объединяются две буквы вместо потенциально возможных трех (например, при K=9 или K=11).

3.3. Блочное кодирование

Повышения эффективности при кодировании сообщений можно добиться, кодируя блоки букв первичного алфавита одинаковой длины (при фиксированном размере блока формируются все возможные сочетания с повторениями). С увеличением длины блоков уменьшается

среднее число символов вторичного алфавита на букву первичного алфавита.

Пример 5

Рассмотрим процедуру эффективного кодирования сообщений, образованных с помощью алфавита, состоящего всего из двух независимых букв A и B с вероятностями появления p(A)=0,9 и p(B)=0,1.

Так как вероятности появления букв A и B не равны, последовательность из таких букв обладает избыточностью.

Случай 1 – кодирование по одной букве. На передачу каждой буквы требуется один двоичный знак ($\overline{n}_I=1$):

Cod A=0;
Cod B=1.

Согласно (1.5) энтропия первичного алфавита при побуквенном кодировании составляет

$H_I(A)=-\log_2 0{,}9-\log_2 0{,}1=0{,}1368+0{,}3322=0{,}4690$ (бит/символ).

Вычислим коэффициент относительной эффективности (1.3) и коэффициент статистического сжатия (1.4):

$$K_{o.э.I}=\frac{H_I(A)}{\overline{n}_I}=\frac{0{,}469}{1}=0{,}469;$$

$$K_{c.c.I}=\frac{H_{max\,I}(A)}{\overline{n}_I}=\frac{\log_2 2}{1}=1.$$

При побуквенном кодировании никакого эффекта не получено.

Случай 2 – кодирование блоков, содержащих по две буквы. В таблице 4 представлен код Шеннона-Фано для этого случая. Так как знаки A и B статистически не связаны, вероятности каждого блока определяются как произведение вероятностей составляющих букв ($\overline{N}_{II}$ – среднее число двоичных знаков, приходящееся на один блок из двух букв).

Таблица 4 – Кодирование блоков по две буквы

Блок	Вероятность блока q_j	Код	n_j	n_jq_j
AA	0,81	0	1	0,81
AB	0,09	10	2	0,18
BA	0,09	110	3	0,27
BB	0,01	111	3	0,03
$\sum_{j=1}^{4} q_j = 1{,}00$		$\overline{N}_{II}=1{,}29$		

Так как каждый блок содержит две буквы, среднее число двоичных символов на букву

$$\overline{n}_{II} = \frac{\overline{N}_{II}}{2} = \frac{1{,}29}{2} = 0{,}645.$$

Согласно (1.5) энтропия алфавита, образованного блоками из двух букв, составляет

$H_{II}(A) = -\log_2 0{,}81 - 2\log_2 0{,}09 - \log_2 0{,}01 =$

$= 0{,}2462 + 0{,}6253 + 0{,}0664 = 0{,}9379$ (бит/символ).

Вычислим коэффициент относительной эффективности (1.3) и коэффициент статистического сжатия (1.4):

$$K_{о.э.II} = \frac{H_{II}(A)}{\overline{N}_{II}} = \frac{0{,}9379}{1{,}29} = 0{,}7271;$$

$$K_{с.с.II} = \frac{H_{max\,II}(A)}{\overline{N}_{II}} = \frac{\log_2 4}{1{,}29} = \frac{2}{1{,}29} = 1{,}5504.$$

Случай 3 – кодирование блоков, содержащих по три буквы. В таблице 5 представлен код Шеннона-Фано для этого случая, $\overline{N}_{III}$ – среднее количество двоичных знаков, приходящееся на один блок из трех букв.

Таблица 5 – Кодирование блоков по три буквы

Блок	Вероятность q_j	Код	n_j	$n_j q_j$
AAA	0,729	0	1	0,729
AAB	0,081	100	3	0,243
ABA	0,081	101	3	0,243
BAA	0,081	110	3	0,243
ABB	0,009	11100	5	0,045
BAB	0,009	11101	5	0,045
BBA	0,009	11110	5	0,045
BBB	0,001	11111	5	0,005
$\sum_{j=1}^{4} q_j = 1{,}00$		$\overline{N}_{III} = 1{,}598$		

Так как каждый блок содержит три буквы, среднее число двоичных знаков на букву

$$\overline{n}_{III} = \frac{\overline{N}_{III}}{3} = \frac{1{,}598}{3} = 0{,}533.$$

Согласно (1.5) энтропия алфавита, образованного блоками из трех букв, составляет

$H_{III}(A)= -\log_2 0{,}729- 3\log_2 0{,}081- 3\log_2 0{,}009- \log_2 0{,}001=$
$=0{,}3324+0{,}8811+0{,}1835 +0{,}0100=1{,}4070$ (бит/символ).

