



ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ  
БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
**ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ**

**Алексеев А. П., Орлов В.В.**

## **Методы сжатия информации**

**Методические указания к проведению лабораторной работы**

**Самара - 2015**

---

ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ  
Кафедра Информатики и вычислительной техники

Методические указания к проведению лабораторной работы  
**«Методы сжатия информации»**

по дисциплине «Информатика»,  
для студентов первого курса  
специальностей

090302 «Информационная безопасность телекоммуникационных систем»,  
200700 «Фотоника и оптоинформатика»,  
210400 «радиотехника»



Авторы-составители:  
доц., к.т.н. **Алексеев А.П.**,  
к.т.н. **Орлов В.В.**

Под общей редакцией Алексеева А.П.

Рецензент д.т.н., проф. **Николаев Б.И.**

Самара, 2013 г.

---

## **Введение**

Сжатие информации - проблема, имеющая достаточно давнюю историю. Методы сжатия информации разрабатывались одновременно с разработкой новых методов помехоустойчивого кодирования и шифрования информации.

Сжатие информации позволяет в единицу времени передать больший объем данных. Это актуально для сетевых технологий, например, при трансляции футбольного матча через Интернет.

В процессе выполнения данной лабораторной работы исследуются три метода сжатия информации: RLE, Шеннона-Фано и Хаффмана.

---

## **Лабораторная работа** **Методы сжатия информации**

### **1. Подготовка к работе**

По указанной литературе изучить принципы сжатия без потерь, порядок работы с архиваторами, ответить на контрольные вопросы.

### **2. Контрольные вопросы**

- 2.1. Перечислите известные Вам методы сжатия информации без потерь.
- 2.2. В чем состоит отличие методов сжатия с потерями и без потерь?
- 2.3. Сколько бит в управляющем байте отводят для указания числа повторяющихся байтов при сжатии методом кодирования длин серий?
- 2.4. О чем говорит равенство единице старшего бита в управляющем байте при сжатии методом кодирования длин серий?
- 2.5. Перечислите известные Вам архиваторы.
- 2.6. Целесообразно ли выполнять сжатие файлов формата JPEG, MP3, MPEG?
- 2.7. Рисунок какого формата будет сжат сильнее BMP или JPEG?
- 2.8. Какой код является неравномерным: RLE или Хаффмана?
- 2.9. Что называется кодом?
- 2.10. Чем отличаются алгоритмы построения кодов Шеннона-Фано и Хаффмана?
- 2.11. Перечислите коды, которые обладают свойством префиксности.
- 2.12. Что называется входным алфавитом?

### 3. Задания на выполнение лабораторной работы

#### 3.1.Задание 1. Выполнить сжатие информации методом RLE

Выполнить вручную кодирование сообщения методом RLE. В качестве исходной фразы взять текст из табл. 3.1. С помощью таблицы CP-1251 (см. Приложение 1) перевести символы заданной фразы в десятичные числа, а затем десятичные числа перевести в двоичные. Выполнить сжатие информации, вычислить контрольные суммы и коэффициент сжатия.

Табл. 3.1.

Вар	Текст	Вар	Текст
1	Кредитка 2235555666122	17	КккккТттттттто тттттам?
2	Паспорт 25700000333215	18	Длинношеее животное
3	ИНН 7888825555488856	19	Урааааааааааа в атаку
4	Пароль 177775556666612	20	Долг 3255566667444444
5	Пароль abcWWWZZZq	21	Телефон 8904222211111
6	Автомобиль 78999994441	22	Ауууууууу заблудились
7	Алло это 456555544488	23	Свидетельство 22263333
8	Удостоверение 265444111	24	Возраст 100000000 лет
9	Счет 95122244445333333	25	Заработали 522211112
10	Касса 147888555233333	26	До дембеля 60440000 с
11	Прошло 11100002 секунд	27	Кредитка 235556999922
12	Пролетели 82223333352 м	28	ИНН 8825577777488856
13	Вес 159755553333331 кг	29	Шифр 159222666644444
14	Цена 2598888666611 коп	30	Улов 98544477778555 кг
15	Мощность 3574444555 Вт	31	Пароль RRWQQQQ6666
16	Выиграл 10000555 рублей	32	Пароль 778SSЫЫIzzzzN

**3.2.Задание 2. Выполнить сжатие информации методом Шеннона-Фано**  
Используя фразу из табл. 3.1, построить кодовое дерево и определить коэффициент сжатия методом Шеннона-Фано.

