ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Иркутский государственный университет путей сообщения»
Сибирский колледж транспорта и строительства

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

К ПРАКТИЧЕСКИМ РАБОТАМ

по дисциплине

ОП.12. Основы теории информации

по специальности

09.02.06 Сетевое и системное администрирование

базовая подготовка среднего профессионального образования

Иркутск 2022

Электронный документ выгружен из ЕИС ФГБОУ ВО ИрГУПС и соответствует оригиналу
Подписант ФГБОУ ВО ИрГУПС Трофимов Ю.А.
00a73c5b7b623a969ccad43a81ab346d50 с 08.12.2022 14:32 по 02.03.2024 14:32 GMT+03:00
Подпись соответствует файлу документа

РАССМОТРЕНО:
Цикловой методической
комиссией специальности 09.02.06
Сетевое и системное
администрирование
«08» июня 2022 г.
Председатель: _____/Саквенко Т.В.

СОГЛАСОВАНО:
Заместитель директора по УВР
_____/А.П.Ресельс
«09» июня 2022 г.

Разработчик: Помазкина Л.И преподаватель высшей категории Сибирского колледжа транспорта и строительства ФГБОУ ВО «Иркутский государственный университет путей сообщения».

Список практических работ

Практическое занятие № 1. Измерение информации	
Практическое занятие № 2 Скорость передачи информации	
Практическое занятие № 3 Скорость передачи информации	
Практическое занятие №4. Выполнение расчетов по теореме отсчетов. Определение пропускной способности дискретного канала	
Практическое занятие №5 Расчет вероятностей	
Практическое занятие №6 Применение теоремы Шеннона.	
Практическое занятие №7. Применение теоремы Шеннона	
Практическое занятие №8. Системные требования алгоритмов сжатия. Алгоритмы сжатия данных неизвестного формата	
Практическое занятие №9 Практическое применение различных алгоритмов сжатия	
Практическое занятие №10 Кодирование числовой и символьной информации	
Практическое занятие №11 Кодирование мультимедийной информации	
Практическое занятие №12Декодирование информации	
Практическое занятие № 13Шифрование с использованием симметричных методов	
Практическое занятие № 14 Шифрование с использованием симметричных методов	
Практическое занятие №15 Шифрование ассиметричными методами	
Практическое занятие №16 Шифрование ассиметричными методами	

Практическая занятие № 1
Измерение информации

Цель: научится определять количество информации с помощью методов Хартли и Шеннона.

Порядок выполнения работы

Изучение материала и выполнение заданий на компьютере.

Содержательный (вероятностный) подход к определению количества информации

Если заключённые в каком-то сообщении сведения являются для человека новыми, понятными, пополняют его знания, т.е. приводят к уменьшению неопределённости знаний, то сообщение содержит информацию.

1 бит – количество информации, которое содержится в сообщении, которое уменьшает неопределённость знаний в 2 раза.

Пример1. При бросании монеты возможны 2 события (случая) – монета упадёт орлом или решкой, причём оба события равновероятны (при большом количестве бросаний количество случаев падения монеты орлом и решкой одинаковы). После получения сообщения о результате падения монеты неопределённость знаний уменьшилась в 2 раза, и, поэтому, количество информации, полученное при этом равно 1 бит.

Содержательный (вероятностный) подход является субъективным, т.к. одну и ту же информацию разные люди могут оценивать по разному. Для одного человека сведения в сообщении могут быть важными и понятными, для другого бесполезными, непонятными или вредными.

Единицы измерения информации. Перевод единиц измерения.

1 бит – количество информации, которое содержится в сообщении, которое уменьшает неопределённость знаний в 2 раза.

1 бит – наименьшая единица информации. Более крупные единицы – байт, килобайт, мегабайт, гигабайт.

Система единиц измерения информации:

1 байт = 8 бит

1 Кбайт = 210 байт = 1024 байт;

1 Мбайт = 210 Кбайт = 1024 Кбайт = 220 байт;

1 Гбайт = 210 Мбайт = 1024 Мбайт = 220 кбайт = 230 байт

Информационный объём носителей информации: Дискета – 1,44 Мбайт; компакт-диск ≈ 700 Мбайт; DVD-диск – до 17 Гбайт (стандарт – 4,7 Гбайт); жёсткий диск – от 20 Гбайт до 80 Гбайт и более (стандарт 80 Гбайт); Flash-память – 256 Мбайт – 2 Гбайт.

Примеры перевода единиц:

1) 5 байт = 5 * 8 бит = 40 бит;

2) 24 бита = 24/8 байта = 3 байта;

3) 4 Кбайт = 4 * 1024 байт = 4096 байт;

4) 16384 бита = 16384 : 8 байт = 2048 байт;

1) 2048 байт / 1024 = 2 Кбайта.

Вычисление количества информации для равновероятных событий.

Если события равновероятны, то количество информации можно рассчитать по формуле:

$N = 2^I$,

где N – число возможных событий,

I – количество информации в битах.

Формула была предложена американским инженером Р. Хартли в 1928 г.

Задача 1. В коробке 32 карандаша, все карандаши разного цвета. Наугад вытащили красный. Какое количество информации при этом было получено?

Решение.

Так как вытаскивание карандаша любого цвета из имеющихся в коробке 32 карандашей является равновероятным, то число возможных событий

равно 32.

N = 32, I = ?

$N = 2^I$, 32 = 25, I = 5 бит.

Ответ: 5 бит.

Задачи на перевод единиц измерения информации

***Задача 1.* В школьной библиотеке 16 стеллажей с книгами, на каждом – по 8 полок. Ученику сообщили, что нужный учебник находится на 2-ой полке 4-го стеллажа. Какое количество информации получил ученик?

Решение.

1) Число стеллажей (случаев) – 16.

N1 = 16, N1 = 2I, 16 = 2I, 16 = 24, I1= 4 бита.

2) Число полок на каждом стеллаже (случаев) – 8,

N2 = 8, N2 = 2I, 8 = 23, I2 = 3 бит.

3) I = I1 + I2, I = 4 бита + 3 бита = 7 бит.

Ответ: 7 бит.

**Задача 3.* Загадывают число в диапазоне от 1 до 200. Какое наименьшее количество вопросов надо задать, чтобы наверняка отгадать число. На вопросы можно отвечать только «Да» или «Нет».

Решение.

Правильная стратегия состоит в том, чтобы количество вариантов каждый раз уменьшалось вдвое.

Например, загадано число 152.

1 вопрос: Число >100? Да.

2 вопрос: Число < 150? Нет.

3 вопрос: Число > 175? Нет. и т.д.

Количество событий в каждом варианте будет одинаково, и их отгадывание равновероятно. N = 2I, 200 = 2I, 7 < I < 8. Т.к. количество вопросов нецелым числом быть не может, то необходимо задать не более 8 вопросов.

Ответ: 8 вопросов

Вычисление количества информации для событий с различными вероятностями.

Существует множество ситуаций, когда возможные события имеют различные вероятности реализации. Рассмотрим примеры таких событий.

1. В коробке 20 карандашей, из них 15 красных и 5 чёрных. Вероятность вытащить наугад красный карандаш больше, чем чёрный.
2. При случайном падении бутерброда вероятность падения его маслом вниз (более тяжёлой стороной) больше, чем маслом вверх.
3. В пруду живут 8000 карасей, 2000 щук и 40000 пескарей. Самая большая вероятность для рыбака – поймать в этом пруду пескаря, на втором месте – карася, на третьем – щуку.

Количество информации в сообщении о некотором событии зависит от его вероятности. Чем меньше вероятность события, тем больше информации оно несёт.

P = K/N, где K – количество случаев реализации одного из исходов события, N – общее число возможных исходов одного из событий

$2^{I} = \log2(1/p)$, где I – количество информации, p – вероятность события

Задача. В коробке 50 шаров, из них 40 белых и 10 чёрных. Определить количество информации в сообщении о вытаскивании наугад белого шара и чёрного шара.

Решение.

Вероятность вытаскивания белого шара

P1 = 40/50 = 0,8

Вероятность вытаскивания чёрного шара P2 = 10/50 = 0,2

Количество информации о вытаскивании белого шара $I_1 = \log2(1/0,8) = \log21,25 = \log1,25/\log2 \approx 0,32$ бит

Количество информации о вытаскивании чёрного шара

$I_2 = \log2(1/0,2) = \log25 = \log5/\log2 \approx 2,32$ бит

Ответ: 0,32 бит, 2,32 бит

Что такое логарифм?

Логарифмом числа a по основанию b называется показатель степени, в которую надо возвести число a, чтобы получить число b.

$\mathbf{a^{\log_a b} = b,\ a > 0,\ b > 0,\ a \neq 1}$

Вычисление логарифмов чисел по основанию 2 с помощью электронного калькулятора

$\log_2 6 = \log 6 / \log 2$, где log 6 и log 2 – десятичные логарифмы

Программа вычисления логарифма числа 6 по основанию 2 ($\log_2 6$) с помощью инженерного калькулятора: 6, log, / , 2, log, =

Количество информации в случае различных вероятностей событий определяется по формуле:

Формула Шеннона: (американский учёный, 1948 г.)

$$I = -\sum_{i=1}^{N} p_i \log_2 p_i$$

где Pi – вероятность i-го события, N – количество возможных событий

Задача. В озере живут караси и окуни. Подсчитано, что карасей 1500, а окуней - 500. Сколько информации содержится в сообщениях о том, что рыбак поймал карася, окуня, поймал рыбу?

Решение.

События поимки карася или окуня не являются равновероятными, так как окуней в озере меньше, чем карасей.

Общее количество карасей и окуней в пруду 1500 + 500 = 2000. Вероятность попадания на удочку карася

p_1 = 1500/2000 = 0,75, окуня p_2 – 500/2000 = 0,25.

I_1 = log2(1/p1), I_1 = log2(1/p2), где I_1 и I_2 – вероятности поймать карася и окуня соответственно.

I_1 = log2(1 / 0,75) ≈ 0,43 бит, I_2 = log2(1 / 0,25) ≈ 2 бит – количество информации в сообщении поймать карася и поймать окуня соответственно.

Количество информации в сообщении поймать рыбу (карася или окуня) рассчитывается по формуле Шеннона

I = - $p_1$log2p1 - $p_2$log2p2

I = - 0,75*log20,75 - 0,25*log20,25 = - 0,75*(log0,75/log2)-0,25*(log0,25/log2) =

= 0,604 бит ≈ 0.6 бит.

Ответ: в сообщении содержится 0,6 бит информации

Вариант № 1

Задание 1.

Игра рассчитана на 2 игрока. Всего играло 4 человека. В результате прошедших всех игр каждый поиграл с каждым по 2 раза. Определить какое количество информации несет в себе сообщение об одной из этих игр.

Задание 2.

В поселке 9 улиц, на каждой улице по 8 домов, в каждом доме живет по 5 человека. Какое количество бит информации содержит данные о каждом из жильцов поселка.

Задание 3.

Кидают не симметричную фигуру, соотношение сторон которой равно пропорции: **P**. Какое количество бит информации содержит сообщение о свершении одного из бросков?

Пропорция: P=4:2:1:7:1

Задание 4.

В зоомагазине в одной клетке живут **W** котенка, в другой клетке **Q** щенка, в следующей **T** кроликов. Определить какое количество информации будет содержать сообщение о покупке одного и з животных.

W=15 Q=4 T=10

Задание 5.

Вычислить какое количество информации будет содержать зрительное сообщение о цвете вынутого шарика, если в непрозрачном мешочке хранятся **X** белых, **Y** красных, **Z** синих и **R** зеленых шариков.

X=28 Y=47 Z=13 R=22

Содержание отчета:

- Номер практической работы
- Тема практической работы

- Цель работы
- Вариант
- Задание
- Решение с подробным описанием действий
- Вывод

Практические занятия № 2, 3

Тема программы**: Способы измерения информации.**

Тема: **Скорость передачи информации.**

Цель: научиться измерять и вычислять скорость передачи информацию.
Время выполнения: 2 часа
Оборудование: ПК.
Раздаточный материал: дидактический материал
Программное обеспечение: операционная система, текстовый редактор.

Теоретические основы

Скорость передачи информации определяется количеством элементов двоичной информации,, передаваемых за 1 с. В синхронной передачи данных по коммутируемым каналам рекомендуется использовать скорости из следующего ряда: 600, 1200, 2400, 4800, 9600 бит/с. Для асинхронной передачи по коммутьируемым каналм рекомегндуется скорость 300 бит/с.

В синхронной передачи данных по арендованным каналам рекомендуется использовать скорости из следующих рядов:

а) предпочтительные: 600, 1200, 2400, 4800, 9600, 14400 бит/с;

б) дополнительные: 3000, 6000, 7200, 12000 бит/с;

в) допустимого диапазона: Nx600 бит/с (1<=N<=24)

Следует отличать скорость передачи информации от модуляционной (линейной) скорости, измеряемой в Бодах (количество элементов модулированного сигнала, передаваемого за 1 с). Для простых видов модуляции скорость передачи информации совпадает с модуляционной скоростью.

При синхронной передаче скорость должна отличаться от номинального значения не более, чем на 0,01%, а при асинхронной – не более, чем на 2,5%.

Оба компьютера, как правило, могут одновременно обмениваться информацией в обе стороны. Этот режим работы называется полным дуплексным.

Дуплексный режим передачи данных – режим, при котором передача данных осуществляется одновременно в обоих направлениях.

В отличие от дуплексного режима передачи данных, полудуплексный подразумевает передачу в каждый момент времени только в одном направлении.

Кроме собственно модуляции и демодуляции сигналов модемы могут выполнять сжатие и декомпрессию пересылаемой информации, а также заниматься поиском и исправлением ошибок, возникнувших в процессе передачи данных по линиям связи.

Одной из основных характеристик модема является скорость модуляции (modulation speed), которая определяет физическую скорость передачи данных без учета исправления ошибок и сжатия данных. Единицей измерения этого параметра является количество бит в секунду (бит/с), называемое бодом.

Любой канал связи имеет ограниченную пропускную способность (скорость передачи информации), это число ограничивается свойствами аппаратуры и самой линии (кабеля).

Объем переданной информации вычисляется по формуле Q=q*t, где q – пропускная способность канала (в битах в секунду), а t – время передачи

Пример 1. Скорость передачи данных через ADSL-соединение равна 128000 бит/с. Через данное соединение передают файл размером 625 кбайт. Определить время передачи файла в секундах.

Решение:

1) выделим в заданных больших числах степени двойки и переведем размер файла в биты, чтобы «согласовать» единиц измерения:

128000 бит/с = 128 · 1000 бит/с = 27 · 125 · 8 бит/с = 27 · 53 · 23 бит/с = 210·53 бит/с

625 кбайт = 54 кбайт = 54 · 213 бит.

2) чтобы найти время передачи в секундах, нужно разделить размер файла на скорость передачи:

t=(54 · 213)бит / 210·53 бит/с = 40 с.

Ответ: 40 с .

Пример 2. Скорость передачи данных через ADSL-соединение равна 512000 бит/с. Передача файла через это соединение заняла 1 минуту. Определить размер файла в килобайтах.

Решение:

1) выделим в заданных больших числах степени двойки; переведем время в секунды (чтобы «согласовать» единицы измерения), а скорость передачи – в кбайты/с, поскольку ответ нужно получить в кбайтах:

1 мин = 60 с = 4 · 15 с = 22 · 15 с

512000 бит/с = 512 · 1000 бит/с = 29 · 125 · 8 бит/с = 29 · 53 · 23 бит/с = 212 · 53 бит/с = 29 · 53 бит/с = (29 · 53) / 210 кбайт/с = (53 / 2) кбайт/с

2) чтобы найти объем файла, нужно умножить время передачи на скорость передачи:

Q=q*t = 22 · 15 с *(53 / 2) кбайт/с = 3750 кбайт

Ответ: 3750 кбайт.

Пример 3. С помощью модема установлена связь с другим компьютером со скоростью соединения 19200, с коррекцией ошибок и сжатием данных.

а) Можно ли при таком соединении файл размером 2,6 килобайт передать за

1 секунду? Обоснуйте свой ответ.
б) Всегда ли при таком соединении файл размером 2,3 килобайт будет передаваться за 1 секунду? Обоснуйте свой ответ.
в) Можно ли при таком соединении оценить время передачи файла размером 4 Мб? Если можно, то каким образом?

Решение:

а) Для начала узнаем, какое количество килобайт мы можем передать за 1 секунду: 19200/1024/8 = 2,3 (Кбайт). Следовательно, если бы не было сжатия информации, то данный файл за одну секунду при данной скорости соединения было бы невозможно передать. Но сжатие есть, 2.6/2.3 < 4, следовательно, передача возможна.
б) Нет не всегда, так как скорость соединения это максимально возможная скорость передачи данных при этом соединении. Реальная скорость может быть меньше.
в) Можно указать минимальное время передачи этого файла: 4*1024*1024/4/19200, около 55 с (столько времени будет передаваться файл на указанной скорости с максимальной компрессией). Максимальное же время передачи оценить вообще говоря нельзя, так как в любой момент может произойти обрыв связи.

Практические задания

Задание 1. Решите задачу о передаче информации с помощью модема.

Вариант 1	Скорость передачи данных через ADSL-соединение равна 512000 бит Через данное соединение передают файл размером 1500 Кб. Определите время передачи файла в секундах.
Вариант 2	Скорость передачи данных через ADSL-соединение равна 1024000 би Через данное соединение передают файл размером 2500 Кб. Определите время передачи файла в секундах.
Вариант 3	Скорость передачи данных через ADSL-соединение равна 1024000 би Передача файла через данное соединение заняла 5 секунд. Определите размер файла в килобайтах.
Вариант 4	Скорость передачи данных через ADSL-соединение равна 512000 бит Передача файла через данное соединение заняла 8 секунд. Определите размер файла в килобайтах.

