# КФУ (ВШ ИТИС)

**Семетровая работа на тему:**

«Сжатие данных с потерями»

Вариант №9

## Выполнил: студент 11-1011 группы

Нургалиев Ильдар Ильфатович

Проверил: предподаватель ТИиК

### Чикрин Дмитрий Евгеньевич

### *Казань*

*2012/13 уч. год*

* Содержание:

-Введение

-Теория:

1) Типы сжатия

2) Коды префикса:

-алгоритм Хаффмана

-алгоритм Шеннона-Фано

- Подходы к решению задач

- Результаты

- Заключение (Вывод)

- Список литературы

**Введение**

Думая о данных, обычно мы представляем себе ни что иное, как передаваемую этими данными информацию: список клиентов, мелодию на аудио компакт-диске, письмо и тому подобное**.** Характерной особенностью большинства типов данных является их избыточность. Степень избыточности данных зависит от типа данных. Например, для видеоданных степень избыточности в несколько раз больше чем для графических данных, а степень избыточности графических данных, в свою очередь, больше чем степень избыточности текстовых данных. Другим фактором, влияющим на степень избыточности является принятая система кодирования. Примером систем кодирования могут быть обычные языки общения, которые являются ни чем другим, как системами кодирования понятий и идей для высказывания мыслей. Так, установлено, что кодирование текстовых данных с помощью средств русского языка дает в среднем избыточность на 20-25% большую чем кодирование аналогичных данных средствами английского языка.

Для человека избыточность данных часто связана с качеством информации, поскольку избыточность, как правило, улучшает понятность и восприятие информации. Однако, когда речь идет о хранении и передаче информации средствами компьютерной техники, то избыточность играет отрицательную роль, поскольку она приводит к возрастанию стоимости хранения и передачи информации. Особенно актуальной эта проблема стает в случае обработки огромных объемов информации при незначительных объемах носителей данных. В связи с этим, постоянно возникает проблема уменьшения избыточности или сжатия данных.

Хранение и передача информации обходятся участникам информационного процесса недешево. Зная стоимость носителя и его емкость нетрудно подсчитать, во что обходится хранение единицы информации, а зная пропускную способность канала связи и стоимость его аренды, можно определить затраты на передачу единицы информации.

**Теория**

**Типы сжатия.**

Существует два основных типа сжатия данных: «сжатие без потерь» (lossless)и «сжатие с потерями» (lossy). Очевидно, что когда мы имеем дело с информацией типа «номер телефона», то сжатие такой записи за счет потери части символов не ведет ни к чему хорошему. Тем не менее можно представить целый ряд ситуаций, когда потеря части информации не приводит к потери полезности оставшейся. Сжатие с потерями применяется в основном для графики (JPEG), звука (MP3), видео (MPEG), то есть там, где в силу огромных размеров файлов степень сжатия очень важна, и можно пожертвовать деталями, не существенными для восприятия этой информации человеком. Особые возможности для сжатия информации имеются при компрессии видео. В ряде случаев большая часть изображения передается из кадра в кадр без изменений, что позволяет строить алгоритмы сжатия на основе выборочного отслеживания только части «картинки». В частном случае изображение говорящего человека, не меняющего своего положения, может обновляться только в области лица или даже только рта — то есть в той части, где происходят наиболее быстрые изменения от кадра к кадру.

В целом ряде случаев сжатие графики с потерями, обеспечивая очень высокие степени компрессии, практически незаметно для человека. Так, из трех фотографий, показанных ниже, первая представлена в TIFF-формате (формат без потерь), вторая сохранена в формате JPEG c минимальным параметром сжатия, а третья с максимальным. При этом можно видеть, что последнее изображение занимает почти на два порядка меньший объем, чем первая.Однако методы сжатия с потерями обладают и рядом недостатков.

Первый заключается в том, что компрессия с потерями применима не для всех случаев анализа графической информации. Например, если в результате сжатия изображения на лице изменится форма родинки (но лицо при этом останется полностью узнаваемо), то эта фотография окажется вполне приемлемой, чтобы послать ее по почте знакомым, однако если пересылается фотоснимок легких на медэкспертизу для анализа формы затемнения — это уже совсем другое дело. Кроме того, в случае машинных методов анализа графической информации результаты кодирования с потерей (незаметные для глаз) могут быть «заметны» для машинного анализатора.

