

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра систем сбора и обработки данных
Кафедра вычислительной техники

“УТВЕРЖДАЮ”
ДЕКАН АВТФ

к.т.н. Рева И. Л.

“ ____ ” _____ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Теория информации

Образовательная программа: 09.03.02 Информационные системы и технологии, профиль:
Информационные системы в промышленности и бизнесе

Курс: 2, семестр : 4

Факультет автоматики и вычислительной техники,

		Семестр
№	Вид деятельности	4
1	Всего зачетных единиц (кредитов)	4
2	Всего часов	144
3	Всего занятий в контактной форме, час.	81
4	Лекции, час.	36
5	Практические занятия, час.	0
6	Лабораторные занятия, час.	36
7	из них в активной и интерактивной форме, час.	12
8	Аттестация, час.	2
9	Консультации, час.	7
10	Самостоятельная работа, час.	63
11	Виды самостоятельной работы (курсовой проект, курсовая работа, РГЗ, подготовка к контрольной работе)	РГЗ
12	Вид аттестации	ДЗ

Рабочая программа составлена на основании федерального государственного образовательного стандарта высшего образования по направлению (специальности): 09.03.02 Информационные системы и технологии

ФГОС введен в действие приказом №219 от 12.03.2015 г. , дата утверждения: 30.03.2015 г.

Место дисциплины в структуре учебного плана: Блок 1, вариативная, по выбору студента

Рабочая программа разработана на основе компетентностной модели выпускника по направлению (специальности): 09.03.02 Информационные системы и технологии

Рабочая программа обсуждена на заседании кафедры

ССОД, протокол заседания кафедры №2/1 от 20.06.2017
ВТ, протокол заседания кафедры №6 от 20.06.2017

Утверждена на совете факультета автоматике и вычислительной техники, протокол № 6 от 21.06.2017

Программу разработал:

профессор, д.т.н. Рабинович Е. В.

Заведующий кафедрой:

доцент, к.т.н. Прохоренко Е. В.
доцент, к.т.н. Якименко А. А.

Ответственный за образовательную программу:

профессор Гужов В. И.

1. Внешние требования

Таблица 1.1

Компетенция ФГОС: ПК.17 способность использовать технологии разработки объектов профессиональной деятельности в различных областях промышленности; в части следующих результатов обучения:
з1. Знать методологии разработки объектов профессиональной деятельности
Компетенция ФГОС: ПК.20 способность проводить оценку производственных и непроизводственных затрат на обеспечение качества объекта проектирования; в части следующих результатов обучения:
з2. Знать правила проведения оценки производственных затрат
у2. Уметь проводить оценку затрат на обеспечение качества
Компетенция ФГОС: ПК.6 способность оценивать надежность и качество функционирования объекта проектирования; в части следующих результатов обучения:
у3. Уметь оценивать надежность объекта проектирования

2. Требования НГТУ к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий
<i>Теория информации</i>	
ПК.6.у3 Уметь оценивать надежность объекта проектирования	
1. уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств	Лекции; Лабораторные работы; Самостоятельная работа
ПК.17.з1 Знать методологии разработки объектов профессиональной деятельности	
2. о понятии "информация" и ее свойствах	Лекции; Самостоятельная работа
3. знать сущность и значение информации в развитии современного общества, опасности и угроз, возникающие в этом процессе	Лекции; Самостоятельная работа
4. методы кодирования и декодирования сообщений	Лекции; Лабораторные работы; Самостоятельная работа
5. количественные оценки информации	Лекции; Самостоятельная работа
6. уметь использовать языки и системы программирования для решения профессиональных задач	Лекции; Самостоятельная работа
7. уметь пользоваться наиболее распространенными офисными и математическими пакетами прикладных программ	Лекции; Лабораторные работы; Самостоятельная работа
ПК.20.з2 Знать правила проведения оценки производственных затрат	
8. Знать правила проведения оценки производственных затрат	Лабораторные работы; Самостоятельная работа
ПК.20.у2 Уметь проводить оценку затрат на обеспечение качества	
9. Уметь проводить оценку затрат на обеспечение качества	Лабораторные работы; Самостоятельная работа

3. Содержание и структура учебной дисциплины

Таблица 3.1

Темы лекций	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 4				
Дидактическая единица: ВВЕДЕНИЕ В ТЕОРИЮ ИНФОРМАЦИИ				

1. 1.1. Что такое информация, данные, знания 1.2. Модель системы передачи и хранения информации 1.3. Задачи и постулаты прикладной теории информации	0	4	2, 3, 6, 7	Лекция проводится в форме расширенного обсуждения
Дидактическая единица: ВВЕДЕНИЕ В ТЕОРИЮ СИГНАЛОВ				
2. 2.1. Общие сведения и понятия 2.2. Типы сигналов 2.3. Модели сигналов 2.3.1. Детерминированные сигналы 2.3.2. Классификация случайных сигналов	0	4	1, 2, 3	Лекция проводится в форме расширенного обсуждения
Дидактическая единица: ВРЕМЕННОЕ И ЧАСТОТНОЕ ПРЕДСТАВЛЕНИЕ СИГНАЛОВ				
3. 3.1. Временное представление сигналов 3.2. Частотное представление сигналов 3.3. Ряд и Интеграл Фурье	0	4	2, 3, 6, 7	Лекция проводится в форме расширенного обсуждения
Дидактическая единица: ПРЕОБРАЗОВАНИЕ ТИПОВ СИГНАЛОВ				
4. 4.1. Дискретизация сигналов 4.1.1. Дискретизация одномерных сигналов 4.1.2. Дискретизация двумерных сигналов (изображений) 4.2. Воспроизведение сигналов 4.3. Квантование сигналов 4.3.1. Скалярное квантование 4.3.2. Векторное квантование и кодовые книги 4.4. Децимация сигналов 4.5. Цифро-аналоговое преобразование сигналов	0	4	1, 2, 3, 6, 7	Лекция проводится в форме расширенного обсуждения
Дидактическая единица: КОЛИЧЕСТВЕННАЯ ОЦЕНКА ИНФОРМАЦИИ				
5. 5.1. Количество информации, энтропия источника сообщений 5.2. Энтропия сложных сообщений, избыточность источника	0	4	1, 2, 5	Лекция проводится в форме расширенного обсуждения
Дидактическая единица: КОДИРОВАНИЕ СООБЩЕНИЙ ПРИ ОТСУТСТВИИ ПОМЕХ				
6. 6.1. Основы экономного кодирования 6.2. Коды без памяти. Код Шеннона-Фано 6.3. Коды с памятью	0	4	1, 4, 7	Лекция проводится в форме расширенного обсуждения

7. 6.4. Методы сжатия с потерей информации 6.5. Методы сжатия видео 6.6. Методы сжатия речевых сигналов	0	4	1, 4, 7	Лекция проводится в форме расширенного обсуждения
Дидактическая единица: КОДИРОВАНИЕ СООБЩЕНИЙ ПРИ НАЛИЧИИ ПОМЕХ				
8. 7.1. Основы помехоустойчивого кодирования 7.1.1. Основные принципы. Типы кодов 7.1.2. Линейные блочные коды 7.1.3. Полиномиальные коды 7.1.4. Теоремы Шеннона для канала с шумами	0	4	1, 4, 7	Лекция проводится в форме расширенного обсуждения
9. 7.2. Сверточные коды 7.3. Применение корректирующего кодирования в системах связи	0	4	1, 4, 7	Лекция проводится в форме расширенного обсуждения

Таблица 3.2

Темы лабораторных работ	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 4				
Дидактическая единица: КОДИРОВАНИЕ СООБЩЕНИЙ ПРИ ОТСУТСТВИИ ПОМЕХ				
1. Алгоритм LZW	4	8	1, 4, 7	Приобретение практических навыков работы с прикладными программами. Выработка умения пользоваться научно-технической литературой в виде учебников, учебных пособий, материалов в электронном виде. Приобретение практических навыков грамотного оформления отчетной документации.
2. Алгоритм Хаффмана	4	8	1, 4, 7	Приобретение практических навыков работы с прикладными программами. Выработка умения пользоваться научно-технической литературой в виде учебников, учебных пособий, материалов в электронном виде. Приобретение практических навыков грамотного оформления отчетной документации.
Дидактическая единица: КОДИРОВАНИЕ СООБЩЕНИЙ ПРИ НАЛИЧИИ ПОМЕХ				

3. Коды Хемминга	4	8	1, 4, 7, 8, 9	Приобретение практических навыков работы с прикладными программами. Выработка умения пользоваться научно-технической литературой в виде учебников, учебных пособий, материалов в электронном виде. Приобретение практических навыков грамотного оформления отчетной документации.
4. Кодирование источника сообщений и канала связи	0	12	1, 4, 7, 8, 9	Приобретение практических навыков работы с прикладными программами. Выработка умения пользоваться научно-технической литературой в виде учебников, учебных пособий, материалов в электронном виде. Приобретение практических навыков грамотного оформления отчетной документации.

4. Самостоятельная работа обучающегося

№	Виды самостоятельной работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
Семестр: 4				
1	РГЗ	1, 2	18	7
: Теория обработки информации : методические указания к лабораторным работам для студентов 3 и 4 курсов факультета автоматики и вычислительной техники, обучающихся по направлению 220200 "Автоматизация и управление", по специальности 220203 "Автономные информационные и управляющие системы" и по специальности 090105 "Комплексное обеспечение информационной безопасности автоматизированных систем" / [сост. В. П. Ющенко] ; Новосиб. гос. техн. ун-т. - Новосибирск, 2006. - 37, [2] с.. - Режим доступа: http://www.library.nstu.ru/fulltext/metodics/2006/06_Jushenko.rar				
2	Подготовка к занятиям	1, 2, 3, 4, 5, 6, 7, 8, 9	18	0
: Теория обработки информации : методические указания к лабораторным работам для студентов 3 и 4 курсов факультета автоматики и вычислительной техники, обучающихся по направлению 220200 "Автоматизация и управление", по специальности 220203 "Автономные информационные и управляющие системы" и по специальности 090105 "Комплексное обеспечение информационной безопасности автоматизированных систем" / [сост. В. П. Ющенко] ; Новосиб. гос. техн. ун-т. - Новосибирск, 2006. - 37, [2] с.. - Режим доступа: http://www.library.nstu.ru/fulltext/metodics/2006/06_Jushenko.rar				
3	Подготовка к аттестации	2, 3	27	0

: Теория обработки информации : методические указания к лабораторным работам для студентов 3 и 4 курсов факультета автоматики и вычислительной техники, обучающихся по направлению 220200 "Автоматизация и управление", по специальности 220203 "Автономные информационные и управляющие системы" и по специальности 090105 "Комплексное обеспечение информационной безопасности автоматизированных систем" / [сост. В. П. Ющенко] ; Новосиб. гос. техн. ун-т. - Новосибирск, 2006. - 37, [2] с. - Режим доступа: http://www.library.nstu.ru/fulltext/metodics/2006/06_Jushenko.rar

5. Технология обучения

Для организации и контроля самостоятельной работы обучающихся, а также проведения консультаций применяются информационно-коммуникационные технологии (табл. 5.1).

Таблица 5.1

Деятельность	Информационно-коммуникационные технологии
Информирование	e-mail; Среда электронного обучения НГТУ
Консультирование	Среда электронного обучения НГТУ
Контроль	e-mail; Среда электронного обучения НГТУ
Размещение учебных материалов	Среда электронного обучения НГТУ

6. Правила аттестации обучающихся по учебной дисциплине

Для аттестации обучающихся по дисциплине используется балльно-рейтинговая система (БРС), позволяющая выставять оценки по традиционной шкале и 15-уровневой ECTS.

Краткая информация о БРС приведена в табл. 6.1.

Таблица 6.1

Оцениваемые виды деятельности обучающихся	Мин. балл	Максимальный балл
Семестр: 4		
<i>Лекция:</i>	8	15
<i>Лабораторная:</i>	20	40
Контролирующие материалы приводятся в "Жуков А. Б. Кодирование и передача информации [Электронный ресурс] : электронный учебно-методический комплекс / А. Б. Жуков ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://courses.edu.nstu.ru/index.php?show=155&курс=736. - Загл. с экрана."		
<i>РГЗ:</i>	12	25
<i>Зачет:</i>	10	20

В таблице 6.2 представлено соответствие форм контроля заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.2

Коды компетенций ФГОС	Результаты обучения	Формы контроля		
		Защита ЛР	Защита РГЗ	Зачет
ПК.17	з1. Знать методологии разработки объектов профессиональной деятельности	+	+	+
ПК.20	з2. Знать правила проведения оценки производственных затрат			+
	у2. Уметь проводить оценку затрат на обеспечение качества			+
ПК.6	у3. Уметь оценивать надежность объекта проектирования	+	+	+

Фонд оценочных средств по дисциплине представлен в приложении № 1 к рабочей программе.

7. Литература

Основная литература

1. Гулятьева Т. А. Основы теории информации и криптографии : конспект лекций / Т. А. Гулятьева; Новосиб. гос. техн. ун-т. - Новосибирск, 2010. - 86, [1] с. : ил. - Режим доступа: <http://www.ciu.nstu.ru/fulltext/textbooks/2010/gulyaeva.pdf>
2. Вернер М. Основы кодирования : Учебник для вузов: Пер. с нем. / М. Вернер. - М., 2004. - 286 с. : ил.
3. Жуков А. Б. Кодирование и передача информации [Электронный ресурс] : электронный учебно-методический комплекс / А. Б. Жуков ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: <http://courses.edu.nstu.ru/index.php?show=155&curs=736>. - Загл. с экрана.
4. Золотарев В. В. Помехоустойчивое кодирование. Методы и алгоритмы : справочник / В. В. Золотарев, Г. В. Овечкин. - М., 2004. - 123, [1] с. : ил. - Рез. англ..
5. Рабинович Е. В. Теория информационных процессов и систем [Электронный ресурс] : учебно-методический комплекс / Е. В. Рабинович ; Новосиб. гос. техн. ун-т. - Новосибирск, [2012]. - Режим доступа: <ftp://tkvt.cs.nstu.ru/teacher/EVR/IS>. - Загл. с экрана. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000164478

Дополнительная литература

1. Вернер М. Основы кодирования : [учебник для вузов по направлению "Прикладные математика и физика"] / М. Вернер ; пер. с нем. Д. К. Зигангирова. - М., 2006. - 286 с. : ил., схемы, табл.
2. Вентцель Е. С. Теория вероятностей и ее инженерные приложения : [учебное пособие для вузов] / Е. С. Вентцель, Л. А. Овчаров. - М., 2007. - 490, [1] с. : ил.
3. Кудряшов Б. Д. Теория информации / Б. Д. Кудряшов. – СПб. : СПбГУ ИТМО, 2010. – 188 с. – (Учебник для вузов).
4. Теория информации и кодирование : [учебное пособие для вузов] / Б. Б. Самсонов [и др.]. - Ростов-на-Дону, 2002. - 287 с. : ил.
5. Дмитриев В. И. Прикладная теория информации : учебник для вузов по специальности "Автоматизированные системы обработки информации и управления" / В. И. Дмитриев. - М., 1989. - 319, [1] с. : ил.
6. Орлов В. А. Теория информации в упражнениях и задачах : [учебное пособие для вузов] / В. А. Орлов, Л. И. Филиппов. - М., 1976. - 136 с. : ил.
7. Игнатов В. А. Теория информации и передачи сигналов : учебник для вузов гражданской авиации / В. А. Игнатов. - М., 1991. - 280 с. : ил.
8. Бояринов И. М. Помехоустойчивое кодирование числовой информации / И. М. Бояринов. - М., 1983. - 193, [2] с.
9. Гридин В. Н. Мажоритарное уплотнение и кодирование двоичных сигналов / В. Н. Гридин, Р. Б. Мазепа, Б. В. Рошин. - М., 2001. - 106 с.
10. Кларк Д. Кодирование с исправлением ошибок в системах цифровой связи / Дж. Кларк, Дж. Кейн ; пер. с англ. С. И. Гельфанда ; под ред. Б. С. Цыбакова. - М., 1987. - 391 с. : ил.