Вычислим коэффициент относительной эффективности (1.3) и коэффициент статистического сжатия (1.4):

$$K_{о.э.III} = \frac{H_{III}(A)}{\overline{N}_{III}} = \frac{1{,}4070}{1{,}598} = 0{,}8805;$$

$$K_{с.с.III} = \frac{H_{max\,III}(A)}{\overline{N}_{III}} = \frac{\log_2 8}{1{,}598} = \frac{3}{1{,}598} = 1{,}7773.$$

Анализ результатов

(1) Согласно теореме Шеннона для канала без шума теоретический минимум среднего числа двоичных знаков на букву алфавита $\overline{n}$ может быть достигнут при кодировании блоков, включающих бесконечное число букв:

$$\lim_{n\to\infty} \overline{n} = H(A).$$

Действительно, с увеличением количества букв в блоке среднее число символов на букву уменьшается:

$$\overline{n}_{III} = 0{,}533 < \overline{n}_{II} = 0{,}645 < \overline{n}_{I} = 1$$

и его значение приближается к значению энтропии исходного алфавита, состоящего из двух букв. Вычислим, насколько значение среднего числа двоичных символов на букву алфавита $\overline{n}_{III}$ при кодировании трехбуквенными блоками больше значения энтропии исходного алфавита $H_I(A)$:

$$\frac{\overline{n}_{III} - H_I(A)}{H_I(A)} \times 100\% = \frac{0{,}533 - 0.469}{0{,}469} \times 100\% = 13{,}65\%.$$

Значение $\overline{n}_{III}$ всего на 13,65% больше энтропии заданного алфавита, что гораздо меньше разницы в 53,1% при побуквенном кодировании.

Следует подчеркнуть, что увеличение эффективности кодирования при укрупнении блоков не связано с учетом все более далеких статистических связей, так как в данном примере рассматривается некоррелированная последовательность знаков.

(2) Сравнивая полученные данные, можно убедиться, что с укрупнением кодируемых блоков разница значений между средним числом знаков на блок $\overline{N}$ и энтропией каждого алфавита H(A) быстро уменьшается:

$\overline{N}_I - H_I(A) = 1 - 0{,}4690 = 0{,}531;$

$\overline{N}_{II} - H_{II}(A) = 1{,}29 - 0{,}9379 = 0{,}3521;$

$\overline{N}_{III} - H_{III}(A) = 1{,}598 - 1{,}407 = 0{,}191.$

(3) Сравним эффективность ОНК с помощью коэффициентов относительной эффективности

$$K_{о.э.III}=0{,}8805 > K_{о.э.II}=0{,}7271 > K_{о.э.I}=0{,}469$$

и коэффициентов статистического сжатия

$$K_{с.с.III}=1{,}7773 > K_{с.с.II}=1{,}5504 > K_{с.с.I}=1{,}0.$$

По мере укрупнения блоков эффективность ОНК растет.

4. Список задач

4.1. Построить оптимальный код сообщения, состоящего из:

a) пяти равновероятных букв;

b) шести равновероятных букв;

c) семи равновероятных букв;

d) восьми равновероятных букв.

Дать оценку эффективности построенных кодов. В каких случаях код, построенный для первичного алфавита с равновероятным появлением букв, окажется самым эффективным?

4.2. Первичный алфавит состоит из семи букв (K=6). Построить оптимальный код Шеннона Фано, если вероятность появления букв подчиняется закону $p_i = \left(\frac{1}{2}\right)^i$. Дать оценку построенному коду. Точность вычислений – до четвертого знака.

4.3. Не строя ОНК, напишите две последние кодовые комбинации для первичного алфавита из 20 букв, если известно, что вероятность появления в сообщениях каждой очередной буквы в два раза меньше вероятности появления предыдущей буквы?

4.4. Построить оптимальный код сообщения, если вероятности появления букв первичного алфавита распределены в соответствии:

a) с таблицей 6;

b) с таблицей 7;

c) с таблицей 8;

d) с таблицей 9.