Вторая причина заключается в том, что повторная компрессия и декомпрессия с потерями приводят к эффекту накопления погрешностей. Если говорить о степени применимости формата JPEG, то, очевидно, он полезен там, где важен большой коэффициент сжатия при сохранении исходной цветовой глубины. Именно это свойство обусловило широкое применение данного формата в представлении графической информации в Интернете, где скорость отображения файла (его размер) имеет первостепенное значение. Отрицательное свойство формата JPEG — ухудшение качества изображения, что делает практически невозможным его применение в полиграфии, где этот параметр является определяющим.

**КОДЫ ПРЕФИКСОВ.**

Большинство широко изучаемых алгоритмов сжатия данных основаны на кодах Хаффмана. В коде Хаффмана каждая буква исходного текста представляется в архиве кодом переменной длины. Более частые буквы представляются короткими кодами, менее частые - длинными. Коды, используемые в сжатом тексте должны подчиняться свойствам префикса, а именно: код, использованный в сжатом тексте не может быть префиксом любого другого кода.

Коды префикса могут быть найдены посредством дерева, в котором каждый лист соответствует одной букве алфавита источника.

**Метод сжатия Шеннона-Фано.**

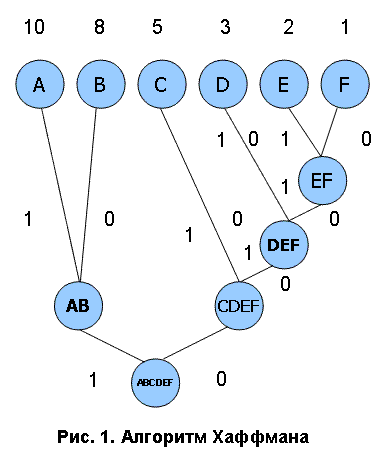
##### Общие принципы, на которых основано сжатие данных

Все методы сжатия данных основаны на простом логическом принципе. Если представить, что наиболее часто встречающиеся элементы закодированы более короткими кодами, а реже встречающиеся – более длинными, то для хранения всех данных потребуется меньше места, чем если бы все элементы представлялись кодами одинаковой длины.   
Точная взаимосвязь между частотами появления элементов, и оптимальными длинами кодов описана в так называемой теореме Шеннона о источнике шифрования(Shannon's source coding theorem), которая определяет предел максимального сжатия без потерь и энтропию Шеннона.

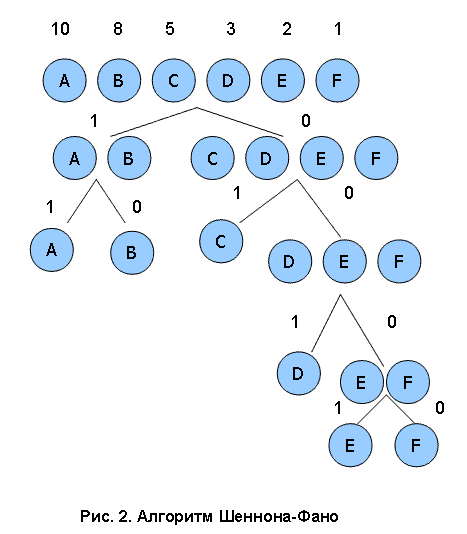
Перейдем к алгоритму сжатия данных, которая была реализована в этой семестровой работе: метод Шеннона-Фано — один из первых алгоритмов сжатия, который впервые сформулировали американские учёные Шеннон и Фано, и он имеет большое сходство с алгоритмом Хаффмана. Алгоритм основан на частоте повторения. Так, часто встречающийся символ кодируется кодом меньшей длины, а редко встречающийся — кодом большей длины.