Интернет-ресурсы

1. ЭБС НГТУ : <http://elibrary.nstu.ru/>
2. ЭБС «Издательство Лань» : <https://e.lanbook.com/>
3. ЭБС IPRbooks : <http://www.iprbookshop.ru/>
4. ЭБС "Znaniy.com" : <http://znaniy.com/>

5. :

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

1. Теория обработки информации : методические указания к лабораторным работам для студентов 3 и 4 курсов факультета автоматики и вычислительной техники, обучающихся по направлению 220200 "Автоматизация и управление", по специальности 220203 "Автономные информационные и управляющие системы" и по специальности 090105 "Комплексное обеспечение информационной безопасности автоматизированных систем" / [сост. В. П. Ющенко] ; Новосиб. гос. техн. ун-т. - Новосибирск, 2006. - 37, [2] с. - Режим доступа: http://www.library.nstu.ru/fulltext/metodics/2006/06_Jushenko.rar

2. Паршукова Г. Б. Теория массовой информации [Электронный ресурс] : электронный учебно-методический комплекс / Г. Б. Паршукова, П. Н. Паршукова ; Новосиб. гос. техн. ун-т. - Новосибирск, [2016]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000228722. - Загл. с экрана.

8.2 Специализированное программное обеспечение

1 Microsoft Office

2 Microsoft Windows

9. Материально-техническое обеспечение

Презентационное оборудование

№	Наименование	Назначение
1	Презентационное оборудование (мультимедиа-проектор, экран, компьютер для управления)	Лекционные занятия

ЛАБОРАТОРНАЯ РАБОТА № 1

АЛГОРИТМ КОДИРОВАНИЯ LZW

Цель работы

Познакомиться со словарно-ориентированным алгоритмом сжатия данных Лемпела - Зива - Уолша. Освоить кодирование и декодирование простых сообщений при помощи алгоритма LZW.

Введение

Информация играет все более и более важную роль в современной жизни. Ее объемы постоянно возрастают и требуются все большие хранилища, все более быстрые каналы связи для передачи. Но безграничное повышение емкости хранилищ и скорости линий передачи либо невозможно технически, либо не оправдано экономически. Приходится подстраиваться под существующие возможности. Уменьшать объем информации нежелательно и приходится искать другие способы ее уменьшения. Необходимо уменьшить объем информации, не изменяя ее содержание. Такой процесс называется архивацией или сжатием данных.

Представление информации, которым люди привыкли пользоваться, почти всегда избыточно. Избыточность присутствует в текстах, так как в них обязательно имеются повторяющиеся слова, фразы, а то и целые абзацы. Избыточность информации присуща речи, так как в ней обязательно есть звуки, не слышимые человеком, или несущественные для восприятия. Удалив избыточность можно уменьшить потребность в емкости памяти и пропускной способности линий передачи, необходимых для хранения и передачи информации не уменьшив при этом содержательную сторону информации, т.е. сохранив возможность восстановления ее к исходному виду.

Сжатие данных можно разделить на два основных типа:

- сжатие без потерь или полностью обратимое;
- сжатие с потерями, когда несущественная часть данных утрачивается и полное восстановление невозможно.

Первый тип сжатия данных применяют для текстов, числовых массивов и т.п., когда важно восстановить данные после сжатия в неискаженном виде. Сжатие достигается только за счет более экономного, представления данных. Степень сжатия без потерь относительно невелика.

Второй тип сжатия данных применяют, в основном, для изображений и звука. За счет потери несущественной части данных может быть достигнута высокая степень сжатия. В этом случае потери при сжатии означают незначительное искажение изображения или звука, которые не препятствуют нормальному восприятию (незаметны), но при детальном сличении оригинала и восстановленной после сжатия копии могут быть обнаружены.

Основные идеи словарных методов кодирования

Практически все словарные методы кодирования принадлежат семье алгоритмов из работ двух израильских ученых - Зива и Лемпела, опубликованной в 1977 году. Сущность их состоит в том, что фразы в сжимаемом тексте заменяются указателем на то место, где они в этом тексте уже ранее появлялись.

Это семейство алгоритмов называется методом Зива-Лемпела и обозначается как LZ-сжатие. Этот метод быстро приспосабливается к структуре текста и кодирует короткие слова, так как они очень часто в нем встречаются. Новые слова и фразы могут также формироваться из частей ранее встреченных слов.

Декодирование сжатого текста осуществляется напрямую - происходит простая замена указателя готовой фразой из словаря, на которую тот указывает. На практике LZ-метод добивается хорошего сжатия, его важным свойством является очень быстрая работа декодера. (Когда мы говорим о тексте, то предполагаем, что кодированию подвергается некоторый вектор данных с конечным алфавитом, и это не обязательно текст в буквальном смысле этого слова.)

Различают две группы словарных методов кодирования LZ.

Методы, принадлежащие к первой группе, находя в кодируемой последовательности цепочки символов исходного алфавита, которые ранее уже встречались, вместо того, чтобы повторять эти цепочки, заменяют их указателями на предыдущие повторения.

Словарь в этой группе алгоритмов в неявном виде содержится в обрабатываемых данных, сохраняются лишь указатели на встречающиеся цепочки повторяющихся символов.

Все методы этой группы базируются на алгоритме, разработанном и опубликованном в 1977 году Абрахамом Лемпелем и Якобом Зивом, - LZ77. Наиболее совершенным представителем этой группы, включившим в себя все достижения, полученные в данном направлении, является алгоритм LZSS, опубликованный в 1982 году Сторером и Шимански.

Алгоритмы второй группы в дополнение к исходному алфавиту источника создают словарь фраз, представляющих собой повторяющиеся комбинации символов исходного алфавита, встречающиеся во входных данных. При этом размер алфавита источника возрастает, и для его кодирования потребуется большее число бит, но значительная часть этого словаря будет представлять собой уже не отдельные буквы, а буквосочетания или целые слова. Когда кодер обнаруживает фразу, которая ранее уже встречалась, он заменяет ее индексом словаря, содержащим эту фразу. При этом длина кода индекса получается меньше или намного меньше длины кода фразы.

Все методы этой группы базируются на алгоритме, разработанном и опубликованном Лемпелем и Зивом в 1978 году, - LZ78. Наиболее совершенным на данный момент представителем этой группы словарных методов является алгоритм LZW, разработанный в 1984 году Терри Вэлчем. Освоению этого алгоритма посвящена данная лабораторная работа.

Работа LZW-кодера и декодера

Алгоритм LZW-сжатия выглядит следующим образом:

ШАГ 1. Инициализация словаря всеми возможными односимвольными фразами (обычно 256 символами расширенного ASCII).

Инициализация выходного кода w первым символом сообщения.

ШАГ 2. Считать очередной символ K из кодируемого сообщения.

ШАГ 3. Если найден конец сообщения

 Выдать код для w

Конец

Если строка wK уже есть в словаре

 Присвоить выходному коду значение wK

 Перейти к Шагу 2

Иначе

 Выдать код w

 Добавить wK в словарь под следующим номером

 Присвоить выходному коду значение K

 Перейти к Шагу 2.

Кодер последовательно считывает символы входного потока (строки) и проверяет, есть ли в уже созданном словаре такая строка. Процесс начинается со строк, состоящих из двух символов.

Если строка имеется в словаре, то считывается следующий символ, а если такой строки нет, - то она заносится в словарь, а в выходной поток заносится код для предыдущей найденной строки и поиск начинается сначала.

Пусть на вход кодера поступает последовательность символов вида «/WED/WE/WEE/WEB». Схема сжатия выглядит следующим образом:

Входная строка wK (в словарь)	Выходной код w	Словарь	Позиции словаря
ASCII+		ASCII+	0 - 255
/W	cod(/)	/W	256
WE	cod(W)	WE	257
ED	cod(E)	ED	258
D/	cod(D)	D/	259
/WE	<256>	/WE	260
E/	cod(E)	E/	261
/WEE	<260>	/WEE	262
E/W	<261>	E/W	263
WEB	<257>	WEB	264
<END>	cod(B)		

В результате получается выходной код «cod(/) cod(W) cod(E) cod(D) <256> cod(E) <260> <261> <257> cod(B)». Здесь cod(*) означает код ASCII символа *.

Схема работы LZW-декодирования выглядит следующим образом:

Входной код строка	Выходная	Словарь	Позиция словаря
ASCII+		ASCII+	0 - 255
cod(/)	/		
cod(W)	W	/W	256
cod(E)	E	WE	257
cod(D)	D	ED	258
<256>	/W	D/	259
cod(E)	E	/WE	260
<260>	/WE	E/	261
<261>	E/	/WEE	262
<257>	WE	E/W	263
cod(B)	B	WEB	264

Замечательным качеством этого способа сжатия является то, что весь словарь новых символов передается декодеру без собственно передачи. В конце процесса декодирования декодер имеет точно такой же словарь новых символов, какой в процессе кодирования был накоплен кодером, при этом его создание было частью процесса декодирования.

Задание на лабораторную работу

1. Изучить алгоритм LZW и порядок действия кодера и декодера.
2. Запустить программный пакет лабораторных работ CryptLab. Выполнить следующие упражнения.
3. Произвести кодирование строки символов при помощи алгоритма LZW. В качестве кодируемого сообщения использовать свою фамилию, имя и отчество. Например, <Иванов Иван Иванович>. Угловые скобки только выделяют текст (не вносятся в строку ввода и не кодируются), пробел является кодируемым символом. Учесть наличие в строке символов прописных и строчных букв. Если в процессе кодирования не произошло сжатия информации, то в кодируемом сообщении необходимо дважды ввести имя. Например, <Иванов Иван Иван Иванович>.
4. Инвертировать кодируемое сообщение (записать буквы в обратном порядке) и провести кодирование. Сравнить вычисленный код с полученным в пункте 3.

5. Произвести декодирование при помощи алгоритма LZW сообщения, соответствующего Вашему варианту. Вариант совпадает с Вашим номером в списке студентов в журнале преподавателя.
6. Открыть окно для создания алфавита. Ввести в строку: <фамилия имя отчество название университета название группы>. Построить минимальный и достаточный алфавит для данного сообщения. Закодировать сообщение.
7. Заменить все гласные буквы на букву «а» (или на любую другую). Построить минимальный и достаточный алфавит для данного сообщения. Закодировать сообщение. Сравнить длины закодированных сообщений.
8. Составить отчёт по выполненной работе.

Пояснения к выполнению лабораторной работы

Панель инструментов основного окна программы CryptLab представлена на Рис. 1.

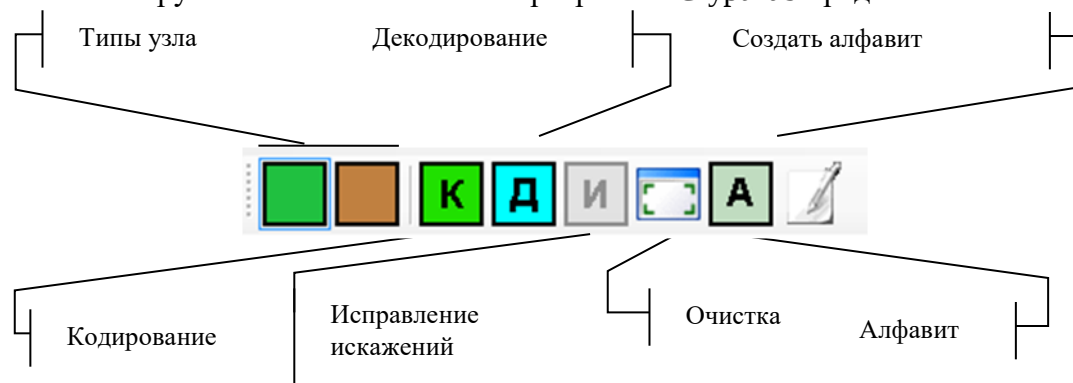


Рис. 1. Панель инструментов

Для выполнения кодирования нажмите кнопку **Кодирование**. Появится новое окно, представленное на рисунке 2. Введите в первую строку заданное кодируемое сообщение. Можно использовать автоматически составляемый в соответствии с введенным сообщением алфавит. Для этого необходимо нажать кнопку **Алфавит**, расположенную в главном окне программы.

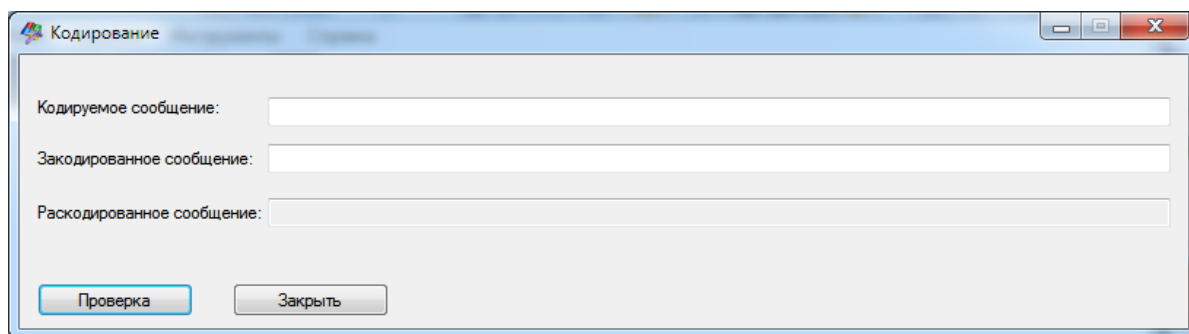


Рис. 2. Кодирование

Для проверки правильности выполнения задания введите полученный Вами код в строку «Закодированное сообщение» и нажмите кнопку «Проверка». Коды символов вводятся через пробел, например, 10 11 12 256. Если кодирование произведено неверно, то будет выведено сообщение об ошибке.