Задание 2. Решите задачу о передаче графической информации.

Вариант 1	Определите скорость работы модема, если за 256 с он может передать растровое изображение размером 640x480 пикселей. На каждый пиксел приходится 3 байта.
Вариант 2	Сколько секунд потребуется модему, передающему информацию со скоростью 56 000 бит/с, чтобы передать цветное растровое изображение размером 640 x 480 пикселей, при условии, что цвет каждого пикселя кодируется тремя байтами?

Вариант 3	Определите скорость работы модема, если за 132 с он может передать растровое изображение размером 640x480 пикселей. На каждый пиксель приходится 3 байта.
Вариант 4	Сколько секунд потребуется модему, передающему информацию со скоростью 28800 бит/с, чтобы передать цветное растровое изображение размером 640 x 480 пикселей, при условии, что цвет каждого пикселя кодируется тремя байтами?

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

— наименование работы;
— цель работы;
— задание;
— последовательность выполнения работы;
— ответы на контрольные вопросы;
— вывод о проделанной работе.

Контрольные вопросы

1. Дайте определение пропускной способности (передача информации)?
2. В чем измеряется пропускная способность?
3. Зависит ли скорость передачи информации от объема передаваемой информации?

Практическое занятие № 4

Тема программы: Выполнение расчетов по теореме отсчетов. Определение пропускной способности дискретного канала

Цель: Изучение возможности синтезирования сигналов по дискретным отсчетам в соответствии с теоремой Котельникова.

Время выполнения: 2 часа

Оборудование: ПК.

Программное обеспечение: операционная система, калькулятор, текстовый редактор.

Теоретические основы

Теорема Котельникова

В 1933 году В.А. Котельниковым доказана теорема отсчетов [6, 32], имеющая важное значение в теории связи: непрерывный сигнал $s(t)$ с ограниченным спектром можно точно восстановить (интерполировать) по его отсчетам $s(k\Delta t)$, взятым через интервалы $\Delta t = \frac{1}{(2F)}$, где F – верхняя частота спектра сигнала.

В соответствии с этой теоремой сигнал $s(t)$ можно представить рядом Котельникова

$$s(t) = \sum_{k=-\infty}^{\infty} s\left(\frac{k}{2F}\right) \frac{\sin 2\pi F\left[t - \frac{k}{2F}\right]}{2\pi F\left[t - \frac{k}{2F}\right]}. \qquad (1.21)$$

Таким образом, сигнал $s(t)$, можно абсолютно точно представить с помощью последовательности отсчетов $s\left(\frac{k}{2F}\right)$, заданных в дискретных точках $\frac{k}{2F}$ (рис.1.16).

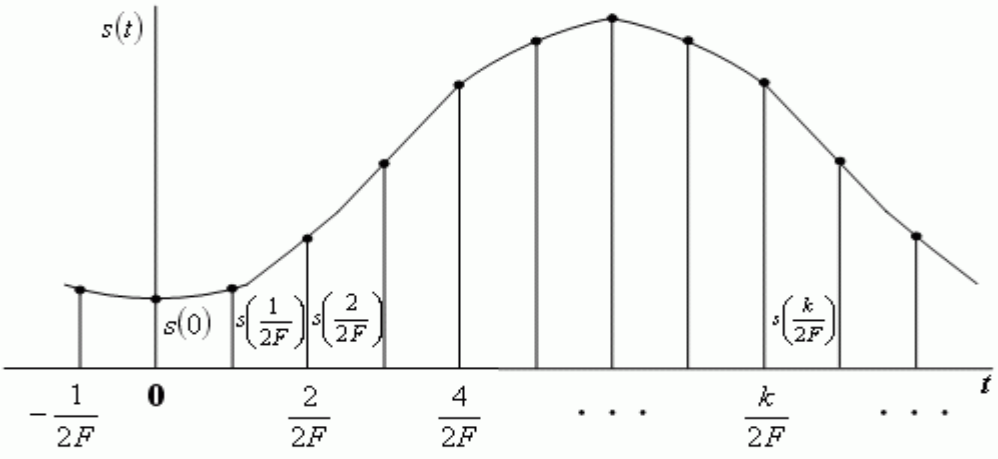


Рис. 1.16. Сигнал и его отсчеты

Функции

$$\psi(t) = \frac{\sin 2\pi F\left[t - \frac{k}{2F}\right]}{2\pi F\left[t - \frac{k}{2F}\right]} \qquad (1.22)$$

образуют ортогональный базис в пространстве сигналов, характеризующихся ограниченным спектром:

$$\Phi(f) = 0 \text{ при } |f| > F. \qquad (1.23)$$

Обычно для реальных сигналов можно указать диапазон частот, в пределах которого сосредоточена основная часть его энергии и которым определяется ширина спектра сигнала. В ряде случаев спектр сознательно сокращают. Это обусловлено тем, что аппаратура и линия связи должны иметь минимальную полосу частот. Сокращение спектра выполняют, исходя из допустимых искажений сигнала. Например, при телефонной связи хорошая разборчивость речи и узнаваемость абонента обеспечиваются при передаче сигналов в полосе частот $\Delta F = 0{,}3...3{,}4\,[\text{кГц}]$. Увеличение ΔF приводит к неоправданному усложнению аппаратуры и повышению затрат. Для передачи телевизионного изображения при стандарте в 625 строк полоса частот, занимаемая сигналом, составляет около 6 МГц.

Из вышесказанного следует, что процессы с ограниченными спектрами могут служить адекватными математическими моделями многих реальных сигналов.

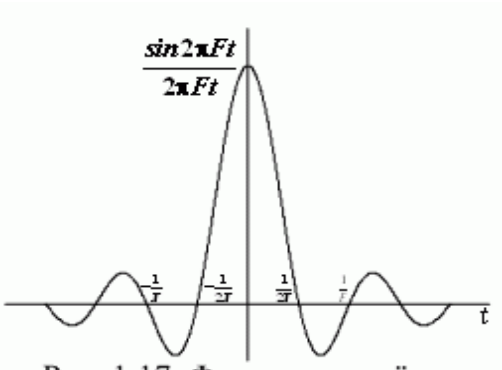


Рис. 1.17. Функция отсчётов

Функция вида $\frac{\sin 2\pi F\left[t - \frac{k}{2F}\right]}{2\pi F\left[t - \frac{k}{2F}\right]}$ называется функцией отсчетов (рис.1.17).

Она характеризуется следующими свойствами. Если $k = 0$, функция отсчетов имеет максимальное значение при $t = 0$, а в моменты времени $t = \frac{i}{2F}$ ($i = 1,2,...$) она обращается в нуль; ширина главного лепестка функции отсчетов на нулевом уровне равна $\frac{1}{F}$, поэтому минимальная длительность импульса, который может существовать на выходе линейной системы с полосой пропускания F, равна $\frac{1}{F}$; функции отсчетов ортогональны на бесконечном интервале времени.

На основании теоремы Котельникова может быть предложен следующий способ дискретной передачи непрерывных сигналов:

Для передачи непрерывного сигнала $s(t)$ по каналу связи с полосой пропускания F определим мгновенные значения сигнала $s(t)$ в дискретные

моменты времени $t_k = \frac{1}{2F}$, ($k = 0,1,2,\ldots$). После этого передадим эти значения по каналу связи каким - либо из возможных способов и восстановим на приемной стороне переданные отсчеты. Для преобразования потока импульсных отсчетов в непрерывную функцию пропустим их через идеальный ФНЧ с граничной частотой F.

Можно показать, что энергия сигнала находится по формуле [6, 32]:

$$E = \int_{-\infty}^{\infty} s^2(t)dt = \frac{1}{2F}\sum_{k=-\infty}^{\infty} s^2\left(\frac{k}{2F}\right). \quad (1.24)$$

Для сигнала, ограниченного во времени, выражение (1.24) преобразуется к виду:

$$E = \int_{1}^{2FT} s^2(t)dt = \frac{1}{2F}\sum_{k=1}^{2FT} s^2\left(\frac{k}{2F}\right). \quad (1.25)$$

Выражение (1.25) широко применяется в теории помехоустойчивого приема сигналов, но является приближенным, т.к. сигналы не могут быть одновременно ограничены по частоте и времени.

Практическое задание

1. Изобразить сигналы, синтезируемые в лабораторной работе:

а) синусоидальный сигнал частотой 5кГц;

б) видеоимпульсы прямоугольной формы длительностью 0,25; 0,5; 1,0 мс;

в) видеоимпульсы пилообразной формы длительностью 0,5 мс; 1,0 мс.

2. Рассчитать и построить идеальные выборочные сигналы для сигналов, указанных в п. 1а, 1б, 1в, при $f_{выб}$=5, 10, 20, 40 кГц.

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

— наименование работы;
— цель работы;
— задание;
— последовательность выполнения работы;
— ответы на контрольные вопросы;
— вывод о проделанной работе.

Контрольные вопросы

1. Сформулируйте теорему Котельникова для сигналов с ограниченным спектром.

2. Объясните погрешности синтезирования реальных сигналов по дискретным отсчетам.

Практическое занятие № 4, 5

Тема программы: **Смысл энтропии Шеннона.**

Тема: **Расчет вероятностей.**

Цель: научиться вычислять вероятности событий (появление символов в сообщении) и рассчитывать энтропию.

Время выполнения: 2 часа

Оборудование: ПК.

Программное обеспечение: операционная система, калькулятор, текстовый редактор.

Теоретические основы

Количество информации по Хартли и Шеннону

Понятие количество информации отождествляется с понятием информация. Эти два понятия являются синонимами. Мера информации должна монотонно возрастать с увеличением длительности сообщения (сигнала), которую естественно измерять числом символов в дискретном сообщении и временем передачи в непрерывном случае. Кроме того, на содержание количества информации должны влиять и статистические характеристики, так как сигнал должен рассматриваться как случайный процесс.

При этом наложено ряд ограничений:

1. Рассматриваются только дискретные сообщения.

2. Множество различных сообщений конечно.

3. Символы, составляющие сообщения равновероятны и независимы.

Хартли впервые предложил в качестве меры количества информации принять логарифм числа возможных последовательностей символов.

$$I=\log m^k=\log N \qquad (1)$$

К.Шеннон попытался снять те ограничения, которые наложил Хартли. На самом деле в рассмотренном выше случае равной вероятности и независимости символов при любом k все возможные сообщения оказываются также равновероятными, вероятность каждого из таких сообщений равна P=1/N. Тогда количество информации можно выразить

через вероятности появления сообщений I=–log P.

В силу статистической независимости символов, вероятность сообщения длиной в k символов равна

$$P = \prod_{i=1}^{k} p_i$$

Если i–й символ повторяется в данном сообщении k_i раз, то $P = \prod_{i=1}^{m} p_i^{k_i}$

так как при повторении i символа k_i раз k уменьшается до m. Из теории вероятностей известно, что, при достаточно длинных сообщениях (большое число символов k) $k_i \approx k \cdot p_i$ и тогда вероятность сообщений будет равняться

$$P = \prod_{i=1}^{m} p_i^{kp_i}$$

Тогда окончательно получим

$$I = -\log P = -k\sum_{i=1}^{m} p_i \log p_i \qquad (2)$$

Данное выражение называется формулой Шеннона для определения количества информации.

Формула Шеннона для количества информации на отдельный символ сообщения совпадает с энтропией. Тогда количество информации сообщения состоящего из k символов будет равняться I=k·H

Количество информации, как мера снятой неопределенности

При передаче сообщений, о какой либо системе происходит уменьшение неопределенности. Если о системе все известно, то нет смысла посылать сообщение. Количество информации измеряют уменьшением энтропии.

Количество информации, приобретаемое при полном выяснении состояния некоторой физической системы, равно энтропии этой системы:

$$I = -\sum_{i=1}^{n} p_i \log p_i$$

Количество информации I – есть осредненное значение логарифма вероятности состояния. Тогда каждое отдельное слагаемое –log p_i

необходимо рассматривать как частную информацию, получаемую от отдельного сообщения, то есть

$$I_i = -\log p_i$$

Избыточность информации

Если бы сообщения передавались с помощью равновероятных букв алфавита и между собой статистически независимых, то энтропия таких сообщений была бы максимальной. На самом деле реальные сообщения строятся из не равновероятных букв алфавита с наличием статистических связей между буквами. Поэтому энтропия реальных сообщений $-H_p$, оказывается много меньше оптимальных сообщений – H_o. Допустим, нужно передать сообщение, содержащее количество информации, равное I. Источнику, обладающему энтропией на букву, равной H_p, придется затратить некоторое число n_p, то есть

$$I = n_p H_p$$

Если энтропия источника была бы H_0, то пришлось бы затратить меньше букв на передачу этого же количества информации

$$I = n_0 H_0 \qquad n_o = \frac{I}{H_o} < n_p$$

Таким образом, часть букв $n_p - n_o$ являются как бы лишними, избыточными. Мера удлинения реальных сообщений по сравнению с оптимально закодированными и представляет собой избыточность D.

$$D = 1 - \frac{H_p}{H_o} = 1 - \frac{n_o}{n_p} = \frac{n_p - n_o}{n_p} \qquad (3)$$

Но наличие избыточности нельзя рассматривать как признак несовершенства источника сообщений. Наличие избыточности способствует повышению помехоустойчивости сообщений. Высокая избыточность естественных языков обеспечивает надежное общение между людьми.

Частотные характеристики текстовых сообщений

Важными характеристиками текста являются повторяемость букв, пар букв

(биграмм) и вообще m-ок (m-грамм), сочетаемость букв друг с другом, чередование гласных и согласных и некоторые другие. Замечательно, что эти характеристики являются достаточно устойчивыми.

Идея состоит в подсчете чисел вхождений каждой n^m возможных m-грамм в достаточно длинных открытых текстах $T=t_1t_2\ldots t_l$, составленных из букв алфавита $\{a_1, a_2, \ldots, a_n\}$. При этом просматриваются подряд идущие m-граммы текста

$t_1t_2\ldots t_m, t_2t_3\ldots t_{m+1}, \ldots, t_{i-m+1}t_{l-m+2}\ldots t_l$.

Если $\vartheta(a_{i1}a_{i2}\ldots a_{im})$ – число появлений m-граммы $a_{i1}a_{i2}\ldots a_{im}$ в тексте T, а L общее число подсчитанных m-грамм, то опыт показывает, что при достаточно больших L частоты

$\frac{\vartheta(a_{i1}a_{i2}\ldots a_{im})}{L}$ для данной m-граммы мало отличаются друг от друга.

В силу этого, относительную частоту считают приближением вероятности $P(a_{i1}a_{i2}\ldots a_{im})$ появления данной m-граммы в случайно выбранном месте текста (такой подход принят при статистическом определении вероятности).

Для русского языка частоты (в порядке убывания) знаков алфавита, в котором отождествлены Е с Ё, Ь с Ъ, а также имеется знак пробела (-) между словами, приведены в таблице 1.

Таблица 1

- 0.175	О 0.090	Е, Ё 0.072	А 0.062
И 0.062	Т 0.053	Н 0.053	С 0.045
Р 0.040	В 0.038	Л 0.035	К 0.028
М 0.026	Д 0.025	П 0.023	У 0.021
Я 0.018	Ы 0.016	З 0.016	Ь, Ъ 0.014
Б 0.014	Г 0.013	Ч 0.012	Й 0.010
Х 0.009	Ж 0.007	Ю 0.006	Ш 0.006
Ц 0.004	Щ 0.003	Э 0.003	Ф 0.002

Некоторая разница значений частот в приводимых в различных источниках таблицах объясняется тем, что частоты существенно зависят не только от длины текста, но и от его характера.

Устойчивыми являются также частотные характеристики биграмм, триграмм и четырехграмм осмысленных текстов.

Порядок выполнения работы

Определить количество информации (по Хартли), содержащееся в заданном сообщении, при условии, что значениями являются буквы кириллицы.

«Фамилия Имя Отчество» завершил ежегодный съезд эрудированных школьников, мечтающих глубоко проникнуть в тайны физических явлений и химических реакций

Построить таблицу распределения частот символов, характерные для заданного сообщения. Производится так называемая частотная селекция, текст сообщения анализируется как поток символов и высчитывается частота встречаемости каждого символа. Сравнить с имеющимися данными в табл 1.

На основании полученных данных определить среднее и полное количество информации, содержащееся в заданном сообщении. Оценить избыточность сообщения.

1. Построить таблицу распределения частот символов, характерных для заданного сообщения путём деления количества определённого символа в данном сообщении на общее число символов

По формуле

$$\sum_{i=1}^{m} p_i \log p_i$$

H= вычислил энтропию сообщения

2. Далее по формуле Шеннона для определения кол-ва информации

$$I = -\log P = -k\sum_{i=1}^{m} p_i \log p_i$$

вычислил кол-во информации в передаваемом сообщении

3. Вычислил избыточность D по формуле

$$D = 1 - \frac{H_p}{H_o} = 1 - \frac{n_o}{n_p} = \frac{n_p - n_o}{n_p}$$

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:
— наименование работы;
— цель работы;
— задание;
— последовательность выполнения работы;
— ответы на контрольные вопросы;
— вывод о проделанной работе.

Контрольные вопросы

1. Дать определение понятие энтропия?

2. Что означает вероятностный способ измерения информации?

3. Что означает статическое определение вероятности?

Практическое занятие № 6, 7

Тема: **Применение теоремы Шеннона. Поиск энтропии случайных величин.**

Цель: научиться вычислять энтропию случайной величины.
Время выполнения: 4 часа
Оборудование: ПК.
Программное обеспечение: операционная система, калькулятор, текстовый редактор.