В свою очередь, коды, полученные при кодировании, префиксные. Это и позволяет однозначно декодировать любую последовательность кодовых слов. Но все это вступление. Для работы оба алгоритма должны иметь таблицу частот элементов алфавита. Итак, алгоритм Хаффмана работает следующим образом:

1. На вход приходят упорядоченные по невозрастанию частот данные.
2. Выбираются две наименьших по частоте буквы алфавита, и создается родитель (сумма двух частот этих «листков»).
3. Потомки удаляются и вместо них записывается родитель, «ветви» родителя нумеруются: левой ветви ставится в соответствие «1», правой «0».
4. Шаг два повторяется до тех пор, пока не будет найден главный родитель — «корень».



Алгоритм Шеннона-Фано работает следующим образом:

1. На вход приходят упорядоченные по невозрастанию частот данные.

2. Находится середина, которая делит алфавит примерно на две части. Эти части (суммы частот алфавита) примерно равны. Для левой части присваивается «1», для правой «0», таким образом мы получим листья дерева

3. Шаг 2 повторяется до тех пор, пока мы не получим единственный элемент последовательности, т.е. листок   
Таким образом, видно, что алгоритм Хаффмана как бы движется от листьев к корню, а алгоритм Шеннона-Фано, используя деление, движется от корня к листьям.

###### **Немного математики**

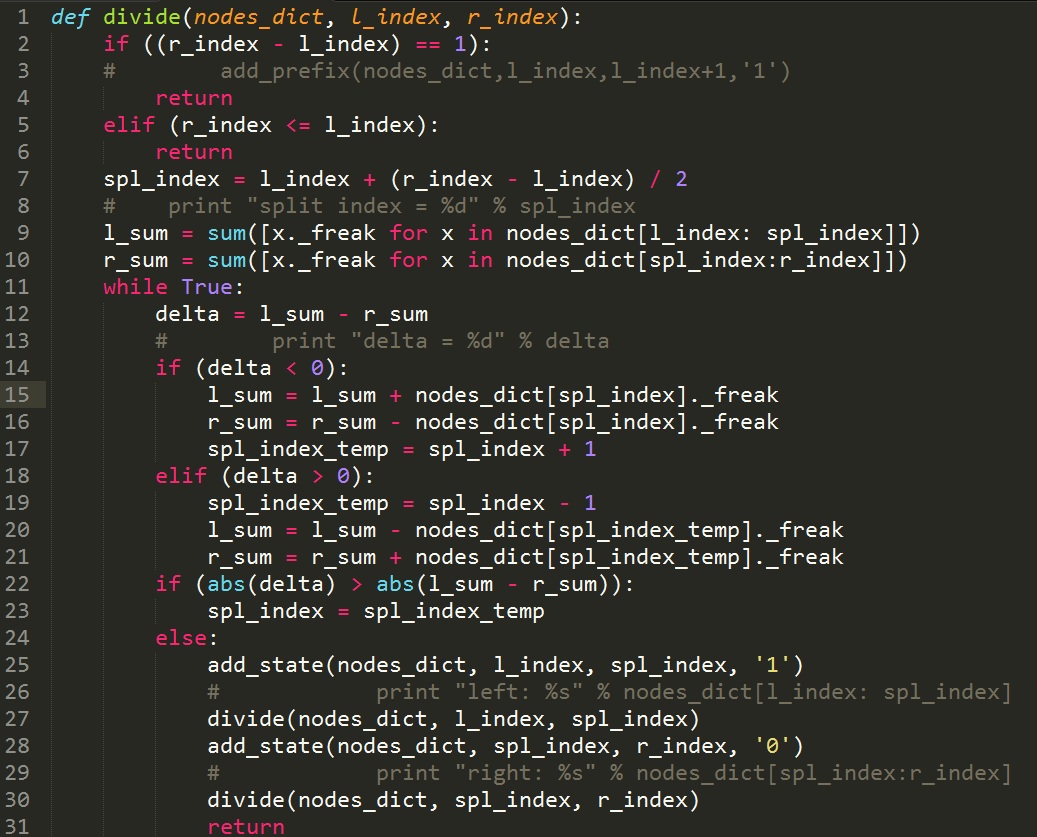
Если вероятность появления элемента si равна p(si), то наиболее выгодно будет представить этот элемент — log2p(si) битами. Если при кодировании удается добиться того, что длина всех элементов будет приведена к log2p(si) битам, то и длина всей кодируемой последовательности будет минимальной для всех возможных методов кодирования. При этом, если распределение вероятностей всех элементов F = {p(si)} неизменно, и вероятности элементов взаимно независимы, то средняя длина кодов может быть рассчитана как  
http://habrastorage.org/storage2/d49/4a2/489/d494a248985cc291d158827125a3d608.png  
Это значение называют энтропией распределения вероятностей F, или энтропией источника в заданный момент времени.