Для выполнения декодирования нажмите кнопку **Декодирование**, размещенную на панели инструментов основного окна программы.

Появится окно, представленное на рисунке 3. Выберите номер своего варианта. В строке «Закодированное сообщение» размещена строка символов, которую Вам надлежит раскодировать. Алфавит, на основании которого закодировано заданное сообщение, можно узнать в окне алфавита.

В строке «Раскодированное сообщение» необходимо ввести полученную в результате декодирования строку символов и нажать кнопку «Проверка».

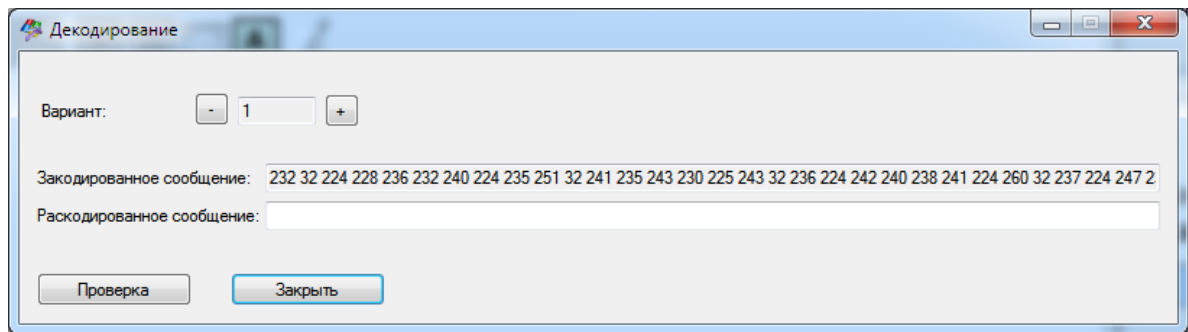


Рис. 3. Декодирование

Если декодирование произведено неверно, то будет выведено сообщение об ошибке.

Для создания алфавита нажмите кнопку **Создать алфавит** основного меню. Появится окно, показанное на рисунке 4.

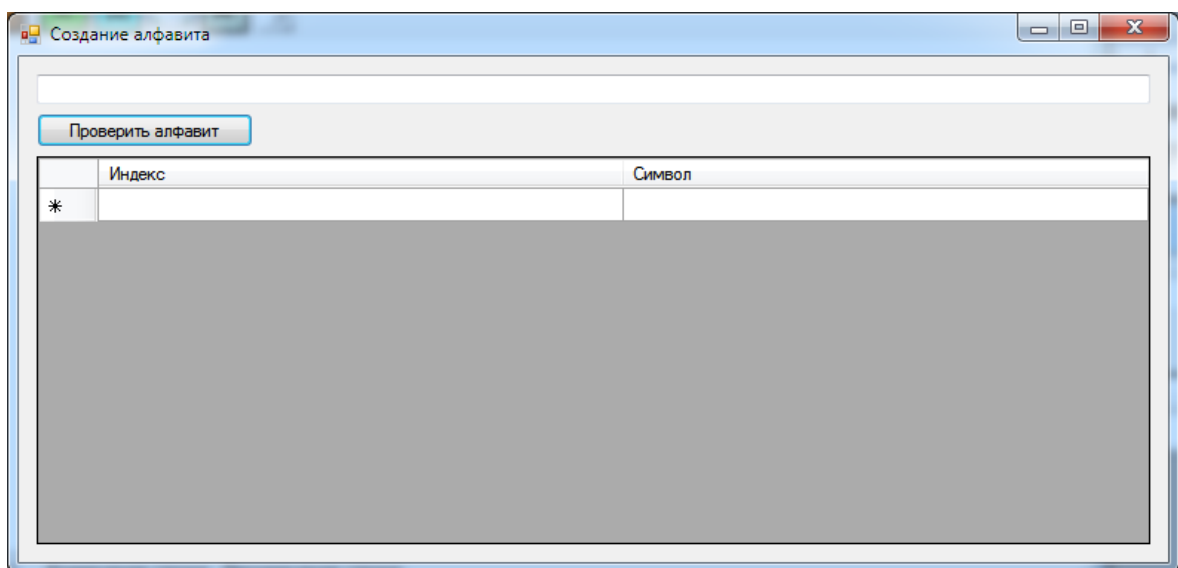


Рис. 4. Окно создания алфавита

В верхнюю строку введите сообщение в соответствии с заданием и впишите символы в нижерасположенную таблицу. Если все символы вписаны правильно, то при нажатии кнопки «Проверить алфавит» будет выведено сообщение о длине закодированного сообщения.

Содержание отчета

1. Титульный лист
2. Результаты выполнения заданий
3. Таблицы генерации алфавитов к каждому заданию
4. Выводы по работе.

Литература к лабораторной работе № 1

1. Ziv J., Lempel A. A universal algorithm for sequential data compression. / IEEE Transactions on Information Theory.- 1977.- Vol. 23.- №3.- pp. 337 – 343.
2. Ziv J., Lempel A. Compression of individual sequences via variable-rate coding. / IEEE Transactions on Information Theory.- 1978.- Vol. 24.- №5.- pp. 530 – 536.
3. Welch T. A. A technique for high-performance data compression. / IEEE Computer.- 1984.- Vol. 17.- №6.- pp. 8 – 19.

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра вычислительной техники
Кафедра систем сбора и обработки данных

“УТВЕРЖДАЮ”
ДЕКАН АВТФ
к.т.н., доцент И.Л. Рева
“ ” _____ Г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УЧЕБНОЙ ДИСЦИПЛИНЫ

Теория информации

Образовательная программа: 09.03.02 Информационные системы и технологии, профиль:
Информационные системы в промышленности и бизнесе

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплине Теория информации приведена в Таблице.

Таблица

Формируемые компетенции	Показатели сформированности компетенций (знания, умения, навыки)	Темы	Этапы оценки компетенций	
			Мероприятия текущего контроля (курсовой проект, РГЗ(Р) и др.)	Промежуточная аттестация (экзамен, зачет)
ПК.17 способность использовать технологии разработки объектов профессиональной деятельности в различных областях промышленности	31. Знать методологии разработки объектов профессиональной деятельности	Алгоритм LZW Алгоритм Хаффмана Кодирование источника сообщений и канала связи Коды Хемминга 1.1. Что такое информация, данные, знания 1.2. Модель системы передачи и хранения информации 1.3. Задачи и постулаты прикладной теории информации 2.1. Общие сведения и понятия 2.2. Типы сигналов 2.3. Модели сигналов 2.3.1. Детерминированные сигналы 2.3.2. Классификация случайных сигналов 3.1. Временное представление сигналов 3.2. Частотное представление сигналов 3.3. Ряд и Интеграл Фурье 4.1. Дискретизация сигналов 4.1.1. Дискретизация одномерных сигналов 4.1.2. Дискретизация двумерных сигналов (изображений) 4.2. Воспроизведение сигналов 4.3. Квантование сигналов 4.3.1. Скалярное квантование 4.3.2. Векторное квантование и кодовые книги 4.4. Децимация сигналов 4.5. Цифро-аналоговое преобразование сигналов 5.1. Количество информации, энтропия источника сообщений 5.2. Энтропия сложных сообщений, избыточность источника 6.1. Основы экономного кодирования 6.2. Коды без памяти. Код Шеннона-Фано 6.3. Коды с памятью 6.4. Методы сжатия с потерей информации 6.5. Методы сжатия видео 6.6. Методы сжатия речевых сигналов 7.1. Основы помехоустойчивого кодирования 7.1.1. Основные принципы. Типы кодов 7.1.2. Линейные блочные коды 7.1.3. Полиномиальные коды 7.1.4. Теоремы Шеннона для канала с шумами 7.2. Сверточные	Отчет по лабораторной работе, РГЗ	Зачет, вопросы 1-9

		коды 7.3. Применение корректирующего кодирования в системах связи		
ПК.20 способность проводить оценку производственных и непроизводственных затрат на обеспечение качества объекта проектирования	з2. Знать правила проведения оценки производственных затрат	Кодирование источника сообщений и канала связи Коды Хемминга		Зачет, вопросы 9-12
ПК.20	у2. Уметь проводить оценку затрат на обеспечение качества	Кодирование источника сообщений и канала связи Коды Хемминга		Зачет, вопросы 13
ПК.6 способность оценивать надежность и качество функционирования объекта проектирования	у3. Уметь оценивать надежность объекта проектирования	Алгоритм LZW Алгоритм Хаффмана Кодирование источника сообщений и канала связи Коды Хемминга 2.1. Общие сведения и понятия 2.2. Типы сигналов 2.3. Модели сигналов 2.3.1. Детерминированные сигналы 2.3.2. Классификация случайных сигналов 4.1. Дискретизация сигналов 4.1.1. Дискретизация одномерных сигналов 4.1.2. Дискретизация двумерных сигналов (изображений) 4.2. Воспроизведение сигналов 4.3. Квантование сигналов 4.3.1. Скалярное квантование 4.3.2. Векторное квантование и кодовые книги 4.4. Децимация сигналов 4.5. Цифро-аналоговое преобразование сигналов 5.1. Количество информации, энтропия источника сообщений 5.2. Энтропия сложных сообщений, избыточность источника 6.1. Основы экономного кодирования 6.2. Коды без памяти. Код Шеннона-Фано 6.3. Коды с памятью 6.4. Методы сжатия с потерей информации 6.5. Методы сжатия видео 6.6. Методы сжатия речевых сигналов 7.1. Основы помехоустойчивого кодирования 7.1.1. Основные принципы. Типы кодов 7.1.2. Линейные блочные коды 7.1.3. Полиномиальные коды 7.1.4. Теоремы Шеннона для канала с шумами 7.2. Сверточные коды 7.3. Применение корректирующего кодирования в системах связи	Отчет по лабораторной работе, РГЗ	Зачет, вопросы 14-21

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по дисциплине проводится в 4 семестре - в форме дифференцированного зачета, который направлен на оценку сформированности компетенций ПК.17, ПК.20, ПК.6.

Зачет проводится в устной форме, по билетам.

Кроме того, сформированность компетенций проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 4 семестре обязательным этапом текущей аттестации является расчетно-графическое задание (работа) (РГЗ(Р)). Требования к выполнению РГЗ(Р), состав и правила оценки сформулированы в паспорте РГЗ(Р).

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе учебной дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ПК.17, ПК.20, ПК.6, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы не достаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

ЛАБОРАТОРНАЯ РАБОТА № 2

АЛГОРИТМ КОДИРОВАНИЯ ХАФФМАНА

Цель работы

Познакомиться с алгоритмом сжатия данных Хаффмана. Освоить кодирование и декодирование простых сообщений при помощи алгоритма Хаффмана.

Коды Хаффмана

Кодирование Хаффмана является алгоритмом для построения кодов переменной длины, имеющих минимальную среднюю длину. Начиная с работ Д. Хаффмана 1952 года, этот алгоритм служит основой многих компьютерных программ сжатия текстовой и графической информации. Некоторые из них используют непосредственно алгоритм Хаффмана, другие берут его в качестве одной из ступеней многоуровневого процесса сжатия. Метод Хаффмана производит идеальное сжатие (то есть, сжимает данные источника до их энтропии), если вероятности символов точно равны отрицательным степеням числа 2.

Код Хаффмана относится к простейшим кодам, на основе которых может выполняться сжатие данных, называемым «кодами без памяти». В коде без памяти каждый символ в кодируемом векторе данных заменяется кодовым словом из префиксного множества двоичных последовательностей или слов.

Префиксным множеством двоичных последовательностей S называется конечное множество двоичных последовательностей, таких, что ни одна последовательность в этом множестве не является префиксом, или началом, никакой другой последовательности в S .

К примеру, множество двоичных слов $S_1 = \{00, 01, 100, 110, 1010, 1011\}$ является префиксным множеством двоичных последовательностей, поскольку, если проверить любую из 30 возможных совместных комбинаций $(w_i w_j)$ из S_1 , то видно, что w_i никогда не является префиксом (или началом) w_j . С другой стороны, множество $S_2 = \{00, 001, 1110\}$ не является префиксным множеством двоичных последовательностей, так как последовательность 00 является префиксом (началом) другой последовательности из этого множества - 001.

Так если необходимо закодировать вектор данных $X = (x_1, x_2, \dots, x_n)$ с алфавитом данных A размера k , то кодирование кодом без памяти осуществляется следующим образом:

- составляют полный список символов $a_1, a_2, a_j, \dots, a_k$ алфавита A , в котором a_j - j -й по частоте появления в X символ, то есть первым в списке будет стоять наиболее часто встречающийся в алфавите символ, вторым – реже встречающийся и т.д.;
- каждому символу a_j назначают кодовое слово w_j из префиксного множества двоичных последовательностей S ;
- выход кодера получают объединением в одну последовательность всех полученных двоичных слов.

Алгоритм Хаффмана реализует общую идею статистического кодирования с использованием префиксных множеств и строит так называемое H -дерево следующим образом:

- выписываем в ряд все символы алфавита в порядке возрастания или убывания частоты (вероятности) их появления в тексте. Символы образуют «листья» H -дерева.
- последовательно объединяем два символа с наименьшими частотами в новый составной символ, частота появления которого полагаем равной сумме частот составляющих его символов. В результате объединения получается иерархическая структура вида дерева, каждый узел которого имеет суммарную частоту всех узлов, находящихся ниже него. Узлы соединяются «ветками» H -дерева.
- прослеживаем путь к каждому листу дерева, пометая направление к каждому узлу (например, направо - 1, налево - 0). Полученная последовательность дает кодовое слово, соответствующее каждому символу.

Недостатком кодов Хаффмана является необходимость иметь таблицы частот (вероятностей) для каждого типа сжимаемых данных. Это не представляет проблемы, если известно, что сжимается текст на известном языке, например английском, тогда компрессору и декомпрессору предоставляется подходящее для английского текста дерево. В общем же случае, когда неизвестны частота появления символов для входных данных, статические коды Хаффмана работают недостаточно эффективно.

Динамическая версия сжатия Хаффмана может строить дерево Хаффмана «на лету» во время чтения и активного сжатия. Дерево постоянно обновляется, чтобы отражать изменения вероятностей входных данных. Однако и здесь существуют свои проблемы.