Теоретические основы

Энтропия в теории информации — мера хаотичности информации, неопределённость появления какого-либо символа первичного алфавита. При отсутствии информационных потерь численно равна количеству информации на символ передаваемого сообщения.

Так, возьмём, например, последовательность символов, составляющих какое-либо предложение на русском языке. Каждый символ появляется с разной частотой, следовательно, неопределённость появления для некоторых символов больше, чем для других. Если же учесть, что некоторые сочетания символов встречаются очень редко, то неопределённость ещё более уменьшается (в этом случае говорят об энтропии n-ого порядка. Концепции информации и энтропии имеют глубокие связи друг с другом, но, несмотря на это, разработка теорий в статистической механике и теории информации заняла много лет, чтобы сделать их соответствующими друг другу.

Энтропия независимых случайных событий x с n возможными состояниями (от 1 до n) рассчитывается по формуле:

$$H(x) = -\sum_{i=1}^{n} p(i) \log_2 p(i)$$

Эта величина также называется *средней энтропией сообщения*.

Величина $\log_2 \frac{1}{p(i)}$ называется *частной энтропией*, характеризующей только i-е состояние.

Таким образом, энтропия события x является суммой с противоположным знаком всех произведений относительных частот появления события i, умноженных на их же двоичные логарифмы (основание 2 выбрано только для удобства работы с информацией, представленной в двоичной форме). Это определение для дискретных случайных событий можно расширить для функции распределения вероятностей.

Шеннон вывел это определение энтропии из следующих предположений:

- мера должна быть непрерывной; т. е. изменение значения величины вероятности на малую величину должно вызывать малое результирующее изменение энтропии;
- в случае, когда все варианты (буквы в приведенном примере) равновероятны, увеличение количества вариантов (букв) должно всегда увеличивать полную энтропию;
- должна быть возможность сделать выбор (в нашем примере букв) в два шага, в которых энтропия конечного результата должна будет является суммой энтропий промежуточных результатов.

Шеннон показал, что любое определение энтропии, удовлетворяющее этим предположениям, должно быть в форме:

$$-K \sum_{i=1}^{n} p(i) \log_2 p(i)$$

где K — константа (и в действительности нужна только для выбора единиц измерения).

Шеннон определил, что измерение энтропии ($H = -p_1 \log_2 p_1 - \ldots - p_n \log_2 p_n$), применяемое к источнику информации, может определить требования к минимальной пропускной способности канала, требуемой для надежной передачи информации в виде закодированных двоичных чисел. Для вывода формулы Шеннона необходимо вычислить математическое ожидания «количества информации», содержащегося в цифре из источника информации. Мера энтропии Шеннона выражает неуверенность реализации случайной переменной. Таким образом, энтропия является разницей между информацией, содержащейся в сообщении, и той частью информации, которая точно известна (или хорошо предсказуема) в сообщении. Примером этого является избыточность языка — имеются явные статистические закономерности в появлении букв, пар последовательных букв, троек и т.д.

В общем случае **b-арная энтропия** (где b равно 2,3,...) источника $\mathcal{S} = (S,P)$ с исходным алфавитом $S = \{a_1, \ldots, a_n\}$ и дискретным распределением вероятности $P = \{p_1, \ldots, p_n\}$ где p_i является вероятностью a_i ($p_i = p(a_i)$) определяется формулой:

$$H_b(\mathcal{S}) = -\sum_{i=1}^{n} p_i \log_b p_i$$

Определение энтропии Шеннона очень связано с понятием термодинамической энтропии. Больцман и Гиббс проделали большую работу по статистической термодинамике, которая способствовала принятию слова «энтропия» в информационную теорию. Существует связь между понятиями энтропии в термодинамике и теории информации. Например, демон Максвелла также противопоставляет термодинамическую энтропию информации, и получение какого-либо количества информации равно потерянной энтропии.

СВОЙСТВА ЭНТРОПИИ

*1.*Энтропия является вещественной и неотрицательной величиной.

*2.*Энтропия – величина ограниченная.

*3.*Энтропия обращается в нуль лишь в том случае, если вероятность одного из состояний равна единице; тогда вероятности всех остальных состояний, естественно, равны нулю. Это положение соответствует случаю, когда состояние источника полностью определено.

*4.*Энтропия максимальна, когда все состояния источника равновероятны.

5. Энтропия источника и с двумя состояниями u1 и u2 изменяется от нуля до единицы, достигая максимума при равенстве их вероятностей:

p(u1) = p= p(u2) = 1 — p = 0,5.

*6.*Энтропия объединения нескольких статистически независимых источников информации равна сумме энтропии исходных источников.

7. Энтропия характеризует среднюю неопределенность выбора одного состояния из ансамбля. При ее определении используют только вероятности состояний, полностью игнорируя их содержательную сторону. Поэтому энтропия не может служить средством решения любых задач, связанных с неопределенностью.

8. Энтропия как мера неопределенности согласуется с экспериментальными данными, полученными при изучении психологических реакций человека, в частности реакции выбора. Установлено, что время безошибочной реакции на последовательность беспорядочно чередующихся равновероятных раздражителей (например, загорающихся лампочек) растет с увеличением их числа так же, как энтропия. Это время характеризует неопределенность выбора одного раздражителя. Замена равновероятных раздражителей неравновероятными приводит к снижению среднего времени реакции ровно настолько, насколько уменьшается энтропия.

Дифференциальной энтропией случайной величины X называется величина:

$$H_Д(x)=H(x)-H(y)= -\int_{-\infty}^{+\infty} p_x(x) * \log_2 d * p_x(x)dx$$

Если произвести квантование случайных величин X_1, $X_2 \ldots X_n$ по уровню с числом уровней квантования равным m , то возможное число реализаций длительностью T_n станет конечным и равным $M = m^n$.

Каждая из реализаций C_1, C_2,….C_i,…C_m будет иметь определенную вероятность появления в эксперименте по наблюдению реализаций. Тогда неопределенность (энтропия) и количество информации в реализации (в среднем по всем реализациям) определяются равенством

$$H_n = -\sum_{i=1}^{M} P(C_i)\log(P(C_i))$$

$$H = \frac{H_n}{n} = -\frac{1}{n}\sum_{i=1}^{M} P(C_i)\log(P(C_i))$$

Энтропия и количество информации на одну степень свободы (на одну выборку) равны

$$H = \frac{H_n}{n} = -\frac{1}{n}\sum_{i=1}^{M} P(C_i)\log(P(C_i))$$

Избыточность показывает, какая доля максимально возможной при заданном объеме алфавита неопределенности не используется источником.

$\mu = (H_{max} - H_u)/H_{max}$,

Где H_u – энтропия рассматриваемого источника, H_{max} – максимально возможное значение его энтропии, которое может быть достигнуто подбором распределения и ликвидацией взаимозависимости элементов алфавита. Так, для дискретного источника с М элементами

$H_{max} = \log M$

Выполнение расчетных задач

Задача №1

Доказать, что $H(x,y) \le H(x) + H(y)$.

Решение:

По определению, $H(x,y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i, y_j)\log p(x_i, y_j)$.

Для статистически зависимых событий $p(x_i, y_j) = p(x_i)\,p(y_j / x_i)$.

$$H(x,y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i)\,p(y_j / x_i)\log\left[p(x_i)\,p(y_j / x_i)\right] =$$

$$= -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i)\,p(y_j / x_i)\left[\log p(x_i) + \log p(y_j / x_i)\right] =$$

$$= -\sum_{i=1}^{n} p(x_i)\log p(x_i)\underbrace{\sum_{j=1}^{m} p(y_j / x_i)}_{1} + \sum_{i=1}^{n} p(x_i)\underbrace{\left(-\sum_{j=1}^{m} p(y_j / x_i)\log p(y_j / x_i)\right)}_{H(y/x_i)} =$$

$$= \underbrace{-\sum_{i=1}^{n} p(x_i)\log p(x_i)}_{H(x)} + \underbrace{\sum_{i=1}^{n} p(x_i)\,H(x, y / x_i)}_{H(y/x)} = H(x) + H(y / x).$$

$H(y / x_i)$ - это частная энтропия Y при условии, что известно состояние $X = x_i$.

Наличие информации о состоянии X не может увеличить неопределенность состояния Y, но может уменьшить его в случае зависимости Y от X. Значит, условная энтропия $H\left(y/x_i\right)$ не больше безусловной энтропии $H\left(y\right)$, то есть $H\left(y/x_i\right) \le H\left(y\right)$. Тогда средняя условная энтропия $H\left(y/x\right) = \sum_{i=1}^{n} p\left(x_i\right) H\left(y/x_i\right) \le \sum_{i=1}^{n} p\left(x_i\right) H\left(y\right) = H\left(y\right) \underbrace{\sum_{i=1}^{n} p\left(x_i\right)}_{1} = H\left(y\right)$,

то есть $H\left(y/x\right) \le H\left(y\right)$.

Значит, $H\left(x,y\right) = H\left(x\right) + H\left(y/x\right) \le H\left(x\right) + H\left(y\right)$.

Задача №2

Показать, что для регулярной марковской цепи энтропия $H(x)^{(r)}$ за r шагов равняется энтропии за один шаг, умноженной на число шагов r.

Решение:

Регулярная цепь Маркова полностью характеризуется матрицей переходных вероятностей $P = \begin{pmatrix} p_{11} & \cdots & p_{1m} \\ . & . & . \\ p_{m1} & & p_{mm} \end{pmatrix}$ и предельным стационарным распределением вероятностей состояний $\left(p_1, p_2, \ldots, p_m\right)$.

В стационарном режиме энтропия за один шаг не зависит от номера шага и равна $H\left(X\right)^{(1)} = \sum_{k=1}^{m} p_k H_k\left(x\right)$,

p_k - стационарная вероятность *k*-го состояния,

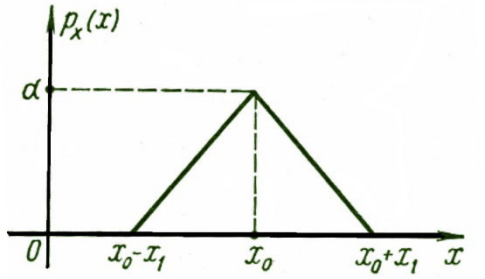
$H_k\left(x\right) = -\sum_{i=1}^{m} p_{ki} \log p_{ki}$ - энтропия в *k*-м состоянии.

Энтропия за *r* шагов равна сумме энтропий за каждый шаг. Так как энтропия за каждый шаг одинакова, то сумма энтропий равна $H\left(X\right)^{(r)} = r \cdot H\left(X\right)^{(1)}$.

Задача №3

Вычислить энтропию $H_д(x)$ распределения $p_x(x)$ изображенного на рисунке.

Построить график зависимости $H_д(x)$ в функции параметра d.



Ответ:

4.36. Энтропия треугольного распределения равна $-\ln d + {}^1/_2$.

Решение:

Из условия нормировки $S = d \cdot x_1 = 1 \Rightarrow x_1 = \dfrac{1}{d}$.

Плотность данного распределения

$$p_X(x) = \begin{cases} d + \dfrac{d(x - x_0)}{x_1}, & x_0 - x_1 \le x \le x_0 \\ d - \dfrac{d(x - x_0)}{x_1}, & x_0 \le x \le x_0 + x_1 \end{cases}$$

Дифференциальная энтропия

$$H_Д(x) = -\int_{-\infty}^{\infty} p_X(x) \log p_X(x)\,dx = -\left(\int_{x_0 - x_1}^{x_0} p_X(x) \ln p_X(x)\,dx + \int_{x_0}^{x_0 + x_1} p_X(x) \ln p_X(x)\,dx \right).$$

$$\int u \ln u\,du = \ln u \cdot \frac{u^2}{2} - \int (\ln u)' \frac{u^2}{2}\,du = -\frac{u^2}{4} + \frac{1}{2} u^2 \ln u + C = \frac{1}{2} u^2 \left(\ln u - \frac{1}{2} \right) + C.$$

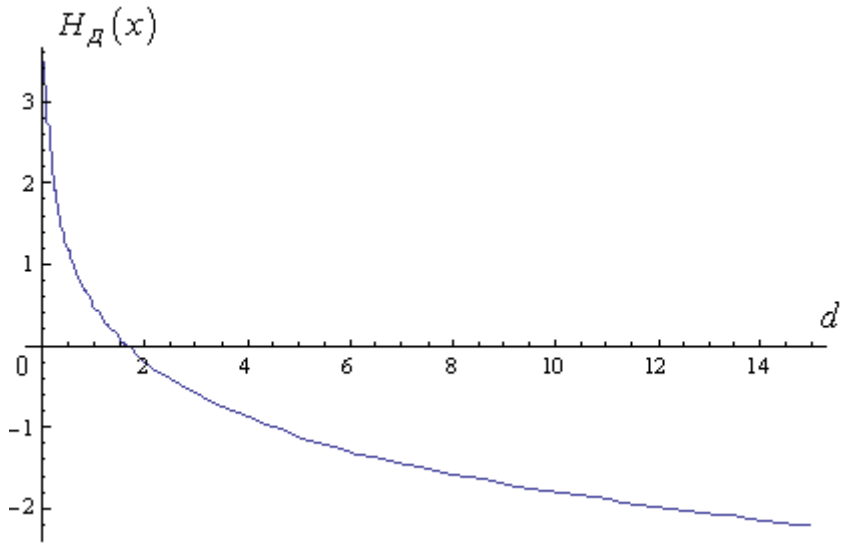
$$\int_{x_0 - x_1}^{x_0} \left[d + \frac{d}{x_1}(x - x_0) \right] \ln \left[d + \frac{d}{x_1}(x - x_0) \right] dx =$$

$$= \frac{x_1}{d} \left(\frac{1}{2} \left[d + \frac{d}{x_1}(x - x_0) \right]^2 \left(\ln \left[d + \frac{d}{x_1}(x - x_0) \right] - \frac{1}{2} \right) \right) \Bigg|_{x_0 - x_1}^{x_0} = \frac{x_1}{d} \cdot \frac{1}{2} d^2 \left(\ln d - \frac{1}{2} \right).$$

$$\int_{x_0}^{x_0 + x_1} \left[d - \frac{d}{x_1}(x - x_0) \right] \ln \left[d - \frac{d}{x_1}(x - x_0) \right] dx =$$

$$= -\frac{x_1}{d} \left(\frac{1}{2} \left[d - \frac{d}{x_1}(x - x_0) \right]^2 \left(\ln \left[d - \frac{d}{x_1}(x - x_0) \right] - \frac{1}{2} \right) \right) \Bigg|_{x_0}^{x_0 + x_1} = \frac{x_1}{d} \cdot \frac{1}{2} d^2 \left(\ln d - \frac{1}{2} \right).$$

$$H_Д(x) = -2 \cdot \frac{x_1}{d} \cdot \frac{1}{2} d^2 \left(\ln d - \frac{1}{2} \right) = -\frac{x_1}{d} d^2 \left(\ln d - \frac{1}{2} \right) = -\frac{1}{d^2} d^2 \left(\ln d - \frac{1}{2} \right) = -\ln d + \frac{1}{2}.$$



Задача 4

В результате полной дезорганизации управления m самолетов летят произвольными курсами. Управление восстановлено, и все самолеты взяли общий курс со среднеквадратической ошибкой отклонения от курса σ=3^0. Найти изменение энтропии, считая, что в первом случае имело место равномерное распределение вероятностей углов, а во втором случае – нормальное.

Ответ: 4.86 бита

Решение.

Начальное распределение вероятностей углов курсов самолетов равномерное в интервале от $a = 0$ до $b = 360° = 2\pi\ рад$ с плотностью вероятности $p_{1X}(x) = \dfrac{1}{b-a},\ \ a \le x \le b$.

Дифференциальная энтропия этого распределения

$$H_{1Д}(x) = -\int_a^b \frac{1}{b-a}\log_2\frac{1}{b-a}dx = -\log_2\frac{1}{b-a} = \log_2(b-a) = \log_2 2\pi = 2{,}65 \text{ бит.}$$

Конечное распределение вероятностей углов курсов самолетов нормальное с параметрами $a = 0,\ \ \sigma = 3° = \dfrac{\pi}{60}\,рад$ и плотностью вероятности

$$p_{2X}(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-x^2/(2\sigma^2)}.$$

Дифференциальная энтропия этого распределения

$$H_{2Д}(x) = -\int_{-\infty}^{\infty} p_{2X}(x)\log_2\left[\frac{1}{\sigma\sqrt{2\pi}}e^{-x^2/(2\sigma^2)}\right]dx =$$

$$= -\log_2\frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{\infty} p_{2X}(x)dx + \frac{\log_2 e}{2\sigma^2}\int_{-\infty}^{\infty} x^2 p_{2X}(x)dx =$$

$$= -\log_2\frac{1}{\sigma\sqrt{2\pi}} + \frac{\log_2 e}{2\sigma^2}\cdot\sigma^2 = \log_2\sigma\sqrt{2\pi} + \frac{\log_2 e}{2} = \log_2\frac{\pi\sqrt{2\pi e}}{60} = -2,21\ \textit{бит}.$$

Изменение энтропии $\Delta H(x) = H_{2Д}(x) - H_{1Д}(x) = -2,21 - 2,65 = -4,86$ бит.

Энтропия уменьшилась на 4,86 бит.

Задача 5

Измерительное устройство вырабатывает временные интервалы, распределенные случайным образом в пределах от 100 до 500 мс. Как изменится энтропия случайной величины при изменении точности измерения с 1 мс до 1 мкс?