**Подходы к решению задач**

**Инструменты:**

Практическая часть семестровой работы заключалась в представлении набора сигналов в виде гистограммы, их прореживания, кодирования методом Шеннона-Фано и декодирования. Для написания программы, которая выполняла бы данный список задач, я использовал язык программирования Python, так как на нем легче всего программировать в процедурном стиле, то есть писать алгоритмы. Для представления user interface отлично подошел java-script с низкоуровневым plotter-ом - "canvas", при помощи которого была нарисована гистограмма. И для полного функционирования клиентского кода, который предоставляет интерфейс взаимодействия с программой, и серверной части которая, выполняло все остальное задачи из списка, кроме построения гистограммы, использовался фреймворк Django, так как он предоставляет шировие возможности для написания любых веб сервисов.

**Кодирование методом Шеннона-Фано:**

Данный метод кодирования, который был описан выше в разделе "теория: коды префикса", я реализовал его через рекурсивный вызов, вызов происходил два раза подряд, чтобы абстрактно описать структуру дерева.  
****

nodes\_dict - список содержащий элементы типа Node, который я описал для оборачивания ресурсов(символ и его кол-во повторений) и добавив характеристику: кодовое слово, которое по мере углубления в рекурсию мог изменяться.

**Декодирование:**

Полученный код декодировался следующим образом (в виде примера).

Пусть имеется следующая таблица префиксных кодов:

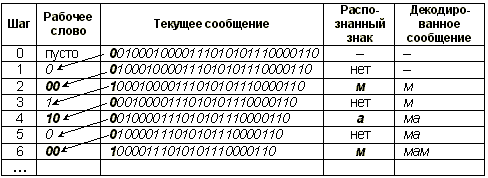
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица 2. Таблица кодов | | | | | |
| а | л | м | р | у | ы |
| 10 | 010 | 00 | 11 | 0110 | 0111 |

    Требуется декодировать сообщение: 00100010000111010101110000110.