Работа кодера и декодера алгоритма Хаффмана

Построим кодовое дерево для сообщения со следующим алфавитом:

Частота	15	7	6	5	3
Символ	A	B	C	D	E

На первом шаге из листьев дерева выбираются 2 с наименьшими частотами - E и D. Они образуют узел-родитель, частота которого $3 + 5 = 8$. Узел D соответствует ветви 1 родителя, узел E - ветви 0.

На следующем шаге объединяются узлы B и C, так как теперь эта пара имеет самую меньшую частоту в H-дереве. Создается новый узел с частотой 13. После этого H-дерево кодирования выглядит, как показано на Рис. 1.

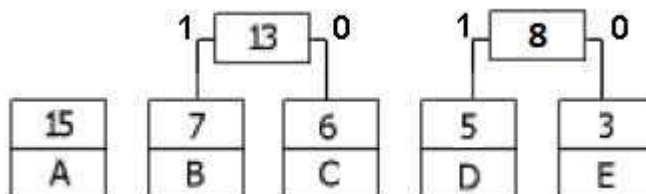


Рис. 1 - Дерево Хаффмана на втором шаге

На следующем шаге «наиредчайшей» парой оказываются узлы BC и DE. Для них еще раз создается родитель, теперь уже с весом 21. Узел BC соответствует ветви 1 родителя, DE - ветви 0. На последнем шаге остается только 2 узла - это A и узел BCDE. В очередной раз создается родитель с весом 36 и бывшие свободные узлы присоединяются к разным его ветвям. Поскольку остается только один узел, то алгоритм построения дерева кодирования Хаффмана завершается. Полное дерево представлено на Рис. 2.

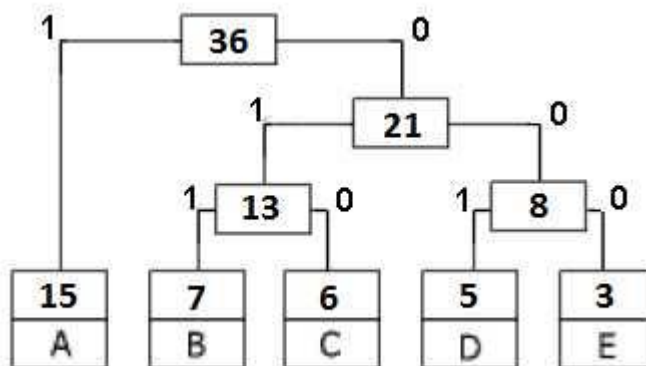


Рис.2 - Завершенное дерево Хаффмана

Чтобы определить код для каждого из символов, входящих в сообщение, необходимо пройти путь от листа дерева, соответствующего этому символу, до корня дерева, накапливая биты при перемещении по ветвям дерева. Полученная таким образом последовательность битов

является кодом данного символа, записанная в обратном порядке. Для данной таблицы символов коды Хаффмана будут выглядеть следующим образом.

Символ	A	B	C	D	E
Код	1	011	010	001	000

Поскольку ни один из полученных кодов не является префиксом другого, они могут быть однозначно декодированы при чтении их из потока. Кроме того, наиболее частый символ сообщения *A* закодирован наименьшим количеством битов, а наиболее редкий символ *E* - наибольшим.

Классический алгоритм Хаффмана требует передачи или хранения таблицы частот, следовательно, длина сжатого сообщения увеличивается на длину этой таблицы.

Кроме того необходимость наличия полной частотной статистики перед началом собственно кодирования требует двух проходов по сообщению: одного для построения модели сообщения (Таблицы частот и *H*-дерева), другого для собственно кодирования.

Задание на лабораторную работу

1. Изучить алгоритм Хаффмана и порядок действия кодера и декодера.
2. Запустить программный пакет лабораторных работ CryptLab. Выполнить следующие упражнения.
3. Собрать в программе *H*-дерево Хаффмана для кодирования заданного сообщения. В качестве сообщения использовать свою фамилию и имя. Например, Иванов Иван. Учесть наличие в строке прописных, строчных букв и пробела.
4. Произвести кодирование заданного сообщения согласно алгоритму Хаффмана. Сравнить размер заданного и закодированного сообщения, оценить степень сжатия.
5. Произвести декодирование при помощи построенного *H*-дерева случайного сообщения.
6. Повторить пункты 3 и 4 для сообщения содержащего повтор Вашей фамилии или имени. Сравнить размер нового закодированного сообщения с размером прежнего, оценить степень сжатия.
7. Составить отчёт по выполненной работе.

Пояснения к выполнению лабораторной работы

Для выполнения Лабораторной работы №2 нужно выбрать пункт меню Задание-> Лабораторная работа №2.

Для выполнения кодирования заданного сообщения согласно алгоритму Хаффмана пользователю предлагается построить *H*-дерево при помощи соединения его листов и узлов.

Листы и узлы представлены на панели инструментов зелеными и коричневыми квадратиками соответственно. Для переноса квадратика в поле программы следует щелкнуть левой кнопкой мыши сначала по соответствующей иконке, а потом по основному полю. Расположение квадратиков можно изменять (потянув мышкой). Операция удаления доступна из контекстного меню объекта.

В контекстном меню листа (см. Рис. 3) вводится символ из заданного сообщения и частота его появления, после чего в соответствующем зеленом квадратике сверху вниз появляются три надписи: частота появления символа, сам символ в кавычках и код символа в круглых скобках.

Для соединения двух квадратиков следует щелкнуть левой кнопкой мыши по свободному порту (маленькому кружку внутри квадратика) одного квадратика, затем – по свободному порту другого. Появится линия соединяющая квадратика - «ветка» *H*-дерева. Если после выбора указателем мыши порта щёлкнуть левой кнопкой по свободной области рабочего поля, линия удаляется. Таким образом, можно удалять уже созданные связи и отменять процесс создания новых соединений.

Узел H -дерева имеет 3 порта: два внизу квадрата (два «сына») и один сверху («родитель»). Лист H -дерева имеет только один порт – «родитель».

Окно свойств листа представлено на Рис. 3. В первую строку вводится кодируемый символ, во вторую – частота его появления в сообщении.

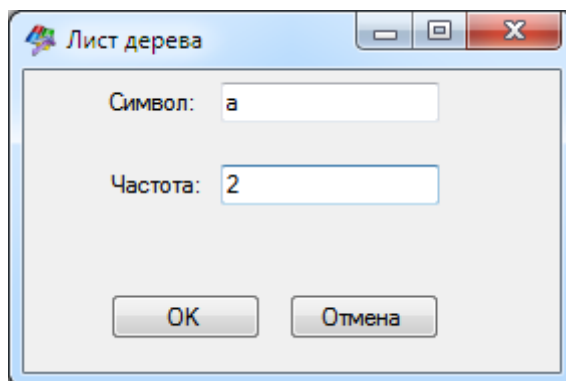


Рис. 3. Свойства объекта

Основные правила построения H -дерева:

- Все узлы должны быть соединены.
- Не допускаются циклы.
- H -дерево должно содержать не менее 2 листьев.
- Не допускается более одного H -дерева.
- Символы разных узлов не могут совпадать.

Внимание! Для устранения неоднозначности построения H -дерева используются правила:

- Листья H -дерева должны располагаться по убыванию частоты слева направо.
- При совпадении частот нескольких узлов или листьев H -дерева упорядочение осуществляется по кодам. За код узла принимается максимальный код его сыновей.

Для проверки правильности сборки H -дерева необходимо нажать кнопку **Кодирование**. Появится окно, представленное на рисунке 4. В первую строку вводим кодируемое сообщение, во вторую закодированное сообщение и нажимаем кнопку «Кодировать». Если H -дерево построено неправильно, то выведется сообщение об ошибке.

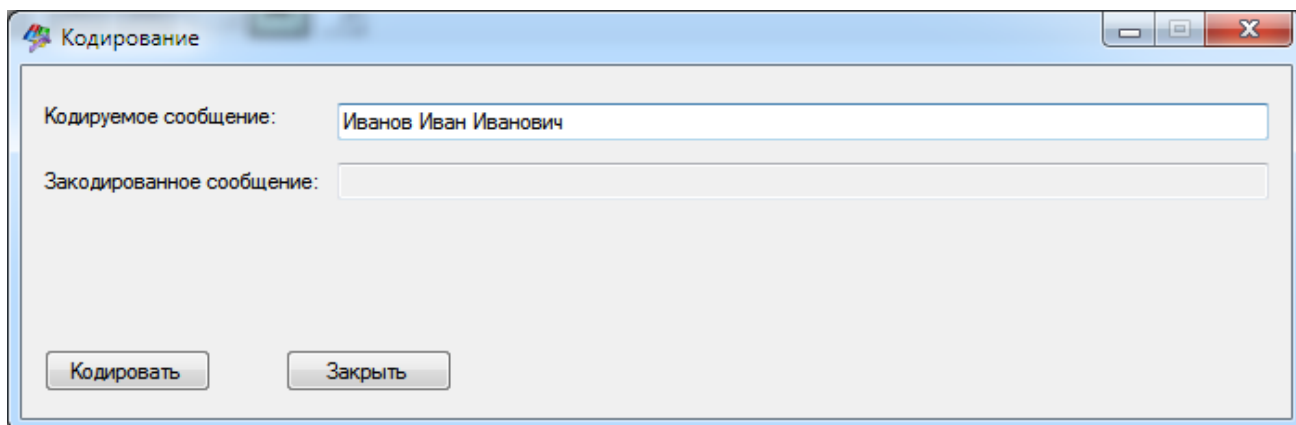


Рис. 4. Кодирование

Для проверки приобретенного навыка декодирования нажмите кнопку **Декодирование** основной панели инструментов. Появится окно, представленное на рисунке 5.

В первую строку введите сообщение, на основании которого будет случайным образом генерироваться строка для декодирования по построенному в пункте 3 задания H -дереву. Нажмите кнопку «Генерация закодированного сообщения», что бы создать эту строку.

Декодируйте с помощью *H*-дерева из пункта 3 задания полученный код и введите декодированное сообщение в третью строку окна. Нажав кнопку «Декодировать» получите сообщение о верном/неверном декодировании.

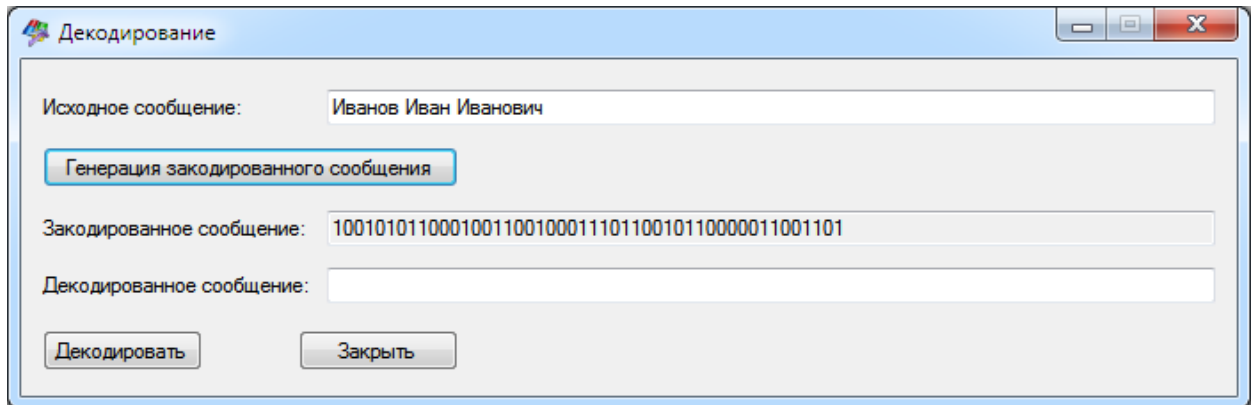


Рис. 5. Декодирование

Содержание отчета

1. Титульный лист
2. Результаты выполнения заданий
3. Выводы по работе.

Пример №1 кодирования и декодирования по алгоритму Хаффмана

Требуется закодировать сообщение «Туркин_Александр».

Выписываем символы сообщения и подсчитываем частоты их появления. В таблицу №1 заносим символы, соответствующие им частоты появления в сообщении и их коды.

Таблица №1

Символ	Т	у	р	к	и	н	_	А	л	е	с	а	д
Код	1058	1091	1088	1082	1080	1085	32	1040	1083	1077	1089	1072	1076
Частота	1	1	2	2	1	2	1	1	1	1	1	1	1

Производим сортировку символов по убыванию частот и кодов. Имеем таблицу №2:

Таблица №2

Символ	р	н	к	у	с	л	и	е	д	а	Т	А	_
Код	1088	1085	1082	1091	1089	1083	1080	1077	1076	1072	1058	1040	32
Частота	2	2	2	1	1	1	1	1	1	1	1	1	1

Согласно таблице №2 создаем 13 листов *H*-дерева. Выбираем два листа с минимальными частотами. Присоединяем их к узлу-родителю, частота появления которого устанавливаются $1 + 1 = 2$. Символы «_» и «А» удаляются из таблицы. Лист «_» соответствует ветви 0 родителя (наименьший располагается справа), лист «А» - ветви 1. В таблицу №3 добавляется новый узел с кодом – 1040 и частотой 2. Обозначим этот узел «А_». Полученная таблица заново сортируется:

Таблица №3

Символ	р	н	к	А_	у	с	л	и	е	д	а	Т
Код	1088	1085	1082	1040	1091	1089	1083	1080	1077	1076	1072	1058
Частота	2	2	2	2	1	1	1	1	1	1	1	1

Снова выбирается два узла, с наименьшими частотами и все операции повторяются до тех пор, пока не останется один узел. Последующие шаги представлены в виде таблицы №4 (красным в таблице выделены новые узлы):

Таблица №4

Шаг 2												
Символ	р	н	к	аТ	А_	у	с	л	и	е	д	
Код	1088	1085	1082	1072	1040	1091	1089	1083	1080	1077	1076	
Частота	2	2	2	2	2	1	1	1	1	1	1	
Шаг 3												
Символ	р	н	к	ед	аТ	А_	у	с	л	и		
Код	1088	1085	1082	1077	1072	1040	1091	1089	1083	1080		
Частота	2	2	2	2	2	2	1	1	1	1		
Шаг 4												
Символ	р	н	ли	к	ед	аТ	А_	у	с			
Код	1088	1085	1083	1082	1077	1072	1040	1091	1089			
Частота	2	2	2	2	2	2	2	1	1			
Шаг 5												
Символ	ус	р	н	ли	к	ед	аТ	А_				
Код	1091	1088	1085	1083	1082	1077	1072	1040				
Частота	2	2	2	2	2	2	2	2				
Шаг 6												
Символ	аТА_	ус	р	н	ли	к	ед					
Код	1072	1091	1088	1085	1083	1082	1077					
Частота	4	2	2	2	2	2	2					
Шаг 7												
Символ	кет	аТА_	ус	р	н	ли						
Код	1082	1072	1091	1088	1085	1083						
Частота	4	4	2	2	2	2						
Шаг 8												
Символ	нли	кет	аТА_	ус	р							
Код	1085	1082	1072	1091	1088							
Частота	4	4	4	2	2							

Шаг 9											
Символ	уср	нли	кет	аТА							
Код	1091	1085	1082	1072							
Частота	4	4	4	4							
Шаг 10											
Символ	кетаТА	уср	нли								
Код	1082	1091	1085								
Частота	8	4	4								
Шаг 11											
Символ	усрнли	кетаТА									
Код	1091	1082									
Частота	8	8									

На 12 шаге остается всего один узел. Обратите внимание, что частота его появления равна 16 – количеству символов в заданной строке.

