**Лабораторная работа №1**

**Измерение информации**

**1. Содержательный подход**

Количество информации, заключенное в сообщении, определяется объемом знаний, которое это сообщение несет получающему его человеку.

*Сообщение содержит информацию для человека, если заключенные в нем сведения являются для этого человека* ***новыми*** *и* ***понятными*** *и, следовательно, пополняют его знания.*

При содержательном подходе возможна качественная оценка информации: полезная, важная, …

Единица измерения количества информации называется битом. *Сообщение, уменьшающее неопределенность знания человека в два раза, несет для него 1 бит информации.*

Пусть в некотором сообщении содержатся сведения о том, что произошло одно из N равновероятных событий. Тогда количество информации, заключенное в этом сообщении:

*.*

**2. Алфавитный подход**

Алфавитный подход к измерению информации позволяет определить количество информации, заключенной в тексте. Такой подход является объективным, т.е. он не зависит от субъекта (человека), воспринимающего текст.

Множество символов, используемых при записи текста, называется алфавитом. Полное количество символов в алфавите называется мощностью (размером). Если допустить, что все символы алфавита встречаются в тексте с одинаковой частотой (вероятностью), то количество информации, которое несет каждый символ, вычисляется по формуле:

,

где N – мощность алфавита.

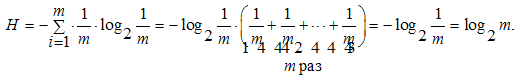
3. **Вероятностный подход к измерению информации**

Формулу для вычисления количества информации, учитывающую *неодинаковую вероятность* событий, предложил К. Шеннон в 1948 году. Количественная зависимость между вероятностью события *р* и количеством информации в сообщении о нем *x* выражается формулой: x=log2 (1/p). Качественную связь между вероятностью события и количеством информации в сообщении об этом событии можно выразить следующим образом - чем меньше вероятность некоторого события, тем больше информации содержит сообщение об этом событии.

Рассмотрим некоторую ситуацию. В коробке имеется 50 шаров. Из них 40 белых и 10 черных. Очевидно, вероятность того, что при вытаскивании "не глядя" попадется белый шар больше, чем вероятность попадания черного. Можно сделать заключение о вероятности события, которые интуитивно понятны. Проведем количественную оценку вероятности для каждой ситуации. Обозначим pч - вероятность попадания при вытаскивании черного шара, рб - вероятность попадания белого шара. Тогда: рч=10/50=0,2; рб40/50=0,8. Заметим, что вероятность попадания белого шара в 4 раза больше, чем черного. Делаем вывод: если *N* - это общее число возможных исходов какого-то процесса (вытаскивание шара), и из них интересующее нас событие (вытаскивание белого шара) может произойти *K* раз, то вероятность этого события равна *K/N*. Вероятность выражается в долях единицы. Вероятность достоверного события равна 1 (из 50 белых шаров вытащен белый шар). Вероятность невозможного события равна нулю (из 50 белых шаров вытащен черный шар).

Количественная зависимость между вероятностью события *р* и количеством информации в сообщении о нем x выражается формулой: . В задаче о шарах количество информации в сообщении о попадании белого шара и черного шара получится: ; 

Рассмотрим некоторый алфавит из *m* символов: и вероятность выбора из этого алфавита какой-то *i*-й буквы для описания (кодирования) некоторого состояния объекта. Каждый такой выбор уменьшит степень неопределенности в сведениях об объекте и, следовательно, увеличит количество информации о нем. Для определения среднего значения количества информации, приходящейся в данном случае на один символ алфавита, применяется формула . В случае *равновероятных* выборов *p=1/m*. Подставляя это значение в исходное равенство, мы получим

****

Рассмотрим следующий пример. Пусть при бросании несимметричной четырехгранной пирамидки вероятности выпадения граней будут следующими: p1=1/2, p2=1/4, p3=1/8, p4=1/8, тогда количество информации, получаемое после броска, можно рассчитать по формуле:

**http://www.ido.rudn.ru/nfpk/inf/f6.gif**

Для симметричной четырехгранной пирамидки количество информации будет: *H=log24=2(бит)*.

Заметим, что для симметричной пирамидки количество информации оказалось больше, чем для несимметричной пирамидки. Максимальное значение количества информации достигается для равновероятных событий.