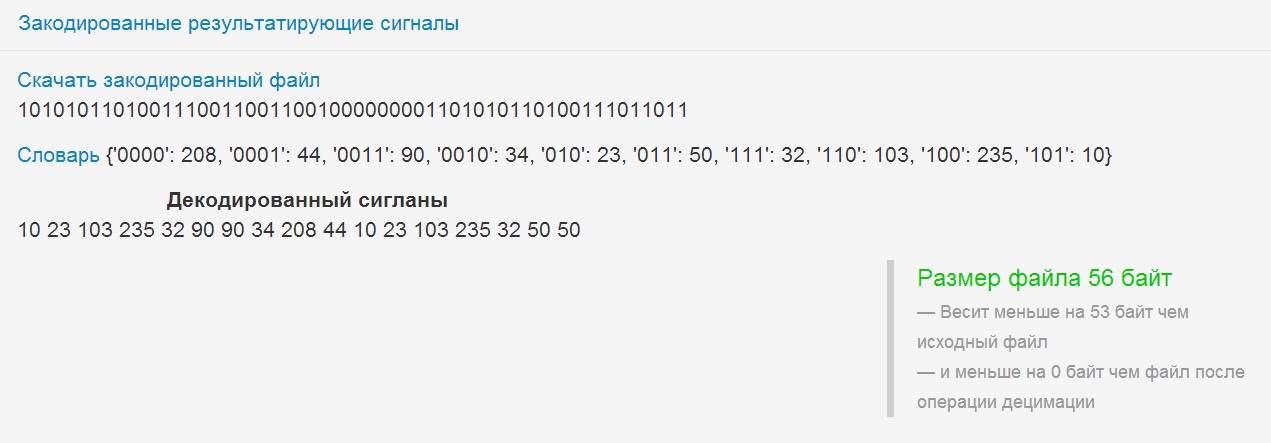
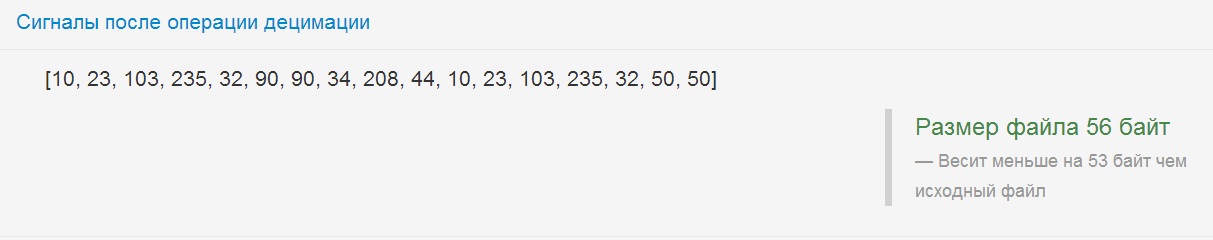
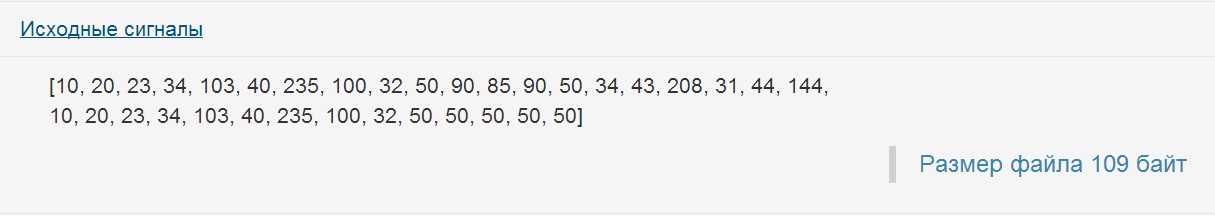
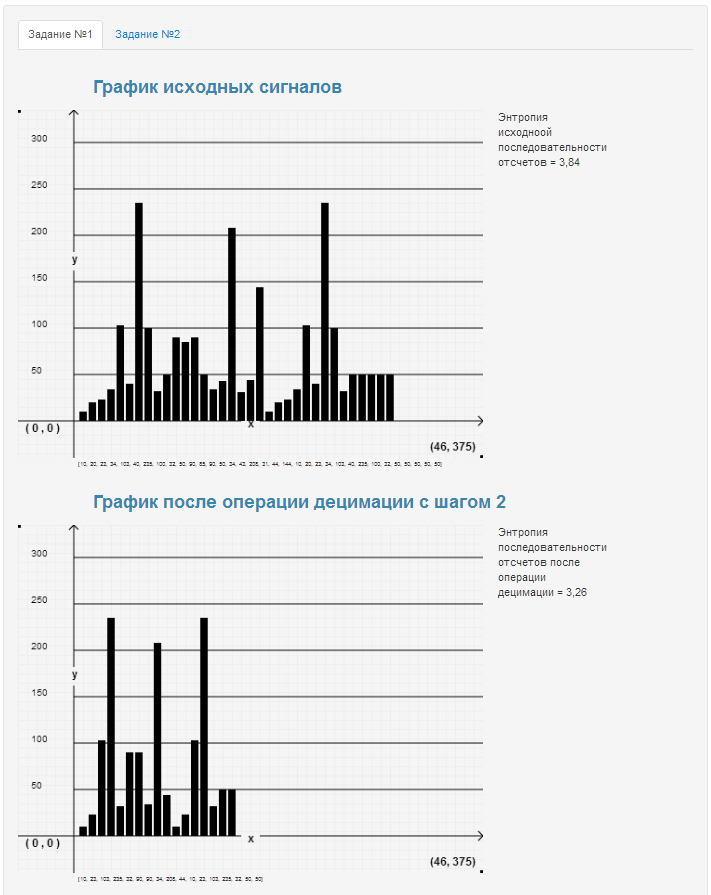
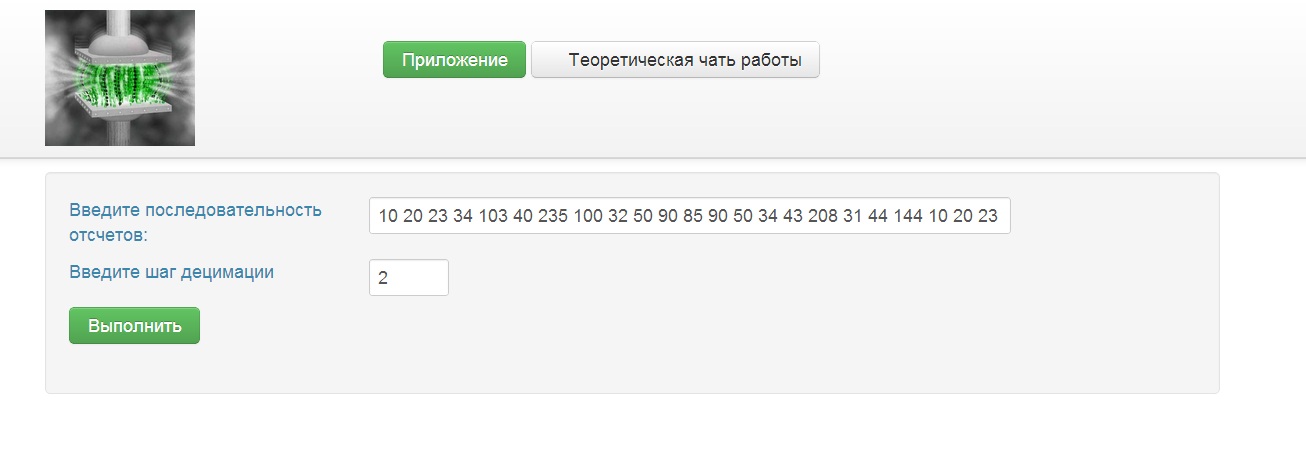
    Декодирование производится циклическим повторением следующих действий.