Дать оценку эффективности построенного кода. Какие положения теории построения эффективных кодов нашли подтверждение?

Таблица 6

Ai	A1	A2	A3	A4	A5	A6
p(Ai)	1/2	1/8	1/8	1/8	1/16	1/16

Таблица **7**

Ai	A1	A2	A3	A4	A5	A6	A7	A8
p(Ai)	1/4	1/4	1/8	1/8	1/16	1/16	1/16	1/16

Таблица 8

Ai	A1	A2	A3	A4	A5	A6	A7	A8
p(Ai)	1/2	1/8	1/8	1/16	1/16	1/16	1/32	1/32

Таблица 9

Ai	A1	A2	A3	A4	A5	A6	A7	A8
p(Ai)	1/2	1/4	1/8	1/16	1/32	1/64	1/128	1/128

4.5. Построить методом Хаффмена двоичный ОНК для алфавита со следующим распределением вероятностей появления букв в тексте: p(A)=0,45; p(B)=0,16; p(C)=0,13; p(D)=0,11; p(E)=0,05; p(F)=0,04; p(G)=0,03; p(H)=0,02; p(I)=0,01. Вычислить коэффициенты относительной эффективности и статистического сжатия.

4.6. Построить методом Хаффмена недвоичный ОНК (m=3) для алфавита со следующим распределением вероятностей появления букв в тексте: p(A1)=0,08; p(A2)=0,01; p(A3)=0,09; p(A4)=0,18; p(A5)=0,07; p(A6)=0,05; p(A7)=0,13; p(A8)=0,22; p(A9)=0,015; p(A10)=0,06; p(A11)=0,035; p(A12)=0,06. Вычислить коэффициенты относительной эффективности и статистического сжатия.

4.7. Вычислить вероятность появления знаков первичного алфавита во фразе «Учиться, учиться и учиться». Построить двоичный ОНК знаков первичного алфавита. Чему равна средняя длина кодовых слов построенного ОНК? Построить во вторичном алфавите заданную фразу.

4.8. Вычислить вероятность появления слов во фразе «Учиться, учиться и учиться» (в данном случае в список слов необходимо включить и знаки «,», «пробел», «и»). Построить двоичный ОНК слов. Чему равна средняя длина кодовых слов построенного ОНК? Построить во вторичном алфавите заданную фразу. Сравните результаты решения задач 4.7 и 4.8.

4.9. Первичный алфавит состоит из двух букв А и В. Построить ОНК для передачи сообщений, если кодировать по одной, две и три буквы в блоке. Сравнить эффективность полученных кодов. Вероятности появления букв первичного алфавита имеют следующие значения:

a) p(A)=0,87; p(B)=0,13;
b) p(A)=0,80; p(B)=0,20;
c) p(A)=0,70; p(B)=0,30;
d) p(A)=0,60; p(B)=0,40.

4.10. Первичный алфавит состоит из трех букв А, В и С. Построить ОНК для передачи сообщений, если кодировать по одной, две и три буквы в блоке. Сравнить эффективность полученных кодов. Вероятности появления первичного алфавита имеют следующие значения:

a) p(A)=0,6; p(B)=0,3; p(C)=0,1;
b) p(A)=0,4; p(B)=0,4; p(C)=0,2;
c) p(A)=0,5; p(B)=0,3; p(C)=0,2;
d) p(A)=0,7; p(B)=0,2; p(C)=0,1.

4.11. В таблице 10 представлены:

- в строках 1, 3, 5 – номера вариантов с первого по 24;
- в строках 2, 4, 6 – соответствующие номерам вариантов значения числа качественных признаков вторичного алфавита m;
- в строке 10 – буквы первичного алфавита и соответствующие им вероятности p(Ai) для каждого из вариантов (для i-го варианта множество значений вероятностей находится в соответствующем столбце).

Для заданного варианта требуется:
(1) Построить оптимальный неравномерный двоичный код методом Шеннона-Фано.

(2) Для заданного значения качественных признаков вторичного алфавита m построить код Хаффмена.

(3) Для построенных в п.п. (1) и (2) кодов вычислить коэффициенты относительной эффективности и статистического сжатия.

(4) Провести сравнительный анализ результатов.