Дерево Хаффмана, соответствующее заданному сообщению приведено на рисунке 6.

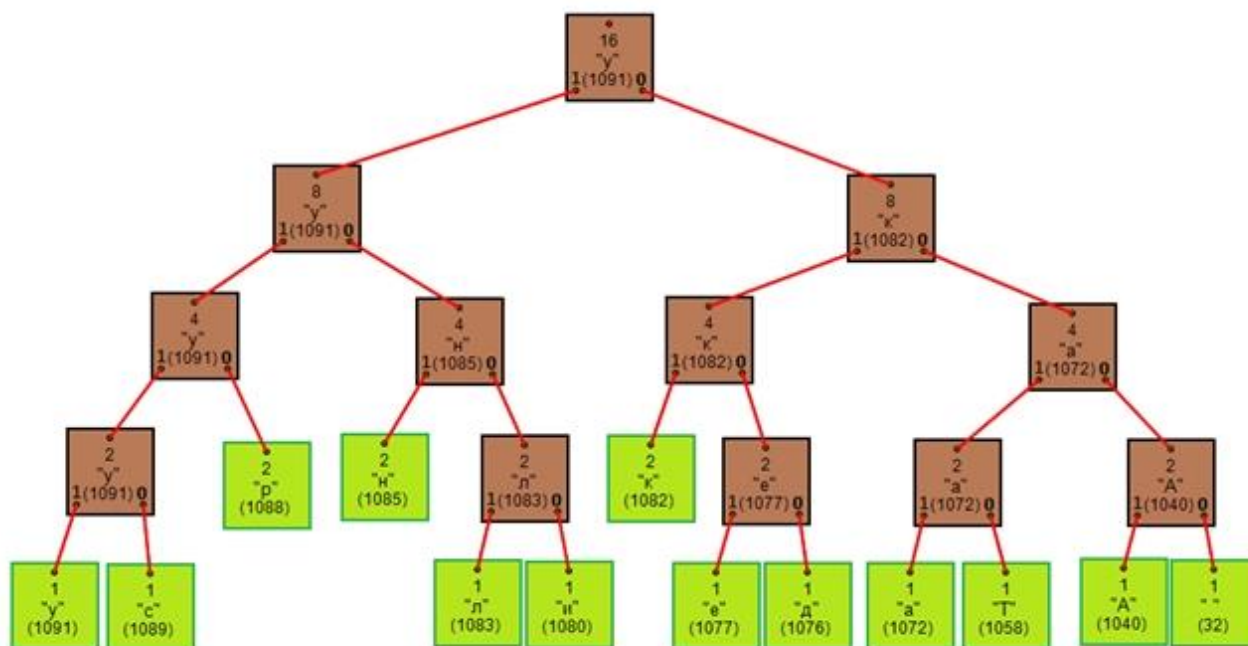


Рис. 6 – Пример №1 дерева Хаффмана.

Таблица №5 кодов Хаффмана символов заданного сообщения выглядит следующим образом.

Таблица №5													
Символ	Т	у	р	к	и	н	_	А	л	е	с	а	д
Код	0010	1111	110	011	1000	101	0000	0001	1001	0101	1110	0011	0100

Поскольку ни один из полученных кодов не является префиксом другого, они могут быть однозначно декодированы при чтении заданного сообщения из потока бит:

0010111111001110001010000 00011001010101111000111010100110.

Пример №2 дерева Хаффмана

Н-дерево для строки «Туркин Александр Александр», состоящей из символов того же алфавита, что и алфавит строки примера №1, изображено на рисунке 7.

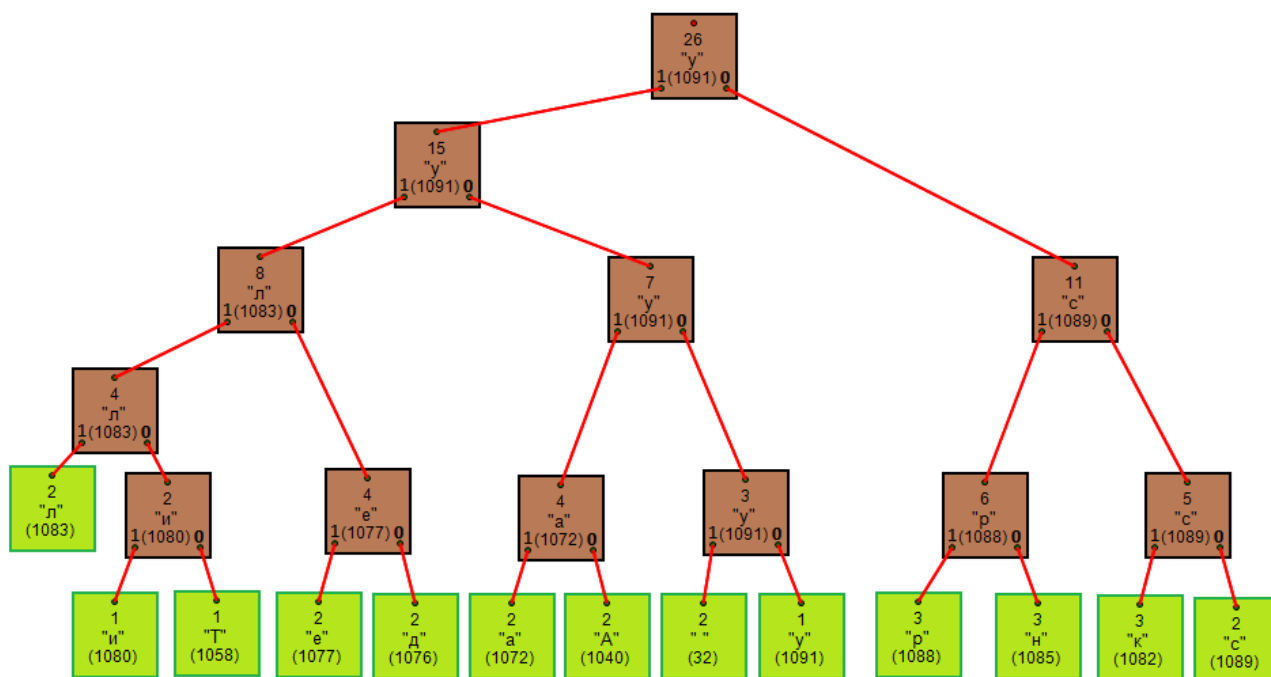


Рис. 7 – Пример №2 дерева Хаффмана.

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра вычислительной техники
Кафедра систем сбора и обработки данных

Паспорт зачета

по дисциплине «Теория информации», 4 семестр

1. Методика оценки

Зачет проводится в устной форме, по билетам. Билет формируется по следующему правилу: первый вопрос выбирается из диапазона вопросов 1-10 второй вопрос из диапазона вопросов 11-21 (список вопросов приведен ниже). В ходе зачета преподаватель вправе задавать студенту дополнительные вопросы из общего перечня (п. 4).

Форма билета для зачета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____

к зачету по дисциплине «Теория информации»

1. Теория информации. Базовые понятия теории информации.
2. Циклические коды.

Утверждаю: зав. кафедрой _____ должность, ФИО
(подпись) (дата)

2. Критерии оценки

- Ответ на билет для зачета считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не способен показать причинно-следственные связи явлений, оценка составляет 0-49 баллов.
- Ответ на билет для зачета засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, может показать причинно-следственные связи явлений, оценка составляет 50-72 баллов.
- Ответ на билет для зачета билет засчитывается на **базовом** уровне, если студент при ответе на вопросы формулирует основные понятия, законы, дает характеристику

процессов, явлений, проводит анализ причин, условий, может представить качественные характеристики процессов, оценка составляет 73-89 баллов.

• Ответ на билет для зачета билет засчитывается на **продвинутом** уровне, если студент при ответе на вопросы проводит сравнительный анализ подходов, проводит комплексный анализ, выявляет проблемы, предлагает механизмы решения, способен представить количественные характеристики определенных процессов, приводит конкретные примеры из практики, оценка составляет 90-100 баллов.

3. Шкала оценки

Зачет считается сданным, если сумма баллов по всем заданиям билета оставляет не менее 51 баллов (из 100 возможных).

В общей оценке по дисциплине баллы за зачет учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Вопросы к зачету по дисциплине «Теория информации»

1. Теория информации. Базовые понятия теории информации.
2. Способы измерения информации. Вероятностный подход к измерению дискретной и непрерывной информации.
3. Понятие энтропии. Энтропия дискретной случайной величины.
4. Понятие префиксного кодирования. Сжатие информации. Основная теорема о кодировании при отсутствии помех.
5. Метод блокирования.
6. Метод Шеннона-Фэно.
7. Арифметическое кодирование.
8. Адаптивные алгоритмы сжатия информации. Адаптивное арифметическое кодирование.
9. Алгоритмы LZ77, LZ78, LZSS, LZW.
10. Особенности программ архиваторов. Непосредственное применение алгоритмов кодирования в архиваторах для обеспечения продуктивной работы в WINDOWS.
11. Сжатие данных с потерями без ущерба для конфиденциальной и важной информации. Основная теорема о кодировании при наличии помех. Помехозащитное кодирование.
12. Коды с исправлением и обнаружением ошибок.
13. Последовательные коды и их применение на практике.
14. Матричное кодирование.
15. Групповой код.
16. Совершенные и квазисовершенные коды. Их свойства.
17. Полиномиальные коды.
18. Циклические коды.
19. Методика построения кодов, минимальное расстояние между кодовыми словами которых равно заданному числу.
20. Коды Боуза-Чоудхури-Хоккенгема.
21. Циклические избыточные коды (CRC).

ЛАБОРАТОРНАЯ РАБОТА № 3

АЛГОРИТМ КОДИРОВАНИЯ ХЭММИНГА

Цель работы

Познакомиться с алгоритмом Хэмминга. Освоить кодирование простых сообщений и поиск искаженных бит при помощи алгоритма Хаффмана.

Коды Хэмминга

Коды Хемминга – это вероятно, первые из самокорректирующихся кодов. Построены они Ричардом Хеммингом в сороковых годах прошлого столетия применительно к двоичной системе счисления.

В то время он работал в лаборатории Белла на электромеханической счетной машине Bell Model V. Ввод данных в эту машину осуществлялся с помощью перфокарт. Это была самая ненадежная часть вычислительной машины. Перфокарты часто считывались неправильно. Исправление и обнаружение ошибок ввода данных отнимало огромное количество времени, а пропущенные ошибки могли привести к неверным результатам работы.

Желая застраховаться от возможных сбоев и ускорить процесс обработки данных, Хемминг много работал над построением эффективных алгоритмов, позволяющих быстро обнаруживать и исправлять ошибки, и в 1950 году опубликовал способ кодирования, являющийся одним из самых известных на сегодняшний день.

Идея кодов Хемминга заключается в разбиении данных на блоки информационных битов фиксированной длины и вводе в эти блоки определенного числа проверочных битов четности. Например, код Хемминга (7,4) чтобы обеспечить исправление однократных ошибок для информационных данных длиной 7 битов требует 4 проверочных бита, а код Хемминга (128,8) для данных длиной 128 битов требует 8 проверочных битов.

Биты полученного блока суммарной длины, порядковые номера которых являются степенью двойки, – это проверочные биты. Если порядковый номер бита обозначить n , то для проверочных битов $n = 2^k$, где k – положительное целое число. Например, для закодированной последовательности длиной 13 битов проверочными будут: 1, 2, 4 и 8 биты.

Каждый проверочный бит, предназначен для проверки определенной группы информационных битов. В проверочном бите записывается сумма по модулю два битов группы, которую он проверяет.

Каждый информационный бит проверяется несколькими проверочными. Для того чтобы определить какими проверочными битами контролируется информационный бит, следует разложить его порядковый номер по степеням двойки. Таким образом, девятый бит будет контролироваться битами 1 и 8, так как $9 = 2^0 + 2^3 = 1 + 8$, а седьмой – битами 1, 2 и 4.

Работа кодера Хэмминга с исправлением однократных ошибок

Рассмотрим пример кодирования бинарной последовательности данных, состоящей из восьми элементов: 10011010 с использованием 4 проверочных битов.

Определим расположение проверочных битов в закодированной последовательности. Обозначим информационные биты символом И, а проверочные биты символом П. Индекс при этих символах означает порядковый номер в закодированной последовательности.

Проверочные биты будут занимать четыре позиции с порядковыми номерами, равными степени двойки: $2^0, 2^1, 2^2, 2^3 \rightarrow 1, 2, 4, 8$.

Размещение информационных и проверочных битов в закодированной последовательности будет следующим:

П ₁	П ₂	И ₃	П ₄	И ₅	И ₆	И ₇	П ₈	И ₉	И ₁₀	И ₁₁	И ₁₂
		1		0	0	1		1	0	1	0

Определим, какие информационные биты, какими проверочными битами проверяются. Для этого разложим номера информационных битов по степеням двойки:

Информационный бит I_3 проверяется проверочными битами P_1 и P_2 : $3 = 2^0 + 2^1 = 1 + 2$.

Информационный бит I_5 проверяется проверочными битами P_1 и P_4 : $5 = 2^0 + 2^2 = 1 + 4$.

Информационный бит I_6 проверяется проверочными битами P_2 и P_4 : $6 = 2^1 + 2^2 = 2 + 4$.

Информационный бит I_7 проверяется проверочными битами P_1 , P_2 и P_4 : $7 = 2^0 + 2^1 + 2^2 = 1 + 2 + 4$.

Информационный бит I_9 проверяется проверочными битами P_1 и P_8 : $9 = 2^0 + 2^3 = 1 + 8$.

Информационный бит I_{10} проверяется проверочными битами P_2 и P_8 : $10 = 2^1 + 2^3 = 2 + 8$.

Информационный бит I_{11} проверяется проверочными битами P_1 , P_2 и P_8 : $11 = 2^0 + 2^1 + 2^3 = 1 + 2 + 8$.