1. Отрезать от текущего сообщения крайний левый символ, присоединить к рабочему кодовому слову.
2. Сравнить рабочее кодовое слово с кодовой таблицей; если совпадения нет, перейти к (1).
3. Декодировать рабочее кодовое слово, очистить его.
4. Проверить, имеются ли еще знаки в сообщении; если "да", перейти к (1).

    Применение данного алгоритма дает:



    Доведя процедуру до конца, получим сообщение: "мама мыла раму".

**Результаты:**

**Заключение.**

В наш век особо актуально стоит проблема архивации данных, потому как, компьютер является, неотъемлемой частью любого предприятия, а на многих предприятиях обрабатываются огромные базы данных и если их копировать не в сжатом виде, то они занимали бы целые склады серверов.

Факт избыточности свидетельствует о возможностях перехода на иную систему кодирования, которая уменьшила бы избыточность передаваемого сообщения. Существуют идеальные метода и не очень, к примеру в кодирование методом Шеннона-Фано может возникнуть такой неудачный случай, когда при построении кода разбиение можно сделать несколькими способами. И на одном из уровней это может привести к неоптимальности кода в целом, т.е. оптимальность на каждом шаге не дает оптимальности всей совокупности. Например, если взять 8 символов с частотами 5,5,5,3,3,3,3,3, то код Хаффмана всем им даст длину 3 бита, а в коде Шеннона-Фано длины будут, соответственно, такими: 2,3,3,3,3,3,4,4. В результате средняя длина символа будет равна 3,125, что больше 3 (при том, что первое деление на две части прошло точно по границе символов).

**Литература**

* Ватолин Д., Ратушняк А., Смирнов М. Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео; ISBN 5-86404-170-X; 2003 г.
* Д. Сэломон. Сжатие данных, изображения и звука; ISBN 5-94836-027-Х; 2004г.
* [www.wikipedia.org](http://www.wikipedia.org/)
* Тропченко А.Ю., Тропченко А.А. Методы сжатия изображений, аудиосигналов и видео: Учебное пособие – СПб: СПбГУ ИТМО, 2009. – 108 с.
* Статья "Сжатие информации с потерями и без потерь", Александр Прохоров