Ответ: Энтропия увеличивается примерно на 10 бит

Решение.

При точности 1мс дискретная случайная величина X – результат измерения – может равновероятно принимать одно из $n = \frac{500-100}{1} = 400$ значений. Энтропия равна $H_1(x) = \log_2 n$.

При точности 1мкс дискретная случайная величина X – результат измерения – может равновероятно принимать одно из $m = \frac{500-100}{10^{-3}} = 400\cdot 10^3 = 1000n$ значений. Энтропия равна $H_2(x) = \log_2 m$.

Изменение энтропии

$\Delta H(x) = H_2(x) - H_1(x) = \log_2 m - \log_2 n = \log_2 1000n - \log_2 n = \log_2 1000 \approx \log_2 1024 = 10$ бит.

Энтропия увеличилась примерно на 10 бит.

Задача 6

Записать отношения между энтропиями:

H(x), H(y), H(x|y), H(y|x), H(x,y), H(x|y_j), H(y|x_i)

Решение.

Связь между энтропией совместного распределения и полными энтропиями и средними условными энтропиями

$H(x,y) = H(x) + H(y/x) = H(y) + H(x/y)$.

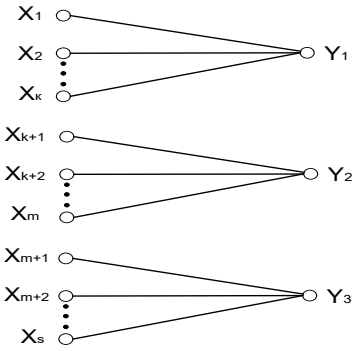
Если x и y независимы, то $H(y/x)=H(y)$, $H(x/y)=H(x)$ и энтропия совместного распределения $H(x,y)=H(x)+H(y)$ максимальна, так как $H(y/x)\le H(y)$, $H(x/y)\le H(x)$.

Связь между частными и средними условными энтропиями

$$H(y/x)=\sum_{i=1}^{n}p(x_i)H(y/x_i),\ \ H(x/y)=\sum_{j=1}^{m}p(y_j)H(x/y_j).$$

Задача 7

На рисунке представлена диаграмма канала со слабым разрешением. Определить количество информации, передаваемое по каналу.



Ответ: I (x,y)=H(x)

Решение:

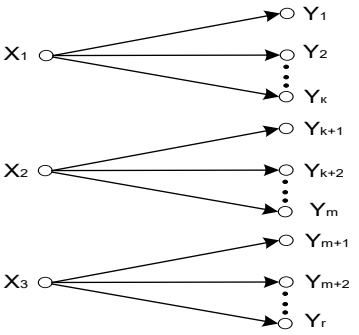
Количество переданной по каналу информации равно $I=H-H_{апост.}$.

$H=H(y)$ – энтропия полученных элементов y до отправления элемента x; $H_{апост.}=H(y/x)$ - энтропия полученных элементов y после отправления элемента x. Для данного канала со слабым разрешением $H(y/x)=0$, так как полученный элемент y однозначно определяется по отправленному элементу x.

Получаем $I=H(y)$.

Задача 8

На рисунке представлена диаграмма канала с неоднозначностью. Определить количество информации, передаваемое по каналу.



ответ: I (x,y)=H(x)

Решение:

Количество переданной по каналу информации равно $I = H - H_{апост.}$.

$H = H(x)$ – энтропия отправленных элементов *x* до получения элемента *y*; $H_{апост.} = H(x/y)$ - энтропия отправленных элементов *x* после получения элемента *y*. Для данного канала с неоднозначностью $H(x/y) = 0$, так как полученный элемент *y* однозначно определяет отправленный элемент *x*.

Получаем $I = H(x)$.

Задачи по вычислению энтропии

1. Найдите энтропию для числа белых шаров при извлечении двух шаров из урны, содержащей два белых и один черный шар.

2. Найдите энтропию для числа козырных карт при извлечении двух карт из колоды в 36 карт.

3. Какую степень неопределенности содержит опыт угадывания суммы очков на извлеченной кости из полного набора домино?

4. Найдите энтропию для числа тузов при извлечении трех карт из карт с картинками.

5. Найдите дифференциальную энтропию для равномерного распределения.

6. Найдите дифференциальную энтропию для показательного закона распределения, если известно, что случайная величина *x* принимает значение меньше единицы с вероятностью 0,5.

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

— наименование работы;
— цель работы;
— задание;
— последовательность выполнения работы;
— ответы на контрольные вопросы;
— вывод о проделанной работе.

Контрольные вопросы

1. Как определяется энтропия дискретных случайных величин?

2. Приведите примеры энтропий для классических законов распределения.

Практическое занятие №8

Системные требования алгоритмов сжатия. Алгоритмы сжатия данных неизвестного формата:

Цель: научиться сжимать информацию с помощью метода Хаффмана и метода RLE.

Время выполнения: 2 часа

Оборудование: ПК.

Программное обеспечение: операционная система, калькулятор, текстовый редактор.

Теоретические основы

Сжатие данных (англ. *data compression*) — алгоритмическое преобразование данных, производимое с целью уменьшения их объёма. Применяется для более рационального использования устройств хранения и передачи данных. Синонимы — упаковка данных, компрессия, сжимающее кодирование, кодирование источника. Обратная процедура называется восстановлением данных (распаковкой, декомпрессией). Сжатие основано на устранении избыточности, содержащейся в исходных данных. Простейшим примером избыточности является повторение в тексте фрагментов (например, слов естественного или машинного языка). Подобная избыточность обычно устраняется заменой повторяющейся последовательности ссылкой на уже закодированный фрагмент с указанием его длины. Другой вид избыточности связан с тем, что некоторые значения в сжимаемых данных встречаются чаще других. Сокращение объёма данных достигается за счёт замены часто встречающихся данных короткими кодовыми словами, а редких — длинными (энтропийное кодирование). Сжатие данных, не обладающих свойством избыточности (например, случайный сигнал или белый шум, зашифрованные сообщения), принципиально невозможно без потерь. В основе любого способа сжатия лежит модель источника данных, или, точнее, модель избыточности. Иными словами, для сжатия данных используются некоторые априорные сведения о том, какого рода данные сжимаются. Не обладая такими сведениями об источнике, невозможно сделать никаких предположений о преобразовании, которое позволило бы уменьшить объём сообщения. Модель избыточности может быть статической, неизменной для всего сжимаемого сообщения, либо строиться или параметризоваться на этапе сжатия (и восстановления). Методы, позволяющие на основе входных данных изменять модель избыточности информации, называются адаптивными. Неадаптивными являются обычно узкоспециализированные алгоритмы, применяемые для работы с данными, обладающими хорошо определёнными и неизменными характеристиками. Подавляющая часть достаточно универсальных алгоритмов являются в той или иной мере адаптивными.

Все методы сжатия данных делятся на два основных класса:

- Сжатие без потерь

- Сжатие с потерями

При использовании сжатия без потерь возможно полное восстановление исходных данных, сжатие с потерями позволяет восстановить данные с искажениями, обычно несущественными с точки зрения дальнейшего использования восстановленных данных. Сжатие без потерь обычно используется для передачи и хранения текстовых данных, компьютерных программ, реже — для сокращения объёма аудио- и видеоданных, цифровых фотографий и т. п., в случаях, когда искажения недопустимы или нежелательны. Сжатие с потерями, обладающее значительно большей, чем сжатие без потерь, эффективностью, обычно применяется для сокращения объёма аудио- и видеоданных и цифровых фотографий в тех случаях, когда такое сокращение является приоритетным, а полное соответствие исходных и восстановленных данных не требуется.

Системные требования алгоритмов

Различные алгоритмы могут требовать различного количества ресурсов вычислительной системы, на которых они реализованы:

- оперативной памяти (под промежуточные данные);
- постоянной памяти (под код программы и константы);
- процессорного времени.

В целом, эти требования зависят от сложности и «интеллектуальности» алгоритма. Общая тенденция такова: чем эффективнее и универсальнее алгоритм, тем большие требования к вычислительным ресурсам он предъявляет. Тем не менее, в специфических случаях простые и компактные алгоритмы могут работать не хуже сложных и универсальных. Системные требования определяют их потребительские качества: чем менее требователен алгоритм, тем на более простой, а следовательно, компактной, надёжной и дешёвой системе он может быть реализован.

Так как алгоритмы сжатия и восстановления работают в паре, имеет значение соотношение системных требований к ним. Нередко можно усложнив один алгоритм значительно упростить другой. Таким образом, возможны три варианта:

Алгоритм сжатия требует больших вычислительных ресурсов, нежели алгоритм восстановления.

Это наиболее распространённое соотношение, характерное для случаев, когда однократно сжатые данные будут использоваться многократно. В качестве примера можно привести цифровые аудио- и видеопроигрыватели.

Алгоритмы сжатия и восстановления требуют приблизительно равных вычислительных ресурсов.

Наиболее приемлемый вариант для линий связи, когда сжатие и восстановление происходит однократно на двух её концах (например, в цифровой телефонии).

Алгоритм сжатия существенно менее требователен, чем алгоритм восстановления.

Такая ситуация характерна для случаев, когда процедура сжатия реализуется простым, часто портативным устройством, для которого объём доступных ресурсов весьма критичен, например, космический аппарат или большая распределённая сеть датчиков. Это могут быть также данные, распаковка которых требуется в очень малом проценте случаев, например запись камер видеонаблюдения.

Алгоритмы сжатия данных неизвестного формата

Имеется два основных подхода к сжатию данных неизвестного формата.

- На каждом шаге алгоритма сжатия очередной сжимаемый символ либо помещается в выходной буфер сжимающего кодера как есть (со специальным флагом, помечающим, что он не был сжат), либо группа из нескольких сжимаемых символов заменяется ссылкой на совпадающую с ней группу из уже закодированных символов. Поскольку восстановление сжатых таким образом данных выполняется очень быстро, такой подход часто используется для создания самораспаковывающихся программ.
- Для каждой сжимаемой последовательности символов однократно либо в каждый момент времени собирается статистика её встречаемости в кодируемых данных. На основе этой статистики вычисляется вероятность значения очередного кодируемого символа (либо последовательности символов). После этого применяется та или иная разновидность энтропийного кодирования, например, арифметическое кодирование или кодирование Хаффмана, для представления часто встречающихся последовательностей короткими кодовыми словами, а редко встречающихся — более длинными.

Код Хаффмана

Определение 1: Пусть $A=\{a_1,a_2,...,a_n\}$ - алфавит из n различных символов, $W=\{w_1,w_2,...,w_n\}$ - соответствующий ему набор положительных целых весов. Тогда набор бинарных кодов $C=\{c_1,c_2,...,c_n\}$, такой что:

1) (c_i не является префиксом для c_j, при $i!=j$

2) ($\sum_{i=1}^{n} w_i|$ минимальна ($|c_i|$ длина кода c_i)

называется *минимально-избыточным префиксным кодом* или иначе *кодом Хаффмана*.

Замечания:

1. Свойство (1) называется *свойством префиксности*. Оно позволяет однозначно декодировать коды переменной длины.

2. Сумму в свойстве (2) можно трактовать как размер закодированных данных в битах. На практике это очень удобно, т.к. позволяет оценить степень сжатия не прибегая непосредственно к кодированию.

3. В дальнейшем, чтобы избежать недоразумений, под кодом будем понимать битовую строку определенной длины, а под минимально-избыточным кодом или кодом Хаффмана - множество кодов (битовых строк), соответствующих определенным символам и обладающих определенными свойствами.

Известно, что любому бинарному префиксному коду соответствует определенное бинарное дерево.

Определение 2: Бинарное дерево, соответствующее коду Хаффмана, будем называть *деревом Хаффмана*.

Задача построения кода Хаффмана равносильна задаче построения соответствующего ему дерева. Приведем общую схему построения дерева Хаффмана:

1. Составим список кодируемых символов (при этом будем рассматривать каждый символ как одноэлементное бинарное дерево, вес которого равен весу символа).
2. Из списка выберем 2 узла с наименьшим весом.
3. Сформируем новый узел и присоединим к нему, в качестве дочерних, два узла выбранных из списка. При этом вес сформированного узла положим равным сумме весов дочерних узлов.
4. Добавим сформированный узел к списку.
5. Если в списке больше одного узла, то повторить 2-5.

Приведем пример: построим дерево Хаффмана для сообщения S="A H F B H C E H E H C E A H D C E E H H H C H H H D E G H G G E H C H H".

Для начала введем несколько обозначений:

1. Символы кодируемого алфавита будем выделять жирным шрифтом: **A**, **B**, **C**.
2. Веса узлов будем обозначать нижними индексами: $\mathbf{A}_5$, $\mathbf{B}_3$, $\mathbf{C}_7$.
3. Составные узлы будем заключать в скобки: $((\mathbf{A}_5+\mathbf{B}_3)_8+\mathbf{C}_7)_{15}$.

Итак, в нашем случае A={**A**, **B**, **C**, **D**, **E**, **F**, **G**, **H**}, W={2, 1, 5, 2, 7, 1, 3, 15}.

1. $\mathbf{A}_2\ \mathbf{B}_1\ \mathbf{C}_5\ \mathbf{D}_2\ \mathbf{E}_7\ \mathbf{F}_1\ \mathbf{G}_3\ \mathbf{H}_{15}$
2. $\mathbf{A}_2\ \mathbf{C}_5\ \mathbf{D}_2\ \mathbf{E}_7\ \mathbf{G}_3\ \mathbf{H}_{15}\ (\mathbf{F}_1+\mathbf{B}_1)_2$
3. $\mathbf{C}_5\ \mathbf{E}_7\ \mathbf{G}_3\ \mathbf{H}_{15}\ (\mathbf{F}_1+\mathbf{B}_1)_2\ (\mathbf{A}_2+\mathbf{D}_2)_4$
4. $\mathbf{C}_5\ \mathbf{E}_7\ \mathbf{H}_{15}\ (\mathbf{A}_2+\mathbf{D}_2)_4\ ((\mathbf{F}_1+\mathbf{B}_1)_2+\mathbf{G}_3)_5$
5. $\mathbf{E}_7\ \mathbf{H}_{15}\ ((\mathbf{F}_1+\mathbf{B}_1)_2+\mathbf{G}_3)_5\ (\mathbf{C}_5+(\mathbf{A}_2+\mathbf{D}_2)_4)_9$
6. $\mathbf{H}_{15}\ (\mathbf{C}_5+(\mathbf{A}_2+\mathbf{D}_2)_4)_9\ (((\mathbf{F}_1+\mathbf{B}_1)_2+\mathbf{G}_3)_5+\mathbf{E}_7)_{12}$

7. $\mathbf{H}_{15}$ $((\mathbf{C}_5+(\mathbf{A}_2+\mathbf{D}_2)_4)_9+(((\mathbf{F}_1+\mathbf{B}_1)_2+\mathbf{G}_3)_5+\mathbf{E}_7)_{12})_{21}$
8. $(((\mathbf{C}_5+(\mathbf{A}_2+\mathbf{D}_2)_4)_9+(((\mathbf{F}_1+\mathbf{B}_1)_2+\mathbf{G}_3)_5+\mathbf{E}_7)_{12})_{21}+\mathbf{H}_{15})_{36}$

В списке, как и требовалось, остался всего один узел. Дерево Хаффмана построено. Теперь запишем его в более привычном для нас виде.

```
                ROOT
                  /\
                 0  1
                /   \
               /\    H
                / \
               /   \
              /     \
             0       1
            /         \
           /           \
          /             \
         /               \
        /                 \
       /\                 /\
      0  1               0  1
     /   \             /   \
    C    /\           /\    E
        0  1         0  1
       /   \        /   \
      A     D      /\    G
                  0  1
                 /   \
                F     B
```

Листовые узлы дерева Хаффмана соответствуют символам кодируемого алфавита. Глубина листовых узлов равна длине кода соответствующих символов.

Путь от корня дерева к листовому узлу можно представить в виде битовой строки, в которой "0" соответствует выбору левого поддерева, а "1" - правого. Используя этот механизм, мы без труда можем присвоить коды всем символам кодируемого алфавита. Выпишем, к примеру, коды для всех символов в нашем примере:

A=0010_{bin}	**C**=000_{bin}	**E**=011_{bin}	**G**=0101_{bin}
B=01001_{bin}	**D**=0011_{bin}	**F**=01000_{bin}	**H**=1_{bin}

Теперь у нас есть все необходимое для того чтобы закодировать сообщение S. Достаточно просто заменить каждый символ соответствующим ему кодом:

$S^{/}$="0010 1 01000 01001 1 000 011 1 011 1 000 011 0010 1 0011 000 011 011 1 1 1 000 1 1 1 0011 011 0101 1 0101 0101 011 1 000 1 1".

Оценим теперь степень сжатия. В исходном сообщении S было 36 символов, на каждый из которых отводилось по $[\log_2|A|]=3$ бита (здесь и далее будем понимать квадратные скобки [] как целую часть, округленную в положительную сторону, т.е. [3,018]=4). Таким образом, размер S равен 36*3=108 бит

Размер закодированного сообщения $S^{/}$ можно получить воспользовавшись замечанием 2 к определению 1, или непосредственно, подсчитав количество бит в $S^{/}$. И в том и другом случае мы получим 89 бит.

Итак, нам удалось сжать 108 в 89 бит.

Теперь декодируем сообщение $S^{/}$. Начиная с корня дерева будем двигаться вниз, выбирая левое поддерево, если очередной бит в потоке равен "0", и правое - если "1". Дойдя до листового узла мы декодируем соответствующий ему символ.

Ясно, что следуя этому алгоритму мы в точности получим исходное сообщение S.

Метод RLE.