Таблица 10 – Исходные данные к задаче 4.11

№	№ варианта, m, $p(A_i)$								
1	Вариант	1	2	3	4	5	6	7	8
2	m	3	4	2	3	4	2	3	4
3	Вариант	9	10	11	12	13	14	15	16
4	m	4	2	3	4	2	3	4	2
5	Вариант	17	18	19	20	21	22	23	24
6	m	2	3	4	2	3	4	2	3
7	A_1	0,21	0,04	0,14	0,05	0,17	0,2	0,09	0,13
	A_2	0,15	0,03	0,03	0,04	0,04	0,01	0,03	0,10
	A_3	0,05	0,01	0,08	0,03	0,06	0,04	0,04	0,05
	A_4	0,03	0,16	0,09	0,09	0,08	0,03	0,11	0,05
	A_5	0,09	0,07	0,07	0,03	0,15	0,05	0,05	0,12
	A_6	0,04	0,03	0,12	0,06	0,12	0,06	0,05	0,04
	A_7	0,06	0,04	0,04	0,20	0,07	0,07	0,14	0,06
	A_8	0,01	0,12	0,02	0,10	0,02	0,2	0,16	0,25
	A_9	0,03	0,16	0,03	0,18	0,01	0,11	0,03	0,07
	A_{10}	0,07	0,19	0,12	0,20	0,10	0,13	0,19	0,11
	A_{11}	0,17	0,15	0,11	0,02	0,11	0,04	0,02	0,02
	A_{12}	0,09	-	0,13	-	0,07	0,04	0,09	-
	A_{13}	-	-	0,02	-	-	0,02	-	-

5. Контрольные вопросы

5.1. Сформулируйте и поясните основную теорему Шеннона о кодировании для канала без шума.
5.2. Какой код называется однозначно декодируемым?
5.3. Какой код называется префиксным?
5.4. Сформулируйте теорему Крафта.
5.5. Как строится код Шеннона-Фано?
5.6. За счет чего при эффективном кодировании уменьшается средняя длина кодовой комбинации?
5.7. До какого предела может уменьшиться средняя длина кодовой комбинации при эффективном кодировании?
5.8. Представьте показатели качества эффективных кодов.
5.9. Сформулируйте теорему Хаффмена об упорядочении букв.
5.10. Сформулируйте теорему Хаффмена об оптимальности кода.
5.11. Как строится двоичный код Хаффмена?
5.12. Как строится код Хаффмена для недвоичного алфавита?
5.13. В чем преимущество методики построения эффективного кода, предложенной Хаффменом, по сравнению с методикой Шеннона-Фано?
5.14. Представьте пример реализации блочного кодирования при построении оптимального неравномерного кода.
5.15. При каком распределении букв первичного алфавита ОНК оказывается самым эффективным?
5.16. В каких случаях код, построенный для первичного алфавита с равновероятным появлением букв, окажется самым эффективным?
5.17. Какой вид имеют две последние кодовые комбинации для первичного алфавита из K букв, если известно, что вероятность появления в сообщениях каждой очередной буквы в два раза меньше вероятности появления предыдущей буквы?

Библиографический список

1. Дмитриев В.И. Прикладная теория информации/В.И.Дмитриев. – М.: Высшая школа, 1989. – 320с.
2. Кузьмин И.В., Кедрус В.А. Основы теории информации и кодирования/Кузьмин И.В., Кедрус В.А. – Киев: Вища школа, 1977. – 280с.
3. Фано Р. Передача информации. Статистическая теория связи/Р.Фано – М.: Мир, 1965. – 438с.
4. Хемминг Р.В. Теория кодирования и теория информации/Р.В. Хемминг – М.: Радио и связь, 1983. – 176с.
5. Цымбал В.П. Задачник по теории информации и кодированию/В.П.Цымбал.- К.: Вища шк., 1976. – 276с.
6. Цымбал В.П.Теория информации и кодирование/В.П.Цымбал.- К.: Вища шк., 1992. – 261с.
7. Шеннон К. Работы по теории информации и кибернетике/К.Шеннон – М: Иностранная литература, 1963.
8. Методические указания к практическим занятиям, часть 1, по дисциплине «Теория информации и кодирование» на тему «Вычисление информационных характеристик систем связи»для студентов специальности 7.091501 «Компьютерные системы и сети» дневной и заочной форм обучения./Сост. Щепин Ю.Н. – Севастополь, Изд-во СевНТУ, 2003. – 24с.

Заказ № ___ от «___» _____ 2010 г. Тираж ___ экз.
Изд-во СевНТУ