Информационный бит I_{12} проверяется проверочными битами P_4 и P_8 : $12 = 2^2 + 2^3 = 4 + 8$.

Определим значения проверочных битов. Для этого просуммируем по модулю два значения информационных битов, проверяемых соответствующим проверочным битом. Результат запишем в проверочный бит. Получим:

$$P_1 = I_3 + I_5 + I_7 + I_9 + I_{11} = 1 + 0 + 1 + 1 + 1 = 0.$$

$$P_2 = I_3 + I_6 + I_7 + I_{10} + I_{11} = 1 + 0 + 1 + 0 + 1 = 1.$$

$$P_4 = I_5 + I_6 + I_7 + I_{12} = 0 + 0 + 1 + 0 = 1.$$

$$P_8 = I_9 + I_{10} + I_{11} + I_{12} = 1 + 0 + 1 + 0 = 0.$$

Закодированная последовательность имеет следующий вид:

P_1	P_2	I_3	P_4	I_5	I_6	I_7	P_8	I_9	I_{10}	I_{11}	I_{12}
0	1	1	1	0	0	1	0	1	0	1	0

Предположим что при передаче был искажен бит I_7 . Закодированная последовательность битов стала иметь следующий вид:

P_1	P_2	I_3	P_4	I_5	I_6	I_7	P_8	I_9	I_{10}	I_{11}	I_{12}
0	1	1	1	0	0	0	0	1	0	1	0

Проверим на четность все группы битов, контролируемые проверочными битами. Для этого просуммируем по модулю 2 значение каждого проверочного бита со значениями информационных битов проверяемых им. Проверка на четность считается пройденной, если сумма по модулю 2 равна 0.

$$Pr_0: P_1 + I_3 + I_5 + I_7 + I_9 + I_{11} = 0 + 1 + 0 + 0 + 1 + 1 = 1 \neq 0.$$

$$Pr_1: P_2 + I_3 + I_6 + I_7 + I_{10} + I_{11} = 1 + 1 + 0 + 0 + 0 + 1 = 1 \neq 0.$$

$$Pr_2: P_4 + I_5 + I_6 + I_7 + I_{12} = 1 + 0 + 0 + 0 + 0 = 1 \neq 0.$$

$$Pr_3: P_8 + I_9 + I_{10} + I_{11} + I_{12} = 0 + 1 + 0 + 1 + 0 = 0 \equiv 0.$$

Проверка четности не прошла для проверочных битов P_1 , P_2 и P_3 . Ошибка содержится в бите, номер которого равен $n = 2^0 \times Pr_0 + 2^1 \times Pr_1 + 2^2 \times Pr_2 + 2^3 \times Pr_3 = 1 + 2 \times 1 + 4 \times 1 + 8 \times 0 = 7$.

Если предположение, что могла произойти только одна ошибка, неверно, то попытка исправить закодированные данные только привнесет дополнительные искажения в сообщение.

Итак, ошибка произошла в бите I_7 и, инвертировав его значение, можно восстановить переданные данные.

Задание на лабораторную работу

1. Изучить алгоритм Хэмминга и порядок действия кодера и поиска искажений.
2. Запустить программный пакет лабораторных работ CryptLab. Выполнить следующие упражнения.
3. Произвести кодирование сообщения согласно алгоритму Хэмминга. В качестве заданного сообщения использовать свою фамилию и имя. Например, Иванов Иван. Учесть наличие в строке прописных, строчных букв и пробела.

4. Произвести поиск искажений битов в заданном сообщении. В качестве заданного сообщения использовать свое имя и фамилию. Например, Иван Иванов. Вариант совпадает с Вашим номером в списке студентов в журнале преподавателя.

5. Составить отчет по выполненной работе.

Пояснения к выполнению лабораторной работы

Для выполнения Лабораторной работы №3 нужно выбрать пункт меню Задание-> Лабораторная работа №3.

Для выполнения 3-го пункта задания нажмите кнопку **Кодирование** (Рис. 1), размещенную на панели инструментов главного окна программы.

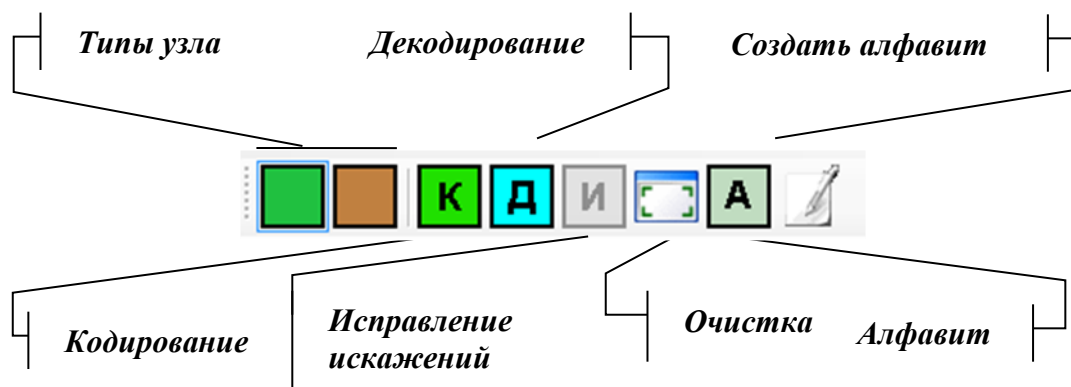


Рис. 1. Панель инструментов

Появится окно, представленное на рисунке 2. В первую строку следует ввести кодируемое сообщение в соответствии с Заданием и нажать кнопку «Показать код». Во второй строке выведется соответствующая битовая последовательность, полученная из таблицы ASCII.

По этой последовательности бит Вам следует произвести кодирование сообщения на основании кода Хэмминга (8,4). В третью строку следует ввести полученное закодированное сообщение (символы вводятся без пробелов, например, 100100010111).

Для проверки правильности кодирования нужно нажать кнопку «Кодировать». Если кодирование произведено неверно, то будет выведено сообщение об ошибке.

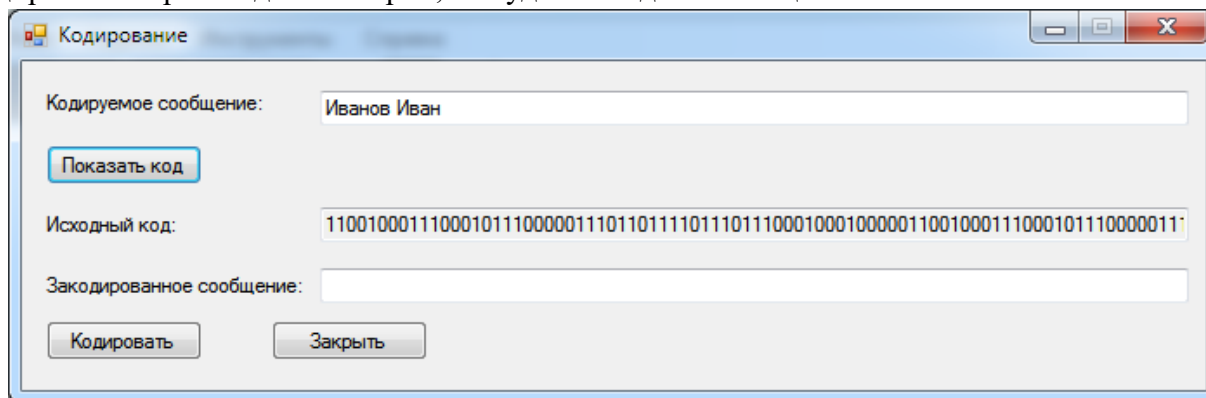


Рис. 2. Кодирование

Для выполнения 4-го пункта задания нажмите кнопку **Исправление искажений** (Рис. 1), размещенную на панели инструментов главного окна программы.

Появится окно, представленное на рисунке 3. Укажите номер своего варианта задания. В первую строку введите заданное сообщение. После этого нажмите кнопку «Генерация». Во второй строке будет выведено закодированное 12-битными байтами на символ заданное сообщение, содержащее ошибки.

Найдите искаженные биты и введите их список в третью строку окна.

Запись искаженного бита состоит из номера байта и номера бита в нем. Записи списка отделяются пробелами, номер байта и бита - двоеточием. Например, если в 2 байте искажен 1 бит, а в 5 искажен - 11, то нужно ввести строку «2:1 5:11».

Нумерация байтов и битов ведется с 1. Максимальный номер бита 12 (8 информационных и 4 проверочных битов).

Записав список искаженных битов, нажмите кнопку «Декодировать». Если исправление последовательности битов произведено неверно, то будет выведено сообщение об ошибке.

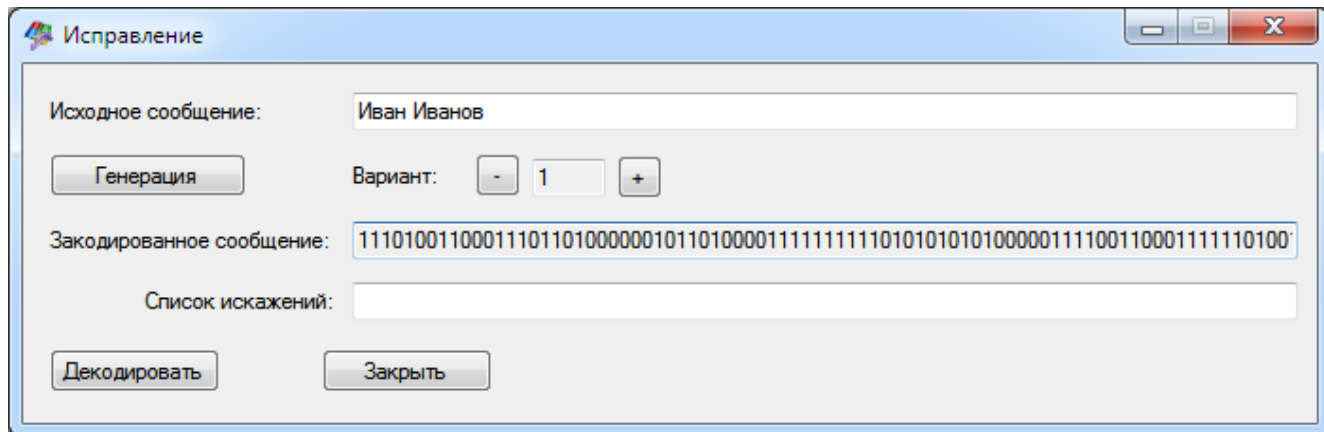


Рис. 3. Исправление искажений последовательности бит

Содержание отчета

1. Титульный лист
2. Результаты выполнения заданий
3. Выводы по работе.

Паспорт расчетно-графического задания (работы)

по дисциплине «Теория информации», 4 семестр

1. Методика оценки

В рамках расчетно-графического задания (работы) по дисциплине студенты должны реализовать один из способов кодирования.

При выполнении расчетно-графического задания (работы) студенты должны провести анализ объекта, выбрать и обосновать признаки и параметры, разработать алгоритмы.

Обязательные структурные части РГЗ.

1. Оглавление.
2. Введение (обоснование актуальности выбранной темы).
3. Обзор литературы (кратко рассматривают существующие решения, их преимущества и недостатки).
4. Результаты.
5. Заключение (краткое подведение итогов).
6. Список литературы (5-10 источников).

Оцениваемые позиции:

1. Качество проведенного поиска литературы.
2. Оригинальность концепции разработанного метода.
3. Корректность проведенных расчетов.

2. Критерии оценки

- Работа считается **не выполненной**, если выполнены не все части РГЗ(Р), отсутствует анализ объекта, диагностические признаки не обоснованы, оценка составляет 0-11 баллов.
- Работа считается выполненной **на пороговом** уровне, если части РГЗ(Р) выполнены формально: анализ объекта выполнен без декомпозиции, диагностические признаки недостаточно обоснованы, оценка составляет 12-16 баллов.
- Работа считается выполненной **на базовом** уровне, если анализ объекта выполнен в полном объеме, признаки и параметры диагностирования обоснованы, алгоритмы разработаны, но не оптимизированы, оценка составляет 17-21 баллов.
- Работа считается выполненной **на продвинутом** уровне, если анализ объекта выполнен в полном объеме, признаки и параметры диагностирования обоснованы, алгоритмы разработаны и оптимизированы, оценка составляет 22-25 баллов.

3. Шкала оценки

В общей оценке по дисциплине баллы за РГЗ(Р) учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Примерный перечень тем РГЗ(Р)

1. Построить код Хаффмена для поговорки, найти среднюю длину кодовых слов.
2. Найти последовательность на входе кодера.
3. Алфавит состоит из букв A, B, C, D . Определить количество информации, приходящееся на символ сообщения, составленного из такого алфавита. Вероятности появления букв равны
4. Сколько информации получено, если выбрана фишка домино, у которой известно, что $3 < \pi < 4$. Сколько дополнительной информации о числе π содержится в записи
5. Имеются две урны, содержащие по m шаров – w_1 и w_2 белых, b_1 и b_2 черных, r_1 и r_2 красных. Из каждой урны извлекают по одному шару. Исход, какого из этих опытов, следует считать более неопределенным, если

ЛАБОРАТОРНАЯ РАБОТА № 4

АЛГОРИТМЫ КОДИРОВАНИЯ ХАФФМАНА И ХЭММИНГА

Цель работы

Освоить кодирование простых сообщений и поиск искаженных бит при помощи алгоритмов Хаффмана и Хэмминга.

Работа кодера

Требуется закодировать сообщение «Александр». На основании уже известного алгоритма Хаффмана (подробно алгоритм расписан в методическом пособии к Лабораторной работе №2) составляется Н-дерево. Полное дерево, полученное в результате, представлено на Рис. 1.