Наиболее известный простой подход и алгоритм сжатия информации обратимым путем - это кодирование серий последовательностей (Run Length Encoding - RLE). Суть методов данного подхода состоит в замене цепочек или серий повторяющихся байтов или их последовательностей на один кодирующий байт и счетчик числа их повторений. Проблема всех аналогичных методов заключается лишь в определении способа, при помощи которого распаковывающий алгоритм мог бы отличить в результирующем потоке байтов кодированную серию от других - некодированных последовательностей байтов. Решение проблемы достигается обычно простановкой меток в начале кодированных цепочек. Такими метками могут быть, например, характерные значения битов в первом байте кодированной серии, значения первого байта кодированной серии и т.п. Данные методы, как правило, достаточно эффективны для сжатия растровых графических изображений (BMP, PCX, TIF, GIF), т.к. последние содержат достаточно много длинных серий повторяющихся последовательностей байтов. Недостатком метода RLE является достаточно низкая степень сжатия или стоимость кодирования файлов с малым числом серий и, что еще хуже - с малым числом повторяющихся байтов в сериях.

Практическое задание

1. Сжатие методом Хаффмана

«КАКАЯ ЗИМА ЗОЛОТАЯ!
КАК БУДТО ИЗ ДЕТСКИХ ВРЕМЕН...

НЕ НАДО НИ СОЛНЦА, НИ МАЯ –
ПУСТЬ ДЛИТСЯ ТОРЖЕСТВЕНИЫЙ СОН.

ПУСТЬ Я В ЭТОМ СНЕ ПОЗАБУДУ
КОГДА-ТО МАНИВШИЙ ОГОНЬ,
И ЛЕТО ПРЕДАМ, КАК ИУДА,
ЗА ТРИДЦАТЬ СНЕЖИНОК В ЛАДОНЬ.

ЗАТЕМ, ЧТО И Я ХОЛОДЕЮ,
ТЕПЛО УЖЕ СТРАШНО ПРИНЯТЬ:
Я СЛИШКОМ ДАВНО НЕ УМЕЮ
НИ ТЛЕТЬ, НИ ГОРЕТЬ, НИ СЖИГАТЬ…

ВСЕ ЧАЩЕ, ВСЕ ДОЛЬШЕ НЕМЕЮ:
К ЗИМЕ УЖЕ ДЕЛО, К ЗИМЕ...
И ТОЛЬКО ТОГО ОТОГРЕЮ,
КОМУ ХОЛОДНЕЕ, ЧЕМ МНЕ»

2. С помощью сжатия по методу RLE.

1 последовательность:

SSSSOOOEEERROOOAAYYYYYDDDDOEUUUUUWWWWJJJORRUUUUUUUUUXXXKHHHHHHMMMMMMGGGLLLLLLLJJJJ

2 последовательность:

FFFFFFFFKKKKKSSSSUURERRRRRRRRPPPPPPPPDDDDKKKKKKGLDDDDDDDDKKKKKKKKGGGGMGMMMM

Отчет

Отчет должен содержать:

— наименование работы;
— цель работы;
— задание;
— последовательность выполнения работы;
— ответы на контрольные вопросы;
— вывод о проделанной работе.

Контрольные вопросы

1. Запишите какие Вы знаете методы сжатия информации?
2. Перечислите основные программы применяемые для сжатия информации?

Практическое занятие №9

Практическое применение различных алгоритмов сжатия

Цель: Целью практической работы является получение навыков работы с архиваторами RAR, ARJ и ZIP, и ознакомление с основными алгоритмами сжатия информации.

Время выполнения: 2 часа

Оборудование: ПК.

Программное обеспечение: операционная система, программы архиваторы: RAR, ARJ и ZIP

Теоретические основы

При эксплуатации персональных компьютеров по самым различным причинам возможны порча или потеря информации на магнитных дисках. Это может произойти из - за физической порчи магнитного диска, неправильной корректировки или случайного уничтожения файлов, разрушения информации компьютерным вирусом и т.д . Для того чтобы уменьшить потери в таких ситуациях, следует иметь архивные копии используемых файлов и систематически обновлять копии изменяемых файлов. Для хранения архивов данных можно использовать внешние запоминающие устройства большой емкости , которые дают возможность легко скопировать жесткий диск (например, магнитооптика, стримеры, " Арвид " и др.) Однако при этом резервные копии занимают столько же места, сколько занимают исходные файлы , и для копирования нужных файлов может потребоваться много дискет. Более удобно для создания архивных копий использовать специально разработанные программы архивации файлов, которые сжимают информацию . При архивировании степень сжатия файлов сильно зависит от их формата. Некоторые форматы данных (графические, Page Maker и др.) имеют упакованные разновидности, при этом сжатие производится создающей исходный файл программой, однако лучшие архиваторы способны поджать и их. Совсем другая картина наблюдается при архивации текстовых файлов. Текстовые файлы обычно сжимаются на 50-70%, а программы на 20-30%. Принцип работы архиваторов основан на поиске в файле " избыточной" информации и последующем ее кодировании с целью получения минимального объема. Самым известным методом архивации файлов является сжатие последовательностей одинаковых символов. Например, внутри вашего файла находятся последовательности байтов , которые часто повторяются. Вместо того чтобы хранить каждый байт , фиксируется количество

повторяющихся символов и их позиция. Для наглядности приведем следующий пример: Упаковываемый файл занимает 15 байт и состоит из следующей последовательности символов: BBBBBLLLLLAAAAA

В шестнадцатиричной системе :

42 42 42 42 42 4 C 4 C 4 C 4 C 4 C 41 41 41 41 41

Архиватор может представить этот файл в следующем шестнадцатиричном

виде : 01 05 42 06 05 4 C OA 05 41

Эти последовательности можно интерпретировать следующим образом: с первой позиции 5 раз повторяется знак B, с шестой позиции 5 раз повторяется знак L и с позиции 11 5 раз повторяется знак A. Согласитесь, очень простая демонстрация алгоритма архивации . Очевидно, что для хранения файла в его последней форме требуется лишь 9 байт - меньше на 6 байт . Описанный метод является простым и очень эффективным способом сжатия файлов. Однако он не обеспечивает большой экономии объема, если обрабатываемый текст содержит небольшое количество последовательностей повторяющихся символов.

Существуют два основных способа проведения сжатия:

- статистический
- словарный.

Лучшие статистические методы применяют арифметическое кодирование, лучшие словарные - метод Зива -Лемпела . В статистическом сжатии каждому символу присваивается код , основанный на вероятности его появления в тексте . Высоко вероятные символы получают короткие коды, и наоборот . Такой способ сжатия называют оптимальным префиксным кодом . Для его построения используют алгоритмы Хаффмана или Шеннона- Фано. Например, анализируя любой английский текст, можно установить , что буква E встречается гораздо чаще, чем Z, а X и Q относятся к наименее встречающимся . Таким образом, используя специальную таблицу соответствия, можно закодировать каждую букву E меньшим числом бит, используя более длинный код для более редких букв , тогда как в обычных кодировках любому символу соответствует битовая последовательность фиксированной длины (как правило, кратной байту). В словарном методе группы последовательных символов или " фраз " заменяются кодом. Замененная фраза может быть найдена в некотором " словаре". Популярные архиваторы ARJ, RAR работают на основе алгоритма Лемпела - Зива . Сущность алгоритмов Зива и Лемпела состоит в том , что фразы заменяются указателем на то место, где они в тексте уже ранее появлялись . Это семейство алгоритмов обозначается как LZ-сжатие . Такой метод быстро приспосабливается к структуре текста и может

кодировать короткие функциональные слова, т.к. они очень часто в нем появляются. 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 декодировщика . Одной из форм такого указателя является пара (m,l), которая заменяет фразу из l символов, начинающуюся со смещения m во входном потоке. Например, указатель (7,2) адресует 7-ой и 8-ой символы исходной строки . Используя это обозначение, строка " abbaabbbabab" будет закодирована как " abba(1,3)(3,2)(8,3) ". Заметим , что несмотря на рекурсию в последнем указателе, производимое кодирование не будет двусмысленным . Распространено не верное представление, что за понятием LZ- метода стоит единственный алгоритм . Из- за большого числа вариантов этого метода лучшее описание можно осуществить только через его растущую семью , где каждый член отражает свое решение разработчика . Эти версии отличаются друг от друга в двух главных факторах : есть ли предел обратного хода указателя, и на какие подстроки из этого множества он может ссылаться.

Продвижение указателя в ранее просмотренную часть текста может быть неограниченным (расширяющееся окно) или ограничено окном постоянного размера из N предшествующих символов, где N обычно составляет несколько тысяч. Выбранные подстроки также могут быть неограниченным или ограниченным множеством фраз, выбранных согласно некоторому замыслу .

Каждая комбинация этих условий является компромиссом между скоростью выполнения, объемом требуемой ОП и качеством сжатия.

Расширяющееся окно предлагает лучшее сжатие за счет организации доступа к большему количеству подстрок . Но по мере роста окна , кодировщик может замедлить свою работу из - за возрастания времени поиска соответствующих подстрок, а сжатие может ухудшиться из - за увеличения размеров указателей. Если памяти для окна будет не хватать, произойдет сброс

процесса, что также ухудшит сжатие до поры нового увеличения окна .

Окно постоянного размера лишено этих проблем , но содержит меньше подстрок, доступных указателю. Ограничение множества доступных подстрок размерами фиксированного окна уменьшает размер указателей и убыстряет кодирование.

К основным функциям архиваторов относятся :

- архивация указанных файлов или всего текущего каталога ;
- извлечение отдельных или всех файлов из архива ;
- просмотр содержимого архивного файла;
- проверка целостности архивов;
- восстановление поврежденных архивов;
- ведение многотомных архивов;
- вывод файлов из архива на экран или на печать ;
- парольная защита архива .

Архиватор ARJ не имеет графического интерфейса , и вся работа с ним осуществляется с командной строки . Формат команд имеет следующий вид :

arj <команда > [- <спецификация 1> [- <спецификация 2>]…] < имя архива>
[< имя файла >…]

Подробную информацию о списке команд архиватора можно получить, набрав в командной строке: arj /?

Рассмотрим наиболее популярные команды архиватора:

Для архивации файлов: arj a < имя архива> <имя файла 1> < имя файла 2>

Для извлечения файлов из архива: arj e < имя архива> <имя файла 1> < имя файла 2>

Для просмотра содержимого архивного файла: arj l < имя архива>

Для проверки целостности архива: arj t < имя архива>

Для восстановления испорченного архива : arj -jr < имя архива>

Для создания многотомного архива: arj a –v< размер тома> <имя архива> <имя файла 1> < имя файла 2>

Для вывода файла из архива на экран: arj p < имя архива> <имя файла >

Для создания архива с паролем: arj a –g< пароль> <имя архива> <имя файла > или arj a –g? < имя архива> <имя файла > в последнем случае пароль будет запрошен отдельной строкой.

Для создания самораспаковывающихся архивов: arj a -je <имя архива> <имя файла 1> < имя файла 2>

Архиватор RAR имеет версии , как для Dos, Win 3.XX так и для Windows 95/98. Последние версии WinRar имеют графический интерфейс и работа с ними очень проста и понятна. Данный архиватор позволяет создавать как архивы *.rar так и архивы *.zip

К достоинствам данного архиватора можно отнести:

- графический интерфейс;
- высокую степень сжатия, даже мультимедийных файлов;
- возможность оценить размер архива, не производя архивирование.
- большую вероятность восстановления поврежденных архивов.

Практическое задание

1.Найдите на компьютере не менее 5-ти текстовых файлов (расширение .txt)

2.Произведите их сжатие архиватором RAR в обычный и SFX- архив.

3.Зафиксируйте размер файла до сжатия и после него.

4.Вычислите коэффициент сжатия (отношение размера исходного файла к размеру сжатого файла)

5.Повторите пункты 1-4 для графических файлов (расширение .bmp)

6.Повторите пункты 1-4 для графических файлов (расширение .jpg)

7.Повторите пункты 1-4 для звуковых файлов (расширение .wav)

8.Сведите полученные результаты в таблицу. Сделайте выводы о том, какие файлы сжимаются лучше.

9.Напишите отчет о проделанной работе.

Содержание отчета

Отчет должен содержать следующие разделы:

Ответы на контрольные вопросы.

Результаты сжатия файлов в виде таблицы.

Выводы о проделанной работе.

Контрольные вопросы

1. Зачем нужно архивировать информацию ?
2. На чем основана работа архиваторов. По какому принципу они сжимают информацию.
3. Каковы функции архиваторов.
4. Чем отличаются SFX – архивы.

Практическое занятие № 10

Кодирование числовой и символьной информации

Цель: Познакомиться с различными кодировками символов, используя текстовые редакторы, выполнить задания в различных текстовых приложениях.
Время выполнения: 2 часа
Оборудование: ПК.
Программное обеспечение: операционная система, текстовые редакторы.

Теоретические основы

Правило цифрового представления символов следующее: каждому символу ставится в соответствие некоторое целое число, то есть каждый символ нумеруется.

Пример:

Рассмотрим последовательность строчных букв русского алфавита: а, б, в, г, д, е, ё, ж, з, и, й. к, л, м. н. о, п, р, с, т, у, ф, х, ц, ч, ш, щ, ъ, ы, в, э, ю, я. Присвоив каждой букве номер от 0 до 33. получим простейший способ представления символов. Последнее число - 32 в двоичной форме имеет вид 100000, то есть для хранения символа в памяти понадобится 6 бит.Так как с помощью шести бит можно представить число $2^6 - 1 = 63$, то шести бит будет достаточно для представления 64 букв.

Имеются разные стандарты для представления, символов, которые отличаются лишь порядком нумерации символов. Наиболее распространён американский стандартный код для информационного обмена - ASCII [American Standard-Code for Information Interchange] введён в США в 1963г. В 1977 году в несколько модифицированном виде он был принят в качестве всемирного стандарта Международной организации стандартов [International Standards Organization -. ISO] под названием ISO-646. Согласно этому стандарту каждому символу поставлено в соответствие число от 0 до 255. Символы от 0 до 127 - латинские буквы, цифры и знаки препинания - составляют постоянную часть таблицы. Остальные символы используются для представления национальных алфавитов. Конкретный состав этих символов определяется кодовой страницей. В русской версии ОС Windows95 используется кодовая, страница 866. В ОС Linux для представления русских букв более употребительна кодировка КОИ-8. Недостатки такого способа кодировки национального, алфавита очевидны. Во-первых, невозможно

одновременное представление русских и ,например, французских букв. Во-вторых, такая кодировка совершенно непригодна для представления, китайских иероглифов. В 1991 году была создана некоммерческая организация Unicode, в которую входят представители ряда фирм (Borland. IBM, Noyell, Sun и др) и которая занимается развитием и внедрением нового стандарта. Кодировка Unicode использует 16 разрядов ,и может содержать 65536 символов. Это символы большинства народов мира, элементы иероглифов, спецсимволы, 5000 – мест для частного использования, резерв из 30000 мест.

Пример:

ASCII-код символа A= 65_{10} =41_{16}= 01000111_2;

Unicode-код символа C= 67_{10}=0000000001100111_2

Задания

1. Закодируйте свое имя, фамилию и отчество с помощью одной из таблиц (win-1251, KOI-8)
2. Раскодируйте ФИО соседа
3. Закодируйте следующие слова, используя таблицы ASCII-кодов: ИНФОРМАТИЗАЦИЯ, МИКРОПРОЦЕССОР, МОДЕЛИРОВАНИЕ
4. Раскодируйте следующие слова, используя таблицы ASCII-кодов:

88 AD E4 AE E0 AC A0 E2 A8 AA A0

50 72 6F 67 72 61 6D

43 6F 6D 70 75 74 65 72 20 49 42 4D 20 50 43

5. Текстовый редактор Блокнот

Открыть блокнот.

а) Используя клавишу Alt и малую цифровую клавиатуру раскодировать фразу: 145 170 174 224 174 255 170 160 173 168 170 227 171 235;

Технология выполнения задания: При удерживаемой клавише Alt, набрать на малой цифровой клавиатуре указанные цифры. Отпустить клавишу Alt, после чего в тексте появится буква, закодированная набранным кодом.

б) Используя ключ к кодированию, закодировать слово – зима;

Технология выполнения задания: Из предыдущего задания выяснить, каким кодом записана буква а. Учитывая, что буквы кодируются в алфавитном порядке, выяснить коды остальных букв.

Что вы заметили при выполнении этого задания во время раскодировки? Запишите свои наблюдения.

6. Текстовый процессор.

Технология выполнения задания: рассмотрим на примере: представить в различных кодировках слово Кодировка

Решение:

- Создать новый текстовый документ в текстовом редакторе;
- Выбрать – Команда – Вставка – Символ.

В открывшемся окне «Символ» установить из: Юникод (шестн.),

- В наборе символов находим букву **К** и щелкнем на ней левой кнопкой мыши (ЩЛКМ).
- В строке код знака появится код выбранной буквы 041А (незначащие нули тоже записываем).
- У буквы **о** код – 043Е и так далее: д – 0434, и – 0438, р – 0440, о – 043Е, в – 0432, к – 043А, а – 0430.
- Установить Кириллица (дес.)
- К – 0202, о – 0238, д – 0228, и – 0232, р – 0240, о – 0238, в –0226, к – 0202, а –0224.