**Задание к лабораторной работе**

Разработать приложение, реализующее алфавитный подход к измерению информации:

1. Алфавит языка находится в отдельном файле (его можно модифицировать). Мощность определяется автоматически.

2. Сообщение задается пользователем.

3. Дополнительно предусмотрены измерения информации для русского и английского языка.

4. Обоснование выбора языка програмирования.

При защите лабораторной работы необходимо решить задачу на один из подходов к измерению информации.

**Задачи по теме "Алфавитный подход к измерению информации".**

6. Световое табло состоит из лампочек, каждая из которых может находиться в двух состояниях («включена» или «выключена»). Какое наименьшее количество лампочек должно находиться на табло, чтобы с его помощью можно было передать 200 различных сигналов?

7. Эллочка-людоедка (в лексиконе которой, как известно, было 30 слов) произносит фразу, состоящую из 50 слов. Какое количество информации в битах сообщает Эллочка?

**Более сложные задачи по теме "Алфавитный подход к измерению информации".**

8. В велокроссе участвуют 119 спортсменов. Специальное устройство регистрирует прохождение каждым из участников промежуточного финиша, записывая его номер с использованием минимально возможного количества бит, одинакового для каждого спортсмена. Каков информационный объем в битах сообщения, записанного устройством, после того как промежуточный финиш прошли 70 велосипедистов?

9. Репетиционный экзамен в школе сдают 125 человек. Каждому из них выделяют специальный номер. При регистрации участника для записи его номера используют минимально возможное количество бит, одинаковое для каждого участника. Каков объем информации в битах, записанный устройством, после регистрации 60 участников?

10. Для передачи секретного сообщения используют код, состоящий из десятичных цифр. При этом все цифры кодируются одним и тем же (минимально возможным) количеством бит. Определите информационный объем в битах такого сообщения длиной в 150 символов.

11. Метеорологическая станция ведет наблюдение за влажностью воздуха. Результатом одного измерения является целое число от 0 до 100 процентов, которое записывается при помощи минимального возможного количества бит. Станция сделала 80 измерений. Определите информационный объем в битах результатов измерения.

12. Для записи результатов детской игры «Зарница» используется таблица, в каждой клетке которой записано либо количество баллов, полученных командой в соответствующем виде состязаний (1, 2, 3), либо прочерк (если команда в этом виде соревнований не участвовала). В «Зарнице» соревнуются 30 команд в 10 видах соревнований. Какое количество информации в битах содержит таблица?

23. Световое табло состоит из лампочек. Каждая лампочка может находиться в одном из трех состояний («включено», «выключено» или «мигает»). Какое наименьшее количество ламп должно находиться на табло, чтобы с его помощью можно было передать 27 различных сигналов?

24. Азбука Морзе позволяет кодировать символы для радиосвязи, задавая комбинацию точек и тире. Сколько различных символов можно закодировать, используя код Морзе длиной не менее пяти и не более шести сигналов?

25. Вася и Петя передают друг другу сообщения, используя синий, красный и зеленый фонарики. Это они делают, включая по одному фонарику на одинаково короткое время в некоторой последовательности. Количество вспышек в одном сообщении – 3 или 4. Между сообщениями – паузы. Сколько различных сообщений могут передавать мальчики?

26. Для передачи 300 различных сообщений используют 5 последовательных цветовых вспышек. Цветовые лампы включаются на одинаково короткое время в некоторой последовательности. Лампы скольких различных цветов должно использоваться при передаче (минимальное количество)?

27. Для передачи 1000 различных сообщений используют 5 последовательных цветовых вспышек. Цветовые лампы включаются на одинаково короткое время в некоторой последовательности. Лампы скольких различных цветов должно использоваться при передаче (минимальное количество)?

**Лабораторная работа №2**

**Алгоритмы архивации без потерь. RLE.**

**Алгоритм RLE**

**Первый вариант алгоритма**

Данный алгоритм необычайно прост в реализации. Групповое кодирование — от английского Run Length Encoding (RLE) — один из самых старых и самых простых алгоритмов архивации графики. Изображение в нем (как и в нескольких алгоритмах, описанных ниже) вытягивается в цепочку байт по строкам растра. Само сжатие в RLE происходит **за счет того, что в исходном изображении встречаются цепочки одинаковых байт**. Замена их на пары <счетчик повторений, значение> уменьшает избыточность данных.