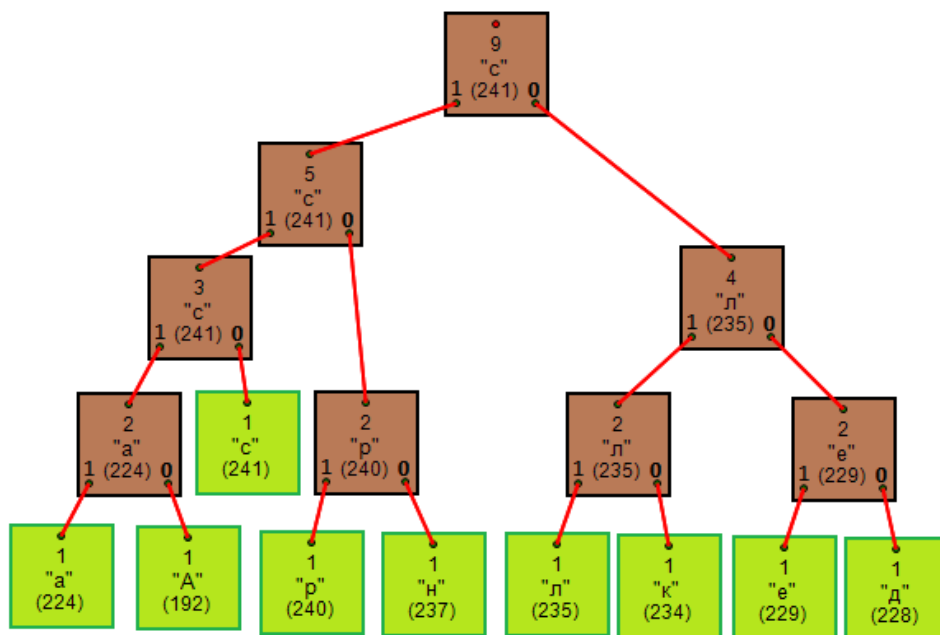


Рис. 1 - Дерево Хаффмана на последнем шаге

Для данной таблицы символов коды Хаффмана будут выглядеть следующим образом.

Символ	А	л	е	к	с	а	н	д	р
Код	1110	011	001	010	110	1111	100	000	101

В результате закодированное сообщение алгоритмом Хаффмана будет иметь вид: 1110011001010110111100000101. Что составляет 29 бит. Для того чтобы закодировать данное сообщение методом Хэмминга, необходимо разделить биты по группам, например, по 8 бит. Как мы видим данное сообщение не кратно 8 битам. Для этого необходимо добавить еще 3 бита, заполнив их нулями. В результате получим сообщение из 32 бит: 1110011001010110111100000101000. В итоге мы получили сообщение кратное 8 битам, но при этом добавленные 3 бита могут быть раскодированы как еще один символ «д». Поэтому в начале сообщения добавим еще 8 бит, в которых будет указано сколько было добавлено бит в конце сообщения. В данном случае это будет «00000011» (3).

В результате сообщение примет вид: «000000111110011001010110111100000101000».

Далее необходимо провести кодирование полученного сообщения с помощью алгоритма Хэмминга. Рассмотрим пример кодирования бинарной последовательности данных, состоящей из восьми элементов, с использованием 4 контрольных бит.

Подробное описание алгоритма Хэмминга представлено в методическом пособии для Лабораторной работы №3. Определим расположение проверочных бит в результирующей закодированной последовательности. Контрольные биты будут занимать четыре позиции с порядковыми номерами, равными степени двойки: $2^0, 2^1, 2^2, 2^3 \rightarrow 1, 2, 4, 8$.

Осталось определить значения проверочных бит. Для этого определим группы для всех контрольных бит, просуммируем их по модулю два, а результат запишем в соответствующие контрольные биты:

$$П_1 = И_3 + И_5 + И_7 + И_9 + И_{11}$$

$$П_2 = И_3 + И_6 + И_7 + И_{10} + И_{11}$$

$$П_4 = И_5 + И_6 + И_7 + И_{12}$$

$$П_8 = И_9 + И_{10} + И_{11} + И_{12}$$

Таким образом, закодированная последовательность будет иметь следующий вид: «11010000001110101100011011001010011001111111000110101011000».

Работа декодера и исправление искажений

Предположим, что при передаче был искажен 7-ой бит первого байта. Таким образом, закодированная последовательность будет иметь следующий вид: «11010010001110101100011011001010011001111111000110101011000».

Проверим на четность все группы битов, контролируемые проверочными битами. Для этого просуммируем по модулю 2 значение каждого проверочного бита со значениями информационных битов проверяемых им. Проверка на четность считается пройденной, если сумма по модулю 2 равна 0. Для первого байта, в котором находится ошибка, это будет выглядеть следующим образом:

$$\text{Пр}_0: П_1 + И_3 + И_5 + И_7 + И_9 + И_{11} = 1 + 0 + 0 + 1 + 0 + 1 = 1 \neq 0.$$

$$\text{Пр}_1: П_2 + И_3 + И_6 + И_7 + И_{10} + И_{11} = 1 + 0 + 0 + 1 + 0 + 1 = 1 \neq 0.$$

$$\text{Пр}_2: П_4 + И_5 + И_6 + И_7 + И_{12} = 1 + 0 + 0 + 1 + 1 = 1 \neq 0.$$

$$\text{Пр}_3: П_8 + И_9 + И_{10} + И_{11} + И_{12} = 0 + 0 + 0 + 1 + 1 = 0 \equiv 0.$$

Проверка четности не прошла для проверочных битов $П_1, П_2$ и $П_3$. Ошибка содержится в бите, номер которого равен $n = 2^0 \times \text{Пр}_0 + 2^1 \times \text{Пр}_1 + 2^2 \times \text{Пр}_2 + 2^3 \times \text{Пр}_3 = 1 + 2 \times 1 + 4 \times 1 + 8 \times 0 = 7$.

Если предположение, что могла произойти только одна ошибка, неверно, то попытка исправить закодированные данные только привнесет дополнительные искажения в сообщение.

Итак, ошибка произошла в бите $И_7$ и, инвертировав его значение, можно восстановить переданные данные. В результате мы получим следующее сообщение без искажений: «11010000001110101100011011001010011001111111000110101011000».

Далее необходимо выписать информационные биты из этого сообщения. Тем самым мы получим сообщение, закодированное с помощью алгоритма Хаффмана: «0000001111100110010101101111100000101000», в котором первый байт указывает сколько бит добавлено на конце сообщения для того, чтобы сообщение было кратно 8 битам. Этот байт равен 00000011 или 3. Следовательно, исходное сообщение было «11100110010101101111100000101», что при известном соотношении кодов и символов легко декодировать.

Задание на лабораторную работу

1. Запустить программный пакет лабораторных работ CryptLab. Выполнить следующие упражнения.
2. Произвести кодирование заданного сообщения согласно алгоритмам Хаффмана и Хэмминга. В качестве сообщения использовать свое имя и отчество. Например, Иван Иванов. Учесть наличие в строке прописных, строчных букв и пробела.
3. Произвести декодирование заданного сообщения, в соответствии с вариантом задания. Вариант совпадает с Вашим номером в списке студентов в журнале преподавателя.
4. Составить отчет по выполненной работе.

Пояснения к выполнению лабораторной работы

Для выполнения Лабораторной работы №4 нужно выбрать пункт меню Задание→Лабораторная работа №4.

Для проверки правильности произведенного кодирования нажмите кнопку *Кодирование* (см. рис. 2), размещенную на панели инструментов главного окна программы.

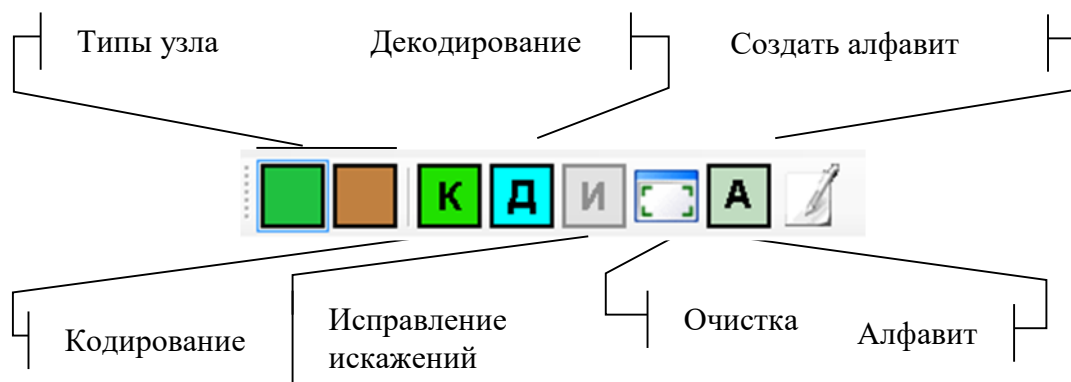


Рис. 2 - Панель инструментов

Появится новое окно, представленное на рисунке 3. В первой строке необходимо ввести кодируемое сообщение в соответствии с заданием лабораторной работы. Во второй строке будет показано битовая последовательность данного сообщения закодированного изначально с помощью алгоритма Хаффмана, а затем алгоритмом Хэмминга (описание есть в методическом пособии). проверки правильности построенного дерева на основании заданного сообщения, необходимо нажать кнопку «Проверить дерево».

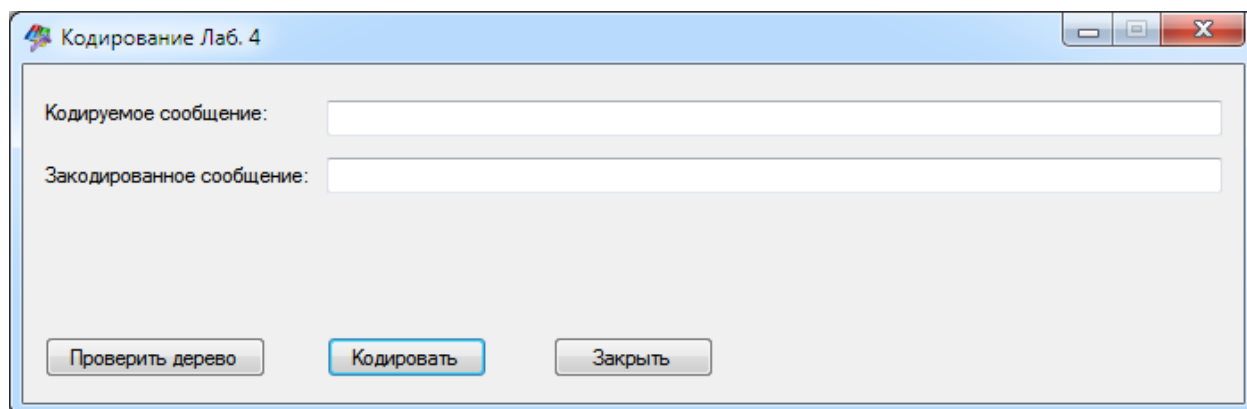


Рис. 3 - Кодирование

Для проверки правильности кодирования нужно нажать кнопку «Кодировать». Если кодирование произведено не верно, то будет выведено соответствующее сообщение об ошибке.

Для проверки правильности произведенного декодирования нажмите кнопку *Декодирование* (см. рис. 2), размещенную на панели инструментов главного окна программы.

Появится новое окно, представленное на рисунке 4. Укажите номер своего варианта задания. В первой строке будет выведено закодированное сообщение, содержащее ошибки. В третью строку необходимо ввести раскодированное сообщение.

Если декодирование произведено не верно, то будет выведено соответствующее сообщение об ошибке.

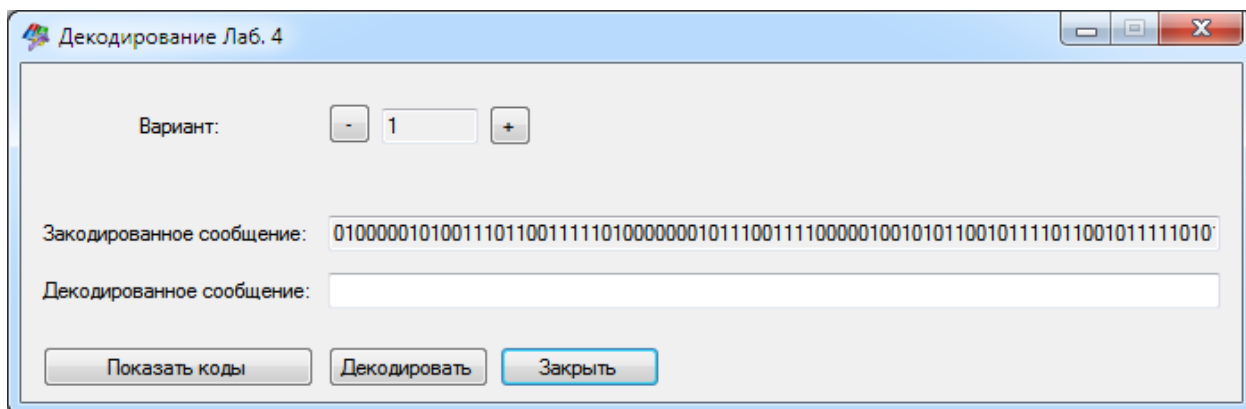


Рис. 4 - Декодирование

Содержание отчета

1. Титульный лист
2. Результаты выполнения заданий
3. Выводы по работе.

Задания для контрольной работы

1. Построить код Хаффмена для поговорки, найти среднюю длину кодовых слов.
2. Найти последовательность на входе кодера.
3. Алфавит состоит из букв A, B, C, D . Определить количество информации, приходящееся на символ сообщения, составленного из такого алфавита. Вероятности появления букв равны
4. Сколько информации получено, если выбрана фишка домино, у которой
5. Известно, что $3 < \pi < 4$. Сколько дополнительной информации о числе π содержится в записи
6. Имеются две урны, содержащие по m шаров – w_1 и w_2 белых, b_1 и b_2 черных, r_1 и r_2 красных. Из каждой урны извлекают по одному шару. Исход, какого из этих опытов, следует считать более неопределенным, если

Вариант 1. В фамилии студента общее число букв четное, число согласных нечетное

- 1.1. Кто сплетничает с Тобой, тот сплетничает о Тебе.
- 1.2. Арифметическое кодирование использовано для кодирования последовательности длины 5 на выходе двоичного постоянного источника с вероятностью единицы 0,4. Кодовое слово на выходе арифметического кодера имеет вид: 101010.
- 1.3. $p(A) = 0,1, p(B) = 0,2, p(C) = 0,3, p(D) = 0,4$.
- 1.4. цифры разные?
- 1.5. $\pi = 3.14159265\dots$?
- 1.6. $m = 10, w_1 = 5, b_1 = 3, r_1 = 2, w_2 = 8, b_2 = 1, r_2 = 1$.

Вариант 2. В фамилии студента общее число букв нечетное, число согласных четное

- 2.1. Ехал Грека через реку, видит Грека в реке рак.
- 2.2. Арифметическое кодирование использовано для кодирования последовательности длины 5 на выходе двоичного постоянного источника с вероятностью единицы 0,2. Кодовое слово на выходе арифметического кодера имеет вид: 01110.
- 2.3. $p(A) = 0,25, p(B) = 0,25, p(C) = 0,25, p(D) = 0,25$.
- 2.4. цифры нечетные?
- 2.5. $\pi = 3.1415926\dots$?
- 2.6. $m = 20, w_1 = 10, b_1 = 5, r_1 = 5, w_2 = 8, b_2 = 8, r_2 = 4$.