7. Открыть Текстовый редактор

Используя окно «Вставка символа» выполнить задания: Закодировать слово **Forest**

а) Выбрать шрифт Courier New, кодировку ASCII(дес.) Ответ: **70 111 114 101 115 116**

б) Выбрать шрифт Courier New, кодировку Юникод(шест.) Ответ: **0046 006F 0072 0665 0073 0074**

в) Выбрать шрифт Times New Roman, кодировку Кирилица(дес.) Ответ: **70 111 114 101 115 116**

г) Выбрать шрифт Times New Roman, кодировку ASCII(дес.) Ответ: **70 111 114 101 115 116**

Вывод: _____

Выполнение лабораторной работы оформить в виде таблицы.

8. Буква Z имеет десятичный код 90, а z – 122. Записать слово «sport» в десятичном коде.

9. С помощью десятичных кодов зашифровано слово «info» 105 110 102 111. Записать последовательность десятичных кодов для этого же слова, но записанного заглавными буквами.

10. Буква Z имеет десятичный код 90, а z – 122. Записать слово «forma» в десятичном коде.

11. С помощью десятичных кодов зашифровано слово «port» 112 111 114 116. Записать последовательность десятичных кодов для этого же слова, но записанного заглавными буквами. Ответ: **80 79 82 84**

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

— наименование работы;
— цель работы;
— задание;
— последовательность выполнения работы;
— ответы на контрольные вопросы;
— вывод о проделанной работе.

Контрольные вопросы

1. Какие возможности предоставляет текстовые редакторы по работе с

символами?
2. Какие вы знаете алгоритмы кодирования информации?
3. Где применяется алгоритм кодирования информации?

Практическое занятие №11
Кодирование мультимедийной информации

Цель: познакомиться с различными кодировками звуковой информации и с характеристиками звуковых файлов.
Время выполнения: 2 часа
Оборудование: ПК.
Программное обеспечение: операционная система, звуковые редакторы.

Теоретические основы

С начала 90-х годов персональные компьютеры получили возможность работать со звуковой информацией. Каждый компьютер, имеющий звуковую плату, может сохранять в виде файлов (файл - это определённое количество информации, хранящееся на диске и имеющее имя) и воспроизводить звуковую информацию. С помощью специальных программных средств (редакторов аудио файлов) открываются широкие возможности по созданию, редактированию и прослушиванию звуковых файлов. Создаются программы распознавания речи, и появляется возможность управления компьютером голосом.

Именно звуковая плата (карта) преобразует аналоговый сигнал в дискретную фонограмму и наоборот, «оцифрованный» звук – в аналоговый (непрерывный) сигнал, который поступает на вход динамика.

Схема записи звука:

звуковая волна
↓
микрофон
↓
переменный электрический ток
↓
аудиоадаптер (звуковая карта)
↓
двоичный код
↓
память ПК

Схема воспроизведения звука:

память ПК
↓
двоичный код
↓
аудиоадаптер
↓
переменный электрический ток
↓
динамик
↓
звуковая волна

При двоичном кодировании аналогового звукового сигнала непрерывный сигнал дискретизируется, т.е. заменяется серией его отдельных выборок - отсчётов. Качество двоичного кодирования зависит от двух параметров: количества дискретных уровней сигнала и количества выборок в секунду. Количество выборок или частота дискретизации в аудиоадаптерах бывает различной: 11 кГц, 22 кГц, 44,1 кГц и др. Если количество уровней равно 65536, то на один звуковой сигнал рассчитано 16 бит (216). 16-разрядный аудиоадаптер точнее кодирует и воспроизводит звук, чем 8-разрядный.

Количество бит, необходимое для кодирования одного уровня звука, называется глубиной звука. Объём моноаудиофайла (в байтах) определяется по формуле:

$$V_{моно} = \frac{\nu \cdot t \cdot G}{8},$$

где ν - частота дискретизации в Гц,
G – глубина звука в битах, t – время в секундах.

При стереофоническом звучании объём аудиофайла удваивается, при квадрофоническом звучании – учетверяется.

По мере усложнения программ и увеличения их функций, а также появления мультимедиа-приложений, растёт функциональный объём программ и данных. Если в середине 80-х годов обычный объём программ и данных составлял десятки и лишь иногда сотни килобайт, то в середине 90-х годов он стал составлять десятки мегабайт. Соответственно растёт объём оперативной памяти.

Пример решения: Подсчитать, сколько места будет занимать одна минута цифрового звука на жестком диске или любом другом цифровом носителе, записанного с частотой

а) 44.1 кГц;

б) 11 кГц;

в) 22 кГц;

г) 32 кГц

и разрядностью 16 бит.

Решение.

а) Если записывают моносигнал с частотой 44.1 кГц, разрядностью 16 бит (2 байта), то каждую минуту аналого-цифровой преобразователь будет выдавать 441000 * 2 * 60 = 529 000 байт (около 5 Мб) данных об амплитуде аналогового сигнала, который в компьютере записываются на жесткий диск.

Если записывают стереосигнал, то 1 058 000 байт (около 10 Мб).

Задания

1. Какой объем памяти требуется для хранения цифрового аудиофайла с записью звука высокого качества при условии, что время звучания составляет 3 минуты?

2. Какой объем данных имеет моноаудиофайл, длительность звучания которого 1 секунда, при среднем качестве звука (16 бит, 24 кГц)?

3. Рассчитайте объем стереоаудиофайла длительностью 20 секунд при 20-битном кодировании и частоте дискредитации 44.1 кГц. Варианты: 44,1 Mb, 4.21 Mb, 3,53 Mb.

4. Оцените информационный объем моноаудиофайла длительностью звучания 20 с, если "глубина" кодирования и частота дискретизации звукового сигнала равны соответственно 8 бит и 8 кГц;

5. Рассчитайте время звучания моноаудиофайла, если при 16-битном кодировании и частоте дискретизации 32 кГц его объем равен 700 Кбайт;

6. Запишите звуковой моноаудиофайл длительностью 20 с, с "глубиной" кодирования 8 бит и частотой дискретизации 8 кГц.

7. Определите качество звука (качество радиотрансляции, среднее качество, качество аудио-CD) если известно, что объем стериоаудиофайла длительностью звучания в 10 сек. Равен 940 Кбайт;

8. Оцените информационный объем стериоаудиофайла длительностью звучания 30 с, если "глубина" кодирования и частота дискретизации звукового сигнала равны соответственно 8 бит и 8 кГц;

9. Запишите звуковой файл длительностью 30с с "глубиной" кодирования 8бит и частотой дискретизации 8 кГц. Вычислите его объем и сверьтесь с полученным на практике значением.

10. Аналоговый звуковой сигнал был дискретизирован сначала с использованием 256 уровней интенсивности сигнала (качество звучания радиотрансляции), а затем с использованием 65536 уровней интенсивности сигнала (качество звучания аудио-CD). Во сколько раз различаются информационные объемы оцифрованного звука?

11. Оцените информационный объем моноаудиофайла длительностью звучания 1 мин. если "глубина" кодирования и частота дискретизации звукового сигнала равны соответственно:

16 бит и 48 кГц.

12. Запишите звуковой моноаудиофайл длительностью 1 минута с "глубиной" кодирования 16 бит и частотой дискретизации 48 кГц.

13. Подсчитать объем файла с 10 минутной речью записанного с частотой дискретизации 11025 Гц при 4 разрядном кодировании

14. Подсчитать время звучания звукового файла объемом 3.5 Мбайт содержащего стерео запись с частотой дискретизации 44100 Гц, 16-ти разрядном кодировании.

15. Определите количество уровней звукового сигнала при использовании 8-битных звуковых карт. Варианты: 256, 512,1024, 65 536.

16. Приведите пример:

а) аналогового способа представления звуковой информации;

б) дискретного способа представления звуковой информации.

17. Подготовить презентацию, демонстрирующую возможности звуковых форматов midi, wav, mp3, mod.

18. Перечислите параметры, от которых зависит качество двоичного кодирования звука.

Отчет

Отчет должен быть оформлен в текстовом редакторе и содержать:

— наименование работы;
— цель работы;
— задание;
— последовательность выполнения работы;
— ответы на контрольные вопросы;
— вывод о проделанной работе.

Контрольные вопросы

1. С какими звуковыми форматами вы встречаетесь чаще в повседневной жизни?
2. Дайте определение аудиоадаптеру?
3. Что значит оцифровка звука?

Практическое занятие №12

Декодирование информации

Цель работы: Ознакомление с правилами и получение практических навыков кодирования и декодирования информации.
Время выполнения: 2 часа
Оборудование: ПК.
Программное обеспечение: операционная система, графические редакторы.

Теоретические основы

Кодирование информации это преобразование формы представления информации с целью ее передачи или хранения. Кодирование это представление символов одного алфавита символами другого. Правила, по которым осуществляется кодирование называются кодом. Под словом понимают последовательность символов, количество символов в которой

называется длиной кода. Слова так же называют кодовыми комбинациями. Если при кодировании получают комбинации одинаковой длины, то такой код называют равномерным, а длину кодовых комбинаций в этом слове называют значимостью кода. Если кодовые комбинации различной длины, то код называется неравномерным.

Процесс обратный кодированию называется декодированием.

Если в коде ни одна более короткая комбинация не является началом более длинной кодовой комбинации, то код называется префиксным.

- кодирование – это перевод информации с одного языка на другой (запись в другой системе символов, в другом алфавите)
- обычно кодированием называют перевод информации с «человеческого» языка на формальный, например, в двоичный код, а декодированием – обратный переход
- один символ исходного сообщения может заменяться одним символом нового кода или несколькими символами, а может быть и наоборот – несколько символов исходного сообщения заменяются одним символом в новом коде (китайские иероглифы обозначают целые слова и понятия)
- кодирование может быть *равномерное* и *неравномерное*;
  при равномерном кодировании все символы кодируются кодами равной длины;
  при неравномерном кодировании разные символы могут кодироваться кодами разной длины, это затрудняет декодирование

Пример задания:

Для кодирования букв А, Б, В, Г решили использовать двухразрядные последовательные двоичные числа (от 00 до 11, соответственно). Если таким способом закодировать последовательность символов БАВГ и записать результат шестнадцатеричным кодом, то получится

1) $4B_{16}$ 2) 411_{16} 3) $BACD_{16}$ 4) 1023_{16}

Решение:

1) из условия коды букв такие: А – 00, Б –01, В – 10 и Г – 11, код равномерный
2) последовательность БАВГ кодируется так: 01 00 10 11 = 1001011
3) разобьем такую запись на тетрады справа налево и каждую тетраду переведем в шестнадцатеричную систему (то есть, сначала в десятичную, а потом заменим все числа от 10 до 15 на буквы A, B, C, D, E, F); получаем
$$1001011 = 0100\ \ 1011_2 = 4B_{16}$$
4) правильный ответ – 1.

Возможные ловушки:

- расчет на то, что при переводе тетрад в шестнадцатеричную систему можно забыть заменить большие числа (10–15) на буквы ($1011_2 = 11$, получаем неверный ответ 411_{16})
- может быть дан неверный ответ, в котором нужные цифры поменяли местами (расчет на невнимательность), например, $B4_{16}$
- в ответах дана последовательность, напоминающая исходную (неверный ответ $BACD_{16}$), чтобы сбить случайное угадывание

Пример задания:

Для 5 букв латинского алфавита заданы их двоичные коды (для некоторых букв – из двух бит, для некоторых – из трех). Эти коды представлены в таблице:

A	B	C	D	E
000	01	100	10	011

Определить, какой набор букв закодирован двоичной строкой 0110100011000

1) EBCEA 2) BDDEA 3) BDCEA 4) EBAEA

Решение (вариант 1, декодирование с начала):

1) здесь используется неравномерное кодирование, при котором декодирование может быть неоднозначным, то есть, заданному коду может соответствовать несколько разных исходных сообщений
2) попробуем декодировать с начала цепочки, первой буквой может быть B или E, эти случаи нужно рассматривать отдельно
3) пусть первая буква – E с кодом 011, тогда остается цепочка 0100011000

— для кода 0100011000 первой буквой может быть только B с кодом 01, тогда остается 00011000 (начало исходной цепочки – EB?)

— для кода 00011000 первой буквой может быть только A с кодом 000, тогда остается 11000, а эта цепочка не может быть разложена на заданные коды букв

— поэтому наше предположение о том, что первая буква – E, неверно

4) пусть первая буква – B с кодом 01, тогда остается цепочка 10100011000

— для кода 10100011000 первой буквой может быть только D с кодом 10, тогда остается 100011000 (можно полагать, что начало исходной цепочки – BD?)

— для кода 100011000 первой буквой может быть только C с кодом 100, тогда остается 011000 (начало исходной цепочки – BDC?)

Несмотря на то, что среди ответов есть единственная цепочка, которая начинается с BDC, здесь нельзя останавливаться, потому что «хвост» цепочки может «не сойтись»

— для кода 011000 на первом месте может быть B (код 01) или E (011); в первом случае «хвост» 1000 нельзя разбить на заданные коды букв, а во втором – остается код 000 (буква A), поэтому исходная цепочка может быть декодирована как BDCEA

5) правильный ответ – 3

Возможные ловушки и проблемы:

- при декодировании неравномерных кодов может быть очень много вариантов, их нужно рассмотреть все; это требует серьезных усилий и можно легко запутаться
- нельзя останавливаться, не закончив декодирование до конца и не убедившись, что все «сходится», на это обычно и рассчитаны неверные ответы

Решение (вариант 2, декодирование с конца):

1) для кода 0110100011000 последней буквой может быть только A (код 000), тогда остается цепочка 0110100011
2) для 0110100011 последней может быть только буква E (011), тогда остается цепочка 0110100
3) для 0110100 последней может быть только буква C (100), тогда остается цепочка 0110
4) для 0110 последней может быть только буква D (10), тогда остается 01 – это код буквы B
5) таким образом, получилась цепочка BDCEA
6) правильный ответ – 3

Возможные ловушки и проблемы:

- при декодировании неравномерных кодов может быть очень много вариантов (здесь *случайно* получилась единственно возможная цепочка), их нужно рассмотреть все; это требует серьезных усилий и можно легко запутаться
- нельзя останавливаться, не закончив декодирование до конца и не убедившись, что все «сходится», на это обычно и рассчитаны неверные ответы

Решение (вариант 3, кодирование ответов):

1) в данном случае самое простое и надежное – просто закодировать все ответы, используя приведенную таблицу кодов, а затем сравнить результаты с заданной цепочкой
2) получим
   1) EBCEA – 01101100011000 2) BDDEA – 011010011000
   3) BDCEA – 0110100011000 4) EBAEA – 01101000011000
3) сравнивая эти цепочки с заданной, находим, что правильный ответ – 3.

Возможные проблемы:

- сложно сравнивать длинные двоичные последовательности, поскольку они однородны, содержат много одинаковых нулей и единиц

Пример задания:

Для передачи по каналу связи сообщения, состоящего только из букв А, Б, В, Г, решили использовать неравномерный по длине код: А=0, Б=10, В=110. Как нужно закодировать букву Г, чтобы длина кода была минимальной и допускалось однозначное разбиение кодированного сообщения на буквы?

1) 1 2) 1110 3) 111 4) 11

Решение (вариант 1, метод подбора):

1) рассмотрим все варианты в порядке увеличения длины кода буквы Г
2) начнем с Г=1; при этом получается, что сообщение «10» может быть раскодировано двояко: как ГА или Б, поэтому этот вариант не подходит
3) следующий по длине вариант – Г=11; в этом случае сообщение «110» может быть раскодировано как ГА или В, поэтому этот вариант тоже не подходит
4) третий вариант, Г=111, дает однозначное раскодирование во всех сочетаниях букв, поэтому…
5) … правильный ответ – 3.

Возможные проблемы:

- при переборе можно ошибиться и «просмотреть» какой-нибудь вариант

Решение (вариант 2, «умный» метод):

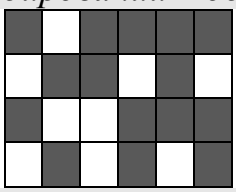
1) для того, чтобы сообщение, записанное с помощью неравномерного по длине кода, однозначно раскодировалось, требуется, чтобы никакой код не был началом другого (более длинного) кода; это условие называют *условием Фано*
2) как и в первом решении, рассматриваем варианты, начиная с самого короткого кода для буквы Г; в нашем случае код Г=1 является началом кодов букв Б и В, поэтому условие Фано не выполняется, такой код не подходит
3) код Г=11 также является началом другого кода (кода буквы В), поэтому это тоже ошибочный вариант
4) третий вариант кода, Г=111, не является началом никакого уже известного кода; кроме того, ни один уже имеющийся код не является началом кода 111; таким образом, условие Фано выполняется
5) поэтому правильный ответ – 3.

Возможные проблемы:

- нужно знать условие Фано

Пример задания:

Черно-белое растровое изображение кодируется построчно, начиная с левого верхнего угла и заканчивая в правом нижнем углу. При кодировании 1 обозначает черный цвет, а 0 – белый.

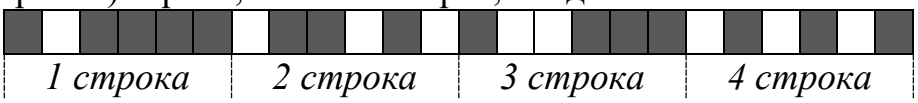


Для компактности результат записали в шестнадцатеричной системе счисления. Выберите правильную запись кода.