Алгоритм *декомпрессии* при этом выглядит так:

Initialization(...);   
do {   
    byte = ImageFile.ReadNextByte();   
    if(является счетчиком(byte)) {   
        counter = Low6bits(byte)+1;   
        value = ImageFile.ReadNextByte();   
        for(i=1 to counter)   
            DecompressedFile.WriteByte(value)   
        }   
    else {   
        DecompressedFile.WriteByte(byte)   
} while(ImageFile.EOF());

В данном алгоритме признаком счетчика (counter) служат единицы в двух верхних битах считанного файла:

http://www.graphicon.ru/oldgr/library/our_publications/fractal/image13.gif

Соответственно оставшиеся 6 бит расходуются на счетчик, который может принимать значения от 1 до 64. Строку из 64 повторяющихся байтов мы превращаем в два байта, т.е. сожмем в 32 раза.

Алгоритм рассчитан на деловую графику — изображения с большими областями повторяющегося цвета. Ситуация, когда файл увеличивается, для этого простого алгоритма не так уж редка. Ее можно легко получить, применяя групповое кодирование к обработанным цветным фотографиям. Для того, чтобы увеличить изображение в два раза, его надо применить к изображению, в котором значения всех пикселов больше двоичного 11000000 и подряд попарно не повторяются.

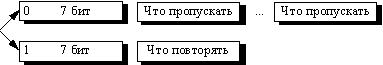
**Второй вариант алгоритма**

Второй вариант этого алгоритма имеет больший максимальный коэффициент архивации и меньше увеличивает в размерах исходный файл.

Алгоритм декомпрессии для него выглядит так:

Initialization(...);   
do {   
    byte = ImageFile.ReadNextByte();   
    counter = Low7bits(byte)+1;   
    if(если признак повтора(byte)) {   
        value = ImageFile.ReadNextByte();   
        for (i=1 to counter)   
        CompressedFile.WriteByte(value)   
    }   
    else {   
    for(i=1 to counter){   
        value = ImageFile.ReadNextByte();   
        CompressedFile.WriteByte(value)   
    }   
    CompressedFile.WriteByte(byte)   
} while(ImageFile.EOF());

Признаком повтора в данном алгоритме является единица в старшем разряде соответствующего байта:



Как можно легко подсчитать, в лучшем случае этот алгоритм сжимает файл в 64 раза (а не в 32 раза, как в предыдущем варианте), в худшем увеличивает на 1/128. Средние показатели степени компрессии данного алгоритма находятся на уровне показателей первого варианта.

**Характеристики алгоритма RLE:**

**Коэффициенты компрессии:** Первый вариант: 32, 2, 0,5. Второй вариант: 64, 3, 128/129. **(Лучший, средний, худший коэффициенты)**

**Класс изображений: Ориентирован алгоритм на изображения с небольшим количеством цветов: деловую и научную графику.**

**Симметричность: Примерно единица.**

**Характерные особенности: К положительным сторонам алгоритма, пожалуй, можно отнести только то, что он не требует дополнительной памяти при архивации и разархивации, а также быстро работает. Интересная особенность группового кодирования состоит в том, что степень архивации для некоторых изображений может быть существенно повышена всего лишь за счет изменения порядка цветов в палитре изображения.**

**Задание к лабораторной работе**

Разработать приложение, реализующее алгоритм кодирования серий.

Пусть дан массив 10000, случайно сформированный из 0 и 1. 0 характеризует белый цвет, 1 – черный. Данный массив находится в файле 1.txt.

1. В приложении должен быть реализован «визуализатор», который исходный массив (в визуализаторе должен быть предусмотрен выбор файла из списка) преобразует в «аватарку» размером 100х100.

2. Затем исходный массив сжимается алгоритмом RLE и результат записывается в файл 2.txt. Данный файл должен быть распознаваемым приложением, разработанным Вашим сокурсником.

3. Файл 2.txt декомпрессируется в файл 3.txt и опять визуализируется в «аватарку» 100х100. Исходный и конечный варианты должны совпадать.

4. Обоснование выбора языка програмирования.