Вариант 3. В фамилии студента общее число букв и число согласных нечетное

- 3.1. Сунул Грека руку в реку, рак за руку Греку цап!
- 3.2. Арифметическое кодирование использовано для кодирования последовательности длины 5 на выходе двоичного постоянного источника с вероятностью единицы 0,4. Кодовое слово на выходе арифметического кодера имеет вид: 11010.
- 3.3. $p(A) = 0,3, p(B) = 0,3, p(C) = 0,3, p(D) = 0,1$.
- 3.4. сумма цифр равна 3?
- 3.5. $\pi = 3.141592\dots$?
- 3.6. $m = 30, w_1 = 10, b_1 = 10, r_1 = 10, w_2 = 15, b_2 = 15, r_2 = 0$.

Вариант 4. В фамилии студента общее число букв и число согласных четное

- 4.1. Шел козел с косой козой, шла коза с босым козлом
- 4.2. Арифметическое кодирование использовано для кодирования последовательности длины 6 на выходе двоичного постоянного источника с вероятностью единицы $1/6$. Кодовое слово на выходе арифметического кодера имеет вид: 10111.
- 4.3. $p(A) = 0,2, p(B) = 0,2, p(C) = 0,25, p(D) = 0,35$.
- 4.4. цифры одинаковые – дубль?
- 4.5. $\pi = 3.14159\dots$?
- 4.6. $m = 40, w_1 = 10, b_1 = 9, r_1 = 21, w_2 = 8, b_2 = 7, r_2 = 25$.

1. Теория информации. Базовые понятия теории информации.
2. Способы измерения информации. Вероятностный подход к измерению дискретной и непрерывной информации.
3. Понятие энтропии. Энтропия дискретной случайной величины.
4. Понятие префиксного кодирования. Сжатие информации. Основная теорема о кодировании при отсутствии помех.
5. Метод блокирования.
6. Метод Шеннона-Фэно.
7. Арифметическое кодирование.
8. Адаптивные алгоритмы сжатия информации. Адаптивное арифметическое кодирование.
9. Алгоритмы LZ77, LZ78, LZSS, LZW.
10. Особенности программ архиваторов. Непосредственное применение алгоритмов кодирования в архиваторах для обеспечения продуктивной работы в WINDOWS.
11. Сжатие данных с потерями без ущерба для конфиденциальной и важной информации. Основная теорема о кодировании при наличии помех. Помехозащитное кодирование.
12. Коды с исправлением и обнаружением ошибок.
13. Последовательные коды и их применение на практике.
14. Матричное кодирование.
15. Групповой код.
16. Совершенные и квазисовершенные коды. Их свойства.
17. Полиномиальные коды.
18. Циклические коды.
19. Методика построения кодов, минимальное расстояние между кодовыми словами которых равно заданному числу.
20. Коды Боуза-Чоудхури-Хоккенгема.
21. Циклические избыточные коды (CRC).

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра «Вычислительная техника»

“УТВЕРЖДАЮ”

ДЕКАН АВТФ

к.т.н. Рева И. Л.

“ ” Г.

**ФОНД
ОЦЕНОЧНЫХ СРЕДСТВ
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ**

Теория информации

Образовательная программа: 09.03.01 Информатика и ВТ

бакалавр

Факультет автоматики и вычислительной техники

Новосибирск 2015

Обобщенная структура фонда оценочных средств
по учебной дисциплине
«Теория информации»

Тема	Код формируемой компетенции	Знания/умения	Контролирующее мероприятие
Введение в теорию информации	ОНК.3/И, ОНК.1	Иметь представление о понятии "информация" и ее свойствах. Знать количественные оценки информации; методы кодирования и декодирования сообщений; сущность и значение информации в развитии современного общества, опасности и угрозы, возникающие в этом процессе. Уметь применять методы, способы и средства получения хранения и переработки информации с помощью компьютеров и компьютерных средств; использовать языки и системы программирования для решения профессиональных задач; пользоваться наиболее распространенными офисными и математическими пакетами прикладных программ.	Зачет, вопросы 1 - 3
Введение в теорию сигналов	ОНК.3/И, ОНК.1	Иметь представление о понятии "информация" и ее свойствах. Знать количественные оценки информации; методы кодирования и декодирования сообщений; сущность и значение информации в развитии современного общества, опасности и угрозы, возникающие в этом процессе. Уметь применять методы, способы и средства получения хранения и переработки информации с помощью компьютеров и компьютерных средств; использовать языки и системы программирования для решения профессиональных задач; пользоваться наиболее распространенными офисными и математическими пакетами прикладных программ.	КР, зачет, вопросы 4 - 5
Временное и частотное представление сигналов	ОНК.3/И, ОНК.1	Иметь представление о понятии "информация" и ее свойствах. Знать количественные оценки информации; методы кодирования и декодирования сообщений; сущность и значение информации в развитии современного общества, опасности и угрозы, возникающие в этом процессе. Уметь применять методы, способы и средства получения хранения и переработки информации с помощью компьютеров и компьютерных средств; использовать языки и системы программирования для решения профессиональных задач; пользоваться наиболее распространенными офисными и математическими пакетами прикладных программ.	КР, зачет, вопросы 6 - 8
Преобразование типов сигналов	ОНК.3/И, ОНК.1	Иметь представление о понятии "информация" и ее свойствах. Знать количественные оценки информации; методы кодирования и декодирования сообщений; сущность и значение информации в развитии современного общества, опасности и угрозы, возникающие в этом процессе. Уметь применять методы, способы и средства получения хранения и переработки информации с помощью компьютеров и компьютерных средств; использовать языки и системы программирования для решения профессиональных задач; пользоваться наиболее распространенными офисными и математическими пакетами прикладных программ.	КР, зачет, вопросы 9 - 11
Количественная оценка информации	ОНК.3/И, ОНК.1	Иметь представление о понятии "информация" и ее свойствах. Знать количественные оценки информации; методы кодирования и декодирования сообщений; сущность и значение информации в развитии современного общества, опасности и угрозы, возникающие в этом процессе. Уметь применять методы, способы и средства получения хранения и переработки информации с помощью компьютеров и компьютерных средств; использовать языки и системы программирования для решения профессиональных задач; пользоваться наиболее распространенными офисными и математическими пакетами прикладных программ.	КР, зачет, вопросы 12 - 14

Кодирование сообщений при отсутствии помех	ОНК.3/И, ОНК.1	Иметь представление о понятии "информация" и ее свойствах. Знать количественные оценки информации; методы кодирования и декодирования сообщений; сущность и значение информации в развитии современного общества, опасности и угрозы, возникающие в этом процессе. Уметь применять методы, способы и средства получения хранения и переработки информации с помощью компьютеров и компьютерных средств; использовать языки и системы программирования для решения профессиональных задач; пользоваться наиболее распространенными офисными и математическими пакетами прикладных программ.	КР, зачет, вопросы 15 - 17
Кодирование сообщений при наличии помех	ОНК.3/И, ОНК.1	Иметь представление о понятии "информация" и ее свойствах. Знать количественные оценки информации; методы кодирования и декодирования сообщений; сущность и значение информации в развитии современного общества, опасности и угрозы, возникающие в этом процессе. Уметь применять методы, способы и средства получения хранения и переработки информации с помощью компьютеров и компьютерных средств; использовать языки и системы программирования для решения профессиональных задач; пользоваться наиболее распространенными офисными и математическими пакетами прикладных программ.	КР, зачет, вопросы 18 - 21

Правила аттестации студентов по учебной дисциплине

Для аттестации студентов по дисциплине принят дифференциальный зачет по результатам лабораторного практикума и контрольной работы (КР).

В течение семестра необходимо представить и защитить 4 лабораторные работы и КР в сроки, установленные учебным графиком. К защите допускаются студенты, выполнившие лабораторные работы, в полном объеме (все задания согласно варианту) и оформившие отчет по работе в соответствии с требованиями. На защите предлагается два теоретических вопроса и один практический вопрос (по ходу выполнения работы).

К зачету допускаются студенты, сдавшие лабораторные работы, КР. Зачет проводится в устном виде, предлагаются 2-3 теоретических вопроса.

Критерии оценки результатов обучения

Оценка «отлично» выставляется по дисциплине, если студент полностью и подробно отвечает на вопросы билета (с подготовкой) и заданные во время ответа дополнительные 2-3 вопроса (без дополнительной подготовки) в процессе ответа и на 2-3 вопроса по определениям понятий из лекционного курса, понимает суть излагаемого материала, может свободно беседовать с преподавателем по заданным вопросам.

Оценку «хорошо» студент получает, если не полностью раскрывает суть вопроса, испытывает затруднения в беседе с преподавателем по вопросам билета и/или по 2-3 дополнительным вопросам, неточно определяет и/или понимает понятия из лекционного курса.

Положительную оценку «удовлетворительно» студент получает в том случае, если формально отвечает на вопросы билета и дополнительные, включая определения понятий из лекционного курса, но не может пояснить детали, привести примеры без подсказки, точно нарисовать структурные схемы или написать формулу, дать определение и т.д.

Вопросы для зачета

1. Теория информации. Базовые понятия теории информации.
2. Способы измерения информации. Вероятностный подход к измерению дискретной и непрерывной информации.
3. Понятие энтропии. Энтропия дискретной случайной величины.
4. Понятие префиксного кодирования. Сжатие информации. Основная теорема о кодировании при отсутствии помех.
5. Метод блокирования.
6. Метод Шеннона-Фэно.
7. Арифметическое кодирование.
8. Адаптивные алгоритмы сжатия информации. Арифметическое кодирование.

9. Алгоритмы LZ77, LZ78, LZSS, LZW.

10. Особенности программ архиваторов. Непосредственное применение алгоритмов кодирования в архиваторах для обеспечения продуктивной работы в WINDOWS.

11. Сжатие данных с потерями без ущерба для конфиденциальной и важной информации. Основная теорема о кодировании при наличии помех. Помехозащитное кодирование.

12. Коды с исправлением и обнаружением ошибок.

13. Последовательные коды и их применение на практике.

14. Матричное кодирование.

15. Групповой код.

16. Совершенные и квазисовершенные коды. Их свойства.

17. Полиномиальные коды.

18. Циклические коды.

19. Методика построения кодов, минимальное расстояние между кодовыми словами которых равно заданному числу.

20. Коды Боуза-Чоудхури-Хоккенгема.

21. Циклические избыточные коды (CRC).

**Комплект заданий для выполнения
контрольной работы по учебной дисциплине
«Теория информации»**

1. Построить код Хаффмена для поговорки, найти среднюю длину кодовых слов.
2. Найти последовательность на входе кодера.
3. Алфавит состоит из букв A, B, C, D . Определить количество информации, приходящееся на символ сообщения, составленного из такого алфавита. Вероятности появления букв равны
4. Сколько информации получено, если выбрана фишка домино, у которой
5. Известно, что $3 < \pi < 4$. Сколько дополнительной информации о числе π содержится в записи
6. Имеются две урны, содержащие по m шаров – w_1 и w_2 белых, b_1 и b_2 черных, r_1 и r_2 красных. Из каждой урны извлекают по одному шару. Исход, какого из этих опытов, следует считать более неопределенным, если

Вариант 1. В фамилии студента общее число букв четное, число согласных нечетное

- 1.1. Кто сплетничает с Тобой, тот сплетничает о Тебе.
- 1.2. Арифметическое кодирование использовано для кодирования последовательности длины 5 на выходе двоичного постоянного источника с вероятностью единицы 0,4. Кодовое слово на выходе арифметического кодера имеет вид: 101010.
- 1.3. $p(A) = 0,1, p(B) = 0,2, p(C) = 0,3, p(D) = 0,4$.
- 1.4. цифры разные?
- 1.5. $\pi = 3.14159265...?$
- 1.6. $m = 10, w_1 = 5, b_1 = 3, r_1 = 2, w_2 = 8, b_2 = 1, r_2 = 1$.

Вариант 2. В фамилии студента общее число букв нечетное, число согласных четное

- 2.1. Ехал Грека через реку, видит Грека в реке рак.
- 2.2. Арифметическое кодирование использовано для кодирования последовательности длины 5 на выходе двоичного постоянного источника с вероятностью единицы 0,2. Кодовое слово на выходе арифметического кодера имеет вид: 01110.
- 2.3. $p(A) = 0,25, p(B) = 0,25, p(C) = 0,25, p(D) = 0,25$.
- 2.4. цифры нечетные?
- 2.5. $\pi = 3.1415926...?$
- 2.6. $m = 20, w_1 = 10, b_1 = 5, r_1 = 5, w_2 = 8, b_2 = 8, r_2 = 4$.

Вариант 3. В фамилии студента общее число букв и число согласных нечетное

- 3.1. Сунул Грека руку в реку, рак за руку Греку цап!
- 3.2. Арифметическое кодирование использовано для кодирования последовательности длины 5 на выходе двоичного постоянного источника с вероятностью единицы 0,4. Кодовое слово на выходе арифметического кодера имеет вид: 11010.
- 3.3. $p(A) = 0,3, p(B) = 0,3, p(C) = 0,3, p(D) = 0,1$.
- 3.4. сумма цифр равна 3?
- 3.5. $\pi = 3.141592...?$
- 3.6. $m = 30, w_1 = 10, b_1 = 10, r_1 = 10, w_2 = 15, b_2 = 15, r_2 = 0$.

Вариант 4. В фамилии студента общее число букв и число согласных четное

- 4.1. Шел козел с косой козой, шла коза с босым козлом
- 4.2. Арифметическое кодирование использовано для кодирования последовательности длины 6 на выходе двоичного постоянного источника с вероятностью единицы $1/6$. Кодовое слово на выходе арифметического кодера имеет вид: 10111.
- 4.3. $p(A) = 0,2, p(B) = 0,2, p(C) = 0,25, p(D) = 0,35$.

4.4. цифры одинаковые – дубль?

4.5. $\pi=3.14159\dots$?

4.6. $m = 40, w1 = 10, b1 = 9, r1 = 21, w2 = 8, b2 = 7, r2 = 25$.

Критерии оценки

Каждое задание оценивается по шкале 0, 4, 8 баллов, в зависимости от полноты и правильности ответов.

Работа считается выполненной **на пороговом** уровне, если студент не полностью выполнил все задания, с серьезными замечаниями, недочетами, оценка составляет 16 - 24 баллов.

Работа считается выполненной **на базовом** уровне, если студент не полностью выполнил все задания, без серьезных замечаний и недочетов, оценка составляет 25 - 37 баллов.

Работа считается выполненной **на продвинутом** уровне, если студент полностью выполнил все задания, без замечаний и недочетов, оценка составляет 38 - 48 баллов.

В случае пересдачи КР происходит потеря баллов (максимальное количество баллов составляет 36).

Составитель _____ Е. В. Рабинович
(подпись)

« ____ » _____ 20 ____ г.