1) BD9AA5 2) BDA9B5 3) BDA9D5 4) DB9DAB

Решение:

1) «вытянем» растровое изображение в цепочку: сначала первая (верхняя) строка, потом – вторая, и т.д.:



2) в этой полоске 24 ячейки, черные заполним единицами, а белые – нулями:

1	*0*	*1*	*1*	*1*	*1*	*0*	***1***	***1***	*0*	***1***	*0*	***1***	*0*	*0*	***1***	***1***	***1***	*0*	***1***	*0*	***1***	*0*	***1***
1 строка						*2 строка*						*3 строка*						*4 строка*					

3) поскольку каждая цифра в шестнадцатеричной системе раскладывается ровно в 4 двоичных цифры, разобьем полоску на **тетрады** – группы из четырех ячеек (в данном случае все равно, откуда начинать разбивку, поскольку в полоске целое число тетрад – 6):

1	*0*	*1*	*1*	*1*	*1*	*0*	***1***	***1***	*0*	***1***	*0*	***1***	*0*	*0*	***1***	***1***	***1***	*0*	***1***	*0*	***1***	*0*	***1***

4) переводя тетрады в шестнадцатеричную систему, получаем последовательно цифры B (11), D(13), A(10), 9, D(13) и 5, то есть, цепочку BDA9D5

5) поэтому правильный ответ – 3.

Возможные проблемы:

- нужно уметь быстро переводить тетрады в шестнадцатеричные цифры (в крайнем случае, это можно сделать через десятичную систему)

Пример задания:

Для передачи чисел по каналу с помехами используется код проверки четности. Каждая его цифра записывается в двоичном представлении, с добавлением ведущих нулей до длины 4, и к получившейся последовательности дописывается сумма её элементов по модулю **2** *(например, если передаём* **23***, то получим последовательность* **0010100110***). Определите, какое число передавалось по каналу в виде* **01010100100111100011***?*

1) 59143 2) 5971 3) 102153 4) 10273

Решение:

1) сначала разберемся, как закодированы числа в примере; очевидно, что используется код равномерной длины; поскольку 2 знака кодируются 10 двоичными разрядами (битами), на каждую цифру отводится 5 бит, то есть

   $2 \rightarrow 00101$ и $3 \rightarrow 00110$

2) как следует из условия, четыре первых бита в каждой последовательности – это двоичный код цифры, а пятый бит (бит четности) используется для проверки и рассчитывается как «сумма по модулю два», то есть остаток от деления суммы битов на 2; тогда

   $2 = 0010_2$, бит четности $(0 + 0 + 1 + 0) \bmod 2 = 1$

   $3 = 0011_2$, бит четности $(0 + 0 + 1 + 1) \bmod 2 = 0$

3) но бит четности нам совсем **не нужен**, важно другое: пятый бит в каждой пятерке **можно отбросить**!
4) разобъем заданную последовательность на группы по 5 бит в каждой:

   01010, 10010, 01111, 00011.

5) отбросим пятый (последний) бит в каждой группе:

   0101, 1001, 0111, 0001.

   это и есть двоичные коды передаваемых чисел:

   $0101_2 = 5$, $1001_2 = 9$, $0111_2 = 7$, $0001_2 = 1$.

6) таким образом, были переданы числа 5, 9, 7, 1 или число 5971.
7) Ответ: 2.

Отчет

Отчет должен содержать:

— наименование работы;
— цель работы;
— задание;
— последовательность выполнения работы;
— ответы на контрольные вопросы;
— вывод о проделанной работе.

Контрольные вопросы

1. Где применяется кодирование информации?
2. Дать определение понятие декодер?
3. Какие существуют методы декодирования?

Практическое занятие № 13, 14

Шифрование с использованием симметричных методов

Цель работы: исследование простейших методов криптографической зашиты информации.
Время выполнения: 6 часа
Оборудование: ПК.
Программное обеспечение: операционная система, калькулятор.

Часть 1
Теоретические основы

Под ***конфиденциальностью*** понимают невозможность получения информации из преобразованного массива без знания дополнительной информации (ключа).

Аутентичность информации состоит в подлинности авторства и целостности.

Криптоанализ объединяет математические методы нарушения конфиденциальности и аутентичности информации без знания ключей.

Алфавит - конечное множество используемых для кодирования информации знаков.

Текст - упорядоченный набор из элементов алфавита. В качестве примеров алфавитов можно привести следующие:

алфавит Z_{33-32} буквы русского алфавита (исключая "ё") и пробел;

алфавит Z_{256} - символы, входящие в стандартные коды ASCII и КОИ-8;

двоичный алфавит - $Z_2 = \{0, 1\}$;

восьмеричный или шестнадцатеричный алфавит

Под ***шифром*** понимается совокупность обратимых преобразований множества открытых данных на множество зашифрованных данных, заданных алгоритмом криптографического преобразования. В шифре всегда различают два элемента: алгоритм и ключ. Алгоритм позволяет использовать сравнительно короткий ключ для шифрования сколь угодно большого текста.

Криптографическая система, или ***шифр*** представляет собой семейство ***T*** обратимых преобразований открытого текста в шифрованный. Членам этого семейства можно взаимно однозначно сопоставить число ***k***, называемое ***ключом.*** Преобразование ***Tk*** определяется соответствующим алгоритмом и значением ключа ***k***.

Ключ - конкретное секретное состояние некоторых параметров алгоритма криптографического преобразования данных, обеспечивающее выбор одного варианта из совокупности всевозможных для данного алгоритма. Секретность ключа должна обеспечивать невозможность

восстановления исходного текста по шифрованному.

Пространство ключей ***K*** - это набор возможных значений ключа.

Обычно ключ представляет собой последовательный ряд букв алфавита. Следует отличать понятия "ключ" и "пароль". ***Пароль*** также является секретной последовательностью букв алфавита, однако используется не для шифрования (как ключ), а для аутентификации субъектов.

Электронной (цифровой) подписью называется присоединяемое к тексту его криптографическое преобразование, которое позволяет при получении текста другим пользователем проверить авторство и целостность сообщения.

Зашифрованием данных называется процесс преобразования открытых данных в зашифрованные с помощью шифра, а ***расшифрованием*** данных - процесс преобразования закрытых данных в открытые с помощью шифра.

Дешифрованием называется процесс преобразования закрытых данных в открытые при неизвестном ключе и, возможно, неизвестном алгоритме, т.е. методами криптоанализа.

Шифрованием называется процесс зашифрования или расшифрования данных. Также термин шифрование используется как синоним зашифрования. Однако неверно в качестве синонима шифрования использовать термин "кодирование" (а вместо "шифра" - "код"), так как под кодированием обычно понимают представление информации в виде знаков (букв алфавита).

Криптостойкостью называется характеристика шифра, определяющая его стойкость к дешифрованию. Обычно эта характеристика определяется периодом времени, необходимым для дешифрования.

Криптология - это наука о преобразовании информации для обеспечения ее секретности, состоящая из двух ветвей: криптографии и криптоанализа.

Криптоанализ - наука (и практика ее применения) о методах и способах вскрытия шифров.

Криптография - наука о способах преобразования (шифрования) информации с целью ее защиты от незаконных пользователей. Исторически первой задачей криптографии была защита передаваемых текстовых сообщений от несанкционированного ознакомления с их содержанием, известного только отправителю и получателю, все методы шифрования являются лишь развитием этой философской идеи. С усложнением информационных взаимодействий в человеческом обществе возникли и продолжают возникать новые задачи по их защите, некоторые из них были решены в рамках криптографии, что потребовало развития новых подходов и

методов.

Шифры простой замены

Система шифрования Цезаря - частный случай шифра простой замены. Метод основан на замене каждой буквы сообщения на другую букву того же алфавита, путем смещения от исходной буквы на K букв.

Известная фраза Юлия Цезаря

VENI VI D I VICI, где

A	B	C	D	E	F	G	H	I	G	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

пришел, увидел, победил, зашифрованная с помощью данного метода, преобразуется в

SBKF SFAF SFZF

при смещении на 4 символа влево.

Греческим писателем Полибием за 100 лет до н.э. был изобретен так называемый **полибианский квадрат** размером 5*5, заполненный алфавитом в случайном порядке. Греческий алфавит имеет 24 буквы, а 25-м символом является пробел. Для шифрования на квадрате находили букву текста и записывали в зашифрованное сообщение букву, расположенную ниже ее в том же столбце. Если буква оказывалась в нижней строке таблицы, то брали верхнюю букву из того же столбца.

M	T	L	E	X
A	K	F	Q	Y
N	B	R	O	W
C	J	H	D	P
U	I	S	G	V

Схема шифрования Вижинера. Таблица Вижинера представляет собой квадратную матрицу с n^2 элементами, где n — число символов используемого алфавита. На рисунке показана верхняя часть таблицы Вижинера для кириллицы. Каждая строка получена циклическим сдвигом алфавита на символ. Для шифрования выбирается буквенный ключ, в соответствии с которым формируется рабочая матрица шифрования.

И т.д. до 33-ей строки..																															

Таблица Вижинера

Осуществляется это следующим образом. Из полной таблицы выбирается первая строка и те строки, первые буквы которых соответствуют буквам ключа. Первой размещается первая строка, а под нею — строки, соответствующие буквам ключа в порядке следования этих букв в ключе шифрования. Пример такой рабочей матрицы для ключа «книга» .

<u>Процесс шифрования осуществляется следующим образом:</u>

1. под каждой буквой шифруемого текста записываются буквы ключа. Ключ при этом повторяется необходимое число раз.

2. каждая буква шифруемого текста заменяется по подматрице буквами находящимися на пересечении линий, соединяющих буквы шифруемого текста в первой строке подматрицы и находящимися под ними букв ключа.

3. полученный текст может разбиваться на группы по несколько знаков.

Пусть, например, требуется зашифровать сообщение: *максимально допустимой ценой является пятьсот руб. за штуку*. В соответствии с первым правилом записываем под буквами шифруемого текста буквы ключа. Получаем:

максимально допустимой ценой является пятьсот руб. за штуку

книгакнигак нигакнигак нигак нигакниг акнигак ниг ак нигак

Дальше осуществляется непосредственное шифрование в соответствии со вторым правилом, а именно: берем первую букву шифруемого текста (М) и соответствующую ей букву ключа (К); по букве шифруемого текста (М) входим в рабочую матрицу шифрования и выбираем под ней букву, расположенную в строке, соответствующей букве ключа (К),— в нашем примере такой буквой является Ч; выбранную таким образом букву

помещаем в зашифрованный текст. Эта процедура циклически повторяется до зашифрования всего текста.

Эксперименты показали, что при использовании такого метода статистические характеристики исходного текста практически не проявляются в зашифрованном сообщении. Нетрудно видеть, что замена по таблице Вижинера эквивалентна простой замене с циклическим изменением алфавита, т.е. здесь мы имеем полиалфавитную подстановку, причем число используемых алфавитов определяется числом букв в слове ключа. Поэтому стойкость такой замены определяется произведением стойкости прямой замены на число используемых алфавитов, т.е. число букв в ключе.

Расшифровка текста производится в следующей последовательности:

1. над буквами зашифрованного текста последовательно надписываются буквы ключа, причем ключ повторяется необходимое число раз.
2. в строке подматрицы Вижинера, соответствующей букве ключа отыскивается буква, соответствующая знаку зашифрованного текста. Находящаяся под ней буква первой строки подматрицы и будет буквой исходного текста.
3. полученный текст группируется в слова по смыслу.

Нетрудно видеть, что процедуры как прямого, так и обратного преобразования являются строго формальными, что позволяет реализовать их алгоритмически. Более того, обе процедуры легко реализуются по одному и тому же алгоритму.

Одним из недостатков шифрования по таблице Вижинера является то, что при небольшой длине ключа надежность шифрования остается невысокой, а формирование длинных ключей сопряжено с трудностями.

Нецелесообразно выбирать ключи с повторяющимися буквами, так как при этом стойкость шифра не возрастает. В то же время ключ должен легко запоминаться, чтобы его можно было не записывать. Последовательность же букв не имеющих смысла, запомнить трудно.

С целью повышения стойкости шифрования можно использовать усовершенствованные варианты таблицы Вижинера. Приведем только некоторые из них:

- во всех (кроме первой) строках таблицы буквы располагаются в произвольном порядке.
- В качестве ключа используется случайность последовательных чисел. Из таблицы Вижинера выбираются десять произвольных строк, которые кодируются натуральными числами от 0 до 10. Эти строки используются в соответствии с чередованием цифр в выбранном ключе.

Известны также и многие другие модификации метода.

Алгоритм перестановки

Этот метод заключается в том, что символы шифруемого текста переставляются по определенным правилам внутри шифруемого блока символов. Рассмотрим некоторые разновидности этого метода, которые могут быть использованы в автоматизированных системах.

Самая простая перестановка — написать исходный текст задом наперед и одновременно разбить шифрограмму на пятерки букв. Например, из фразы

ПУСТЬ БУДЕТ ТАК, КАК МЫ ХОТЕЛИ.

получится такой шифротекст:

ИЛЕТО ХЫМКА ККАТТ ЕДУБЪ ТСУП

В последней группе (пятерке) не хватает одной буквы. Значит, прежде чем шифровать исходное выражение, следует его дополнить незначащей буквой

(например, О) до числа, кратного пяти:

ПУСТЬ-БУДЕТ-ТАККА-КМЫХО-ТЕЛИО.

Тогда шифрограмма, несмотря на столь незначительные изменения, будет выглядеть по-другому:

ОИЛЕТ ОХЫМК АККАТ ТЕДУБ ЬТСУП

Кажется, ничего сложного, но при расшифровке проявляются серьезные неудобства.

Во время Гражданской войны в США в ходу был такой шифр: исходную фразу писали в несколько строк. Например, по пятнадцать букв в каждой (с заполнением последней строки незначащими буквами).

П У С Т Ь Б У Д Е Т Т А К К А

К М Ы Х О Т Е Л И К Л М Н О П

После этого вертикальные столбцы по порядку писали в строку с разбивкой на пятерки букв:

ПКУМС ЫТХЬО БТУЕД ЛЕИТК ТЛАМК НКОАП

Если строки укоротить, а количество строк увеличить, то получится прямоугольник-решетка, в который можно записывать исходный текст. Но тут уже потребуется предварительная договоренность между адресатом и отправителем посланий, поскольку сама решетка может быть различной длины-высоты, записывать к нее можно по строкам, по столбцам, по спирали

туда или по спирали обратно, можно писать и по диагоналями, а для шифрования можно брать тоже различные направления.

Шифры сложной замены

Шифр Гронсфельда состоит в модификации шифра Цезаря числовым ключом. Для этого под буквами сообщения записывают цифры числового ключа. Если ключ короче сообщения, то его запись циклически повторяют. Зашифрованное сообщение получают примерно также, как в шифре Цезаря, но используют не одно жестко заданное смещение а фрагменты ключа.

Пусть в качестве ключа используется группа из трех цифр – 314, тогда сообщение

С О В Е Р Ш Е Н Н О С Е К Р Е Т Н О

3 1 4 3 1 4 3 1 4 3 1 4 3 1 4 3 1 4

Ф П Ё С Ь З О С С А Х З Л Ф З У С С

В **шифрах многоалфавитной замены** для шифрования каждого символа исходного сообщения применяется свой шифр простой замены (свой алфавит).

	АБВГДЕЁЖЗИКЛМНОПРСТУФХЧШЩЪЫЬЭЮЯ_
А	АБВГДЕЁЖЗИКЛМНОПРСТУФХЧШЩЪЫЬЭЮЯ_
Б	_АБВГДЕЁЖЗИКЛМНОПРСТУФХЧШЩЪЫЬЭЮЯ
В	Я_АБВГДЕЁЖЗИКЛМНОПРСТУФХЧШЩЪЫЬЭЮ
Г	ЮЯ_АБВГДЕЁЖЗИКЛМНОПРСТУФХЧШЩЪЫЬЭ
.	……
Я	ВГДЕЁЖЗИКЛМНОПРСТУФХЧШЩЪЫЬЭЮЯ_АБ
_	БВГДЕЁЖЗИКЛМНОПРСТУФХЧШЩЪЫЬЭЮЯ_А

Каждая строка в этой таблице соответствует одному шифру замены аналогично шифру Цезаря для алфавита, дополненного пробелом. При шифровании сообщения его выписывают в строку, а под ним ключ. Если ключ оказался короче сообщения, то его циклически повторяют. Зашифрованное сообщение получают, находя символ в колонке таблицы по букве текста и строке, соответствующей букве ключа. Например, используя ключ АГАВА, из сообщения ПРИЕЗЖАЮ ШЕСТОГО получаем следующую шифровку:

ПРИЕЗЖАЮ_ШЕСТОГО

АГАВААГАВААГАВАА

ПОИГЗЖЮЮЮШЕПТНГО

Такая операция соответствует сложению кодов ASCII символов сообщения и ключа по модулю 256.

Практическое задание

Придумайте 3 фразы, каждая минимум из 7 слов. Реализуйте шифрование этой фразы всеми перечисленными видами шифрования.

Часть 2

«Практика криптографической защиты информации»

Введение

Криптография - наука о методах преобразования информации с целью ее защиты от незаконных пользователей.

Стеганография - набор средств и методов сокрытия факта передачи информации.

Некоторые методы стеганографии:

1. В древности голову раба брили, на коже головы писали сообщение и, после отрастания волос, раба отправляли к адресанту.
2. Скрытое письмо между строк: молоком, апельсиновым (или лимонным) соком, другими химическими веществами.
3. "Микроточка". Сообщение с помощью современной технологии записывается на очень маленький носитель ("микроточку"), которая пересылается адресату, например, под обычной маркой.
4. Акростих - первые буквы слов стихотворения несут информацию:
   Добрый удод наелся ягод,
   Умный удод наелся на год.
   Наелся удод и песни поет.
   Ягод наелся удод.
   Задание 1: Придумать акростих, в котором скрыто ваше имя.
5. Например, каждое четвертое слово в посылаемом сообщении несет информацию (остальные слова ничего не значат). Пример: "Тридцать первого августа встреча судебного совета округа состоялась. Подтвердите дату следующего как можно скорее. Участники договорились собраться там же. Борис."
   Задание 2: Придумать подобное послание.

Разновидности шифров

1. Шифр замены. Каждая буква заменяется на определенный символ или последовательность символов. Пример: "Пляшущие человечки" Конан Дойля.
2. Шифр перестановки. Буквы в передаваемом сообщении меняются местами в соответствии с определенным правилом. Примеры: МАМА - АМАМ. КРИПТОГРАФИЯ - ИПКРГРТОИЯАФ. (ИП□ КР ГР□ ТО ИЯ□ АФ)

3. Книжный шифр. В шифрованном тексте каждое слово заменено на пару чисел номер страницы в книге и номер этого слова на странице. (т.е. текст выглядит примерно так: 3-45 45-67 …).

Ключ - сменный элемент шифра, который применен для шифрования конкретного сообщения.

Шифры перестановки

Маршрутная транспозиция

1. Т - дополнительная буква.

В	О	С	К	Р	Е
А	М	Я	А	Н	С
Т	Е	М	А	Т	И
Я	А	К	С	Е	Ч
Ш	К	О	Л	А	Т

Фраза "Воскресная математическая школа" становится: "ЕСИЧТ АЕТНР КААСЛ ОКМЯС ОМЕАК ШЯТАВ".
Ключ - число 6.

Задание 3:

1. Зашифровать:
   a) Французский математик Пьер Ферма по образованию был юрист.
   b) Леонардо Пизанского математики знают под именем "сын добряка" или Фибоначчи.
2. Дешифровать (восстановить сообщение, зная ключ) Ключ 8.
   Чинои сечем лчгмс хыеоо еаитн ккыин лтсбч втрйы еоосс ееорс неомв бадер покп.
   Примечание: АБ-дополнительные буквы.
3. Расшифровать (восстановить сообщение, не зная ключа).
   Осуз уаан евем исчи тдьм одоа ьлвь рдво быи.
4. Расшифровать:
   Етгртуой дкмиуиав цлишлаег врныинис аяоплыдб аанполбр.

2. Ключом является правило расстановки.

В	О	Е	С	М	А	Л	А
С	Р	Н	Е	Т	О		
К	А	Т	И	К			
Я	А	Ч	Ш				

Воесмала срнето катик яачш мея са к.
Или
Воес мала срне тока тикя ачшм еяса к.

М Е Я

С А

К

Задание 4:

1. Расшифровать фразу: Сошки ввнлы охеде нванз бркое еуквс изазх.
2. Расшифровать фразу: Леор тиюд тнет мауа ялее очнм кжхо йчей ооот лсеч и_пчс днит _киех са_чл илж_а шоо_в рп_уо_ к_ _

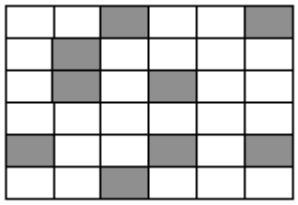
Постолбцовая транспозиция

К	А	Ш	А	Лшше ссис пссл шшао соак ааои ау. Над столбцами записывается ключевое слово, затем в соответствии с порядком букв в алфавите столбцы нумеруются, а затем выписываются подряд: первый столбец, за ним второй и т.д. Ключ здесь - "каша".
3	1	4	2	
Ш	Л	А	С	
А	Ш	А	П	
О	Ш	О	С	
С	Е	И	С	
О	С	А	Л	
А	С	У	Ш	
К	И			

Задание 5:

1. Зашифруйте фразу: Не плюй в колодец: вылетит - не поймаешь.
2. Дешифруйте старинное японское хайку: (ключом будет имя известного японского поэта "Басё") Тйдг адга лвис ыуы лояк пкшр ррув лшсс иеап пнву увет н.
3. Расшифруйте высказывание Козьмы Пруткова: Ако еаь дне дсц тан жод сск даг рео о.

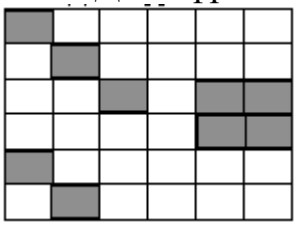
Шифр "Решетка"



Если хочешь быть красивым поступи в гусары. (Высказывание Козьмы Пруткова)
Пьеиис влыбым тивхгь укпрос оарчсе ташуыс.
Ключом является решетка

Задание 6:

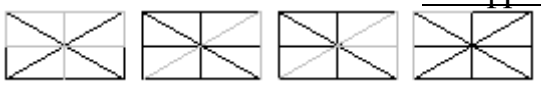
1. Придумать решетку и зашифровать фразу:
   "Евклид был древнегреческим математиком"
2. Дешифровать:



а) Сиекпе тпароо коолко йслтйс тськво нвоски.
б) Двплго раимид рувтай гуекте гньдот ыоруам.

Шифры замены

Шифр "Британский флаг"



Ответ: флаг.
Задание 7: зашифровать фразу: "ВГГУ".

Каждая буква заменяется несколькими символами

Криптография	11179161915417121932
	К1К7А9К6К9К5А4К7А1Ф1А9Ю2
	И2О2ИО1О4ОА3О2АУ1ИЯ

Задание 8:

1. Расшифровать: 1111712 20 111211728 201117112 11517112121228
2. Расшифровать: 11212941191517 16122812 1615 176116 111514415
3. Расшифровать: 11176191832 5157529 1815291716191832

4. Расшифровать: И1А2А5И5О4 Е3И5О4Я1О2 Я1И5О4Е3И5 И6О1И6О1О5 И3Ю1О2И5У8 И1А3И6И3Е3 А2О5А4О3И2 Е3И1О2А5Е1 Е3О3О3А5О2.

Шифры "Пляшущие человечки"

Основной метод расшифровки подобных шифров - частотный анализ. (+ логические рассуждения).
Таблица частот:
В русском языке в каждой тысяче символов в среднем встречается

пробел	175	р	40	я	18	х	9
о	90	в	38	з	16	ж	7
е, ё	72	л	35	ы	16	ш	6
а	62	к	28	б	14	ю	6
и	62	м	26	ъ, ь	14	ц	4
н	53	д	25	г	13	э	3
т	53	п	23	ч	12	щ	3
с	45	у	21	й	10	ф	2

Чаще всего буквы заменяют другими буквами.
Задание 9: расшифровать текст:
Сзргйзю тсуцълн Уйиефнлм цкрго, ъхс зов кргнспфхег ф зиецынсм ргзс тсзсмхл, ритулрцйзиррс тсжсесулхя с тсжзи л тсфои ахсжс туизфхгелхяфв. Рг сзрсм лк тусжцосн ср тсефхуиъго жцовбьцб ф дсосрнсм зиецынц. Тсуцълн тсзсыио н рим, трцо ии дсосрнц хгн, ъхс хг згоинс цоихиог л фнгкго:
- Рлкнс оихлх. Елзгхя, н зсйзб. Нфхгхл, угкуиылхи туизфхгелхяфв, тсуцълн Уйиефнлм.
С – 37
И - 21
Г-18
Н – 18
Л - 18
З – 17
Р -16
Ц - 14
Т – 14
Ф – 13
У – 13

Е – 10
М - 7
Й - 5
Я -3
А – 1
Ю – 1

Шифр Цезаря.

В шифре Цезаря каждая буква заменяется на букву, которая идет через 3 после этой: т.е. А=>Г, О=>С, Я=>В.

Примечание: можно делать сдвиг не на три, а на произвольное количество букв.

Задание 10: расшифровать текст

> Сзргйзю тсуцълн Уйиефнлм цкрго, ъхс зов кргнспфхег ф зиецынсм ргзс тсзсмхл, ритулрцйзиррс тсжсесулхя с тсжсзи л тсфои ахсжс туизфхгелхяфв. Рг сзрсм лк тусжцосн ср тсефхуиъго жцовбьцб ф дсосрнсм зиецынц. Тсуцълн тсзсыио н рим, трцо ии дсосрнц хгн, ъхс хг згоинс цоихиог л фнгкго:
> - Рлкнс оихлх. Елзгхя, н зсйзб. Нфхгхл, угкуиылхи туизфхгелхяфв, тсуцълн Уйиефнлм.

Задание 11: зашифровать фразу: "Идет занятие по криптографии".

Ключом является сдвиг.

Метод полосок: берутся полоски, прикладываются и в определенном месте читается слово. Например, слово НГЫГ легко расшифровать. Получается слово КАША.

Задание 12: расшифровать фразу: Схсоябхфв нсыни пюынлрю фоикнл.

Шифр Виженера.

Ключ ВАЗА: /3 1 8 1/

Сдвиг осуществляется не на постоянную величину, а на номер буквы в ключевом слове. КРИПТОГРАФИЯ => НСРРХПЛСГХРА.

Сложность при расшифровке в том, что одинаковые буквы переходят в разные, а разные - в одинаковые => частотный анализ не применим.

Задание 13:

Зашифровать фразу: Математика - царица наук.

Отчет

Отчет должен содержать:

— наименование работы;
— цель работы;
— задание;
— последовательность выполнения работы;
— ответы на контрольные вопросы;
— вывод о проделанной работе.

Контрольные вопросы

1. Где применяется криптография?
2. Какой смысл в Шифре Гронсфельда?
3. С помощью системы шифрования Цезаря зашифровать свое имя?

Практическое занятие №15,16

Шифрование ассиметричными методами

Цель работы

Изучить принцип работы асимметричных алгоритмов шифрования на примере алгоритма RSA. Освоить методику создания комбинированных алгоритмов шифрования, которые совмещают достоинства методов симметричной и асимметричной криптографии.

Алгоритм шифрования RSA

Алгоритм RSA был разработан в 1977 году Роном Ривестом, Ади Шамиром и Леном Адле- маном и опубликован в 1978 году. С тех пор алгоритм ***Rivest-Shamir-Adleman*** (RSA) широко применяется практически во всех приложениях, использующих криптографию с открытым ключом [1, 2, 3].

Алгоритм RSA состоит из трёх этапов:

I. ***Вычисление ключей.*** Важным моментом в этом криптоалгоритме является создание пары ключей: открытого и закрытого. Для алгоритма RSA этап создания ключей состоит из следующих операций:
   1. Выбираются два простых различных числа и . Вычисляется их произведение q, называемое ***модулем.***
   2. Вычисляется функция Эйлера p (п) = (p — 1) • (q — 1).
   3. Выбирается произвольное число ***e*** (e < п) такое, что 1 < ***e*** < p(п) и ***e*** не имеет с числом p(п) других общих делителей, кроме 1 (т.е. оно является взаимно простым с ним).
   4. Вычисляется d (алгоритмом Евклида) таким образом, что (e • ***d — 1)*** делилось на p(п).
   5. Два числа (e , п) публикуются как ***открытый ключ.***
   6. Число d хранится в секрете. Пара (d , п) есть ***закрытый ключ,*** который позволит читать

      все послания, зашифрованные с помощью пары чисел ( ).

II. ***Шифрование*** с помощью этих ключей производится следующим образом:
   1. Отправитель разбивает своё сообщение M на блоки ***m***j. Значение m j < п, поэтому длина блока открытого текста m j в битах не больше k = [1 o g_2 п] бит,

где квадратные скобки обозначают взятие целой части от дробного числа. Например, если п = 2 1 , то максимальная длина блока открытого текста k = [1 o g_2 2 1] = [4.3 9 ...] = 4 бита.

2. Подобный блок может быть интерпретирован как число из диапазона ( ). Для каждого такого числа m j вычисляется выражение (***c*** j - зашифрованное сообщение):

$$q = (m_j)^e \text{ т o d п.}$$

В качестве размера блока зашифрованного текста следует брать k_e = [1 o g_2 п] бит, где операция [ ] - это округление вверх до ближайшего целого.

Необходимо добавлять нулевые биты слева в двоичное представление блока q до размера бит.

III. ***Дешифрование*** производится следующим образом:

1. Чтобы получить открытый текст, надо каждый блок зашифрованного текста длиной k_e бит дешифровать отдельно:

$$m_j = (q)^d \text{ т o d п.}$$

Пример:

Выбрать два простых числа , .

Вычислить .

Вычислить ( ) ( ) ( ) .

Выбрать так, чтобы было взаимно простым с ( ) и меньше, чем ( ): .

Определить так, чтобы и : , т.к. .

Результирующие ключи: открытый ключ (5 , 1 1 9) и закрытый ключ (7 7 , 1 1 9). Пусть, например, требуется зашифровать сообщение M = 1 9:

$$C = 19^5 = 66 \text{ (т o d 1 1 9).}$$

Для дешифрования вычисляется $6\ 6^{77}$ (т o d 1 1 9) = 1 9.

Комбинирование симметричных и асимметричных алгоритмов

Симметричные алгоритмы и, в частности, DES - быстрые, поэтому ими удобно шифровать большие объёмы информации. Однако для передачи ключа симметричного алгоритма требуется надёжный канал передачи, который очень часто отсутствует. Таким образом, преимущества таких алгоритмов сводятся на нет. С другой стороны, асимметричные алгоритмы не требуют секретного канала для передачи ключа, но на практике криптосистемы с открытым ключом используются для шифрования не сообщений, а ключей. На это есть две основные причины:

1. Алгоритмы шифрования с открытым ключом в среднем работают в тысячи раз медленнее, чем симметричные алгоритмы, а также они требовательны к памяти и вычислительной мощности компьютера, поэтому большие тексты кодировать этими алгоритмами нецелесообразно.
2. Алгоритмы шифрования с открытым ключом уязвимы по отношению к криптоаналитическим атакам со знанием открытого текста. Пусть C = £(P), где ***C*** обозначает шифртекст, ***P*** - открытый текст, - функцию шифрования. Тогда, если принимает значения из некоторого конечного множества, состоящего из п открытых текстов, криптоаналитику достаточно зашифровать все эти тексты, используя известный ему открытый ключ, и сравнить результаты с C. Ключ таким способом ему вскрыть не удастся, однако открытый текст будет успешно определён.

Возможно следующее решение: сообщение шифруется симметричным алгоритмом, что позволяет выиграть в скорости, т.к. сообщение может быть сколь угодно большим, а ключ симметричного алгоритма (обычно маленький, для DES - 56 бит) шифруется

асимметричным алгоритмом [1].

Задание

Результатом данной лабораторной работы должны стать приложения, совмещающие в себе достоинства симметричных и асимметричных методов шифрования.

I. Реализовать приложение для шифрования, позволяющее выполнять следующие действия:

1. Вычислять открытый и закрытый ключи для алгоритма RSA:
   1) числа p и q генерируются программой или задаются из файла;
   2) числа p и q должны быть больше, чем 2^{128};
   3) сгенерированные ключи сохраняются в файлы: открытый ключ (е, п) - в один файл, закрытый ( ) - в другой.
2. Шифровать указанным в варианте симметричным алгоритмом открытый текст, а асимметричным - ключ симметричного алгоритма:
   1) шифруемый текст должен храниться в одном файле, открытый ключ ( ) для алгоритма RSA - в другом;
   2) ключ для симметричного алгоритма должен генерироваться случайным образом;
   3) зашифрованный текст должен сохраняться в одном файле, а зашифрованный асимметричным алгоритмом ключ симметричного алгоритма - в другом;
   4) в процессе шифрования предусмотреть возможность просмотра и изменения шифруемого текста в шестнадцатеричном и символьном виде;
   5) программа должна уметь работать с текстом произвольной длины.

II. Реализовать приложение для дешифрования.

1. Зашифрованный текст должен храниться в одном файле, зашифрованный ключ симметричного алгоритма - в другом, а секретный ключ для алгоритма RSA - в третьем.
2. Приложение расшифровывает зашифрованный ключ К ***с*** помощью алгоритма RSA, а затем с помощью симметричного алгоритма с ключом расшифровывает зашифрованный текст.
3. Расшифрованный текст должен сохраняться в файл.
4. В процессе дешифрования предусмотреть возможность просмотра и изменения зашифрованного текста в шестнадцатеричном и символьном виде.
5. Программа должна уметь работать с текстом произвольной длины.

III. С помощью реализованных приложений выполнить следующие задания.

1. Протестировать правильность работы разработанных приложений.
2. Сделать выводы о проделанной работе.

Дополнительные критерии оценивания качества работы

1. Наглядность приложений:

1 - приложения позволяют просматривать и изменять ключи, шифруемый и зашифрованный тексты во всех предусмотренных заданием представлениях;

0 - приложения позволяют просматривать ключи, шифруемый и зашифрованный тексты только в каком-то одном представлении; ***л.р. не принимается*** - иначе.

Студент с нечётным номером в качестве симметричного алгоритма должны использовать алгоритм DES, а студент с чётным - ГОСТ.

Контрольные вопросы

I. **Первая часть защиты** (обязательная):

1. В чём заключается алгоритм RSA?
2. Для чего и почему используют комбинированные криптоалгоритмы?
3. В чём заключаются достоинства и недостатки асимметричных алгоритмов?
4. В чём заключаются достоинства и недостатки симметричных алгоритмов?

II. **Вторая часть защиты**: Найти алгоритмом Евклида элемент d такой, что ***e • d*** = 1 (m o d п), если:

1. 5, ; 3. , ; 5. , ;
2. e = 5 8, п = 1 1 5; 4. e = 2 4, п = 9 5 ; 6. e = 1 8, п = 1 0 7.

Список литературы

1 Мао, В. Современная криптография: теория и практика : Пер. с англ. / В. Мао. - М. : Издательский дом "Вильямс", 2005. - 768 с.

2 Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. - М. : Триумф, 2002. - 816 с.

3 PKCS #1 v2.1: RSA Cryptography Standard. - Bedford : RSA Laboratories, 2002. - 61 p.