**3.3.Задание 3. Выполнить сжатие информации методом Хаффмана**  
Используя фразу из табл. 3.1, построить кодовое дерево и определить коэффициент сжатия методом Хаффмана.

### **3.4.Задание 4. Исследовать эффективность сжатия файлов различных форматов**

С помощью стандартного архиватора (WinZip, WinRar, 7-Zip и т.п.) выполнить сжатие различных документов, тип которых указан в таблице 3.4.1

Табл. 3.4.1

Документ	Расширение	Объем файла до архивации, Кбайт	Объем файла после архивации, Кбайт	Коэффициент сжатия
Текст	.doc			
Фотография	.jpg			
Рисунок	.bmp			
Видео	.avi			
Звук	.mp3			

В качестве текстового документа нужно взять файл, который не содержит рисунков. Число символов должно быть более 3000 знаков.

Фотографии нужно взять на сайте **samara.psati.ru** в соответствии с вариантом (таблица 3.4.2.). Следует взять одну фотографию из указанного раздела.

Табл. 3.4.2

Вар	Раздел	Вар	Раздел
1	Администрация города	17	Музеи, выставки
2	Архивные материалы	18	Набережные, пляжи
3	Банки	19	Ночной город
4	Водоёмы	20	Окраины города
5	Вокзалы	21	Памятники и скульптуры
6	Гостиницы	22	Парки, сады, скверы
7	Дворцы, дома	23	Площади
8	Деревянный город	24	Растительный мир
9	Животный мир	25	Рестораны, кафе, бары

10	Заводы, фабрики	26	Сооружения
11	Закаты, рассветы	27	Спортивные сооружения
12	Кинотеатры	28	Театры, концертные залы
13	Культовые сооружения	29	Торговые центры
14	Массовые мероприятия	30	Транспорт
15	Медицинские учреждения	31	Учебные заведения
16	Улицы, проспекты	32	Фонтаны

В качестве рисунка формата BMP нужно взять изображение флага государства в соответствии с таблицей 3.4.3.

Табл.3.4.3

Вариант	Страна	Вариант	Страна
1.	Австралия	17.	Германия
2.	Австрия	18.	Греция
3.	Азербайджан	19.	Испания
4.	Албания	20.	Италия
5.	Алжир	21.	Казахстан
6.	Ангола	22.	КНР
7.	Андорра	23.	Нидерланды
8.	Аргентина	24.	Норвегия
9.	Армения	25.	Польша
10.	Афганистан	26.	Россия
11.	Белоруссия	27.	Румыния
12.	Бельгия	28.	США
13.	Болгария	29.	Украина
14.	Бразилия	30.	Уругвай
15.	Великобритания	31.	Франция
16.	Венгрия	32.	Япония

Видеоклип желательно снять самостоятельно (с помощью видеокамеры, цифрового фотоаппарата, мобильного телефона, планшетника).

В качестве звукового файла используйте своё любимое музыкальное произведение.

---

## 4. Методические указания

### 4.1. Методические указания к заданию 3.1

Несмотря на то, что объемы внешней памяти ЭВМ постоянно растут, потребность в сжатии информации не уменьшается. Это объясняется тем, что сжатие необходимо не только для экономии места в памяти ЭВМ, но и для быстрой передачи информации по Сети.

Кроме того, возможность отказа магнитных носителей информации, разрушающее действие вирусов заставляют пользователей делать резервное копирование ценной информации на другие (запасные) носители информации. Очевидно, что разумнее информацию хранить сжатой.

**Сжатие информации (архивация)** — это такое преобразование информации, при котором объем файла уменьшается, а количество информации, содержащейся в архиве, остается прежним.

Степень сжатия информации зависит от содержимого файла и формата файла, а также от выбранного метода сжатия информации. Степень (качество) сжатия файлов характеризуется **коэффициентом сжатия**  $K_c$ , определяемым как отношение объема исходного файла  $V_o$  к объему сжатого файла  $V_c$ :

$$K_c = \frac{V_o}{V_c}.$$

Чем больше величина  $K_c$ , тем выше степень сжатия информации.

Все существующие методы сжатия информации можно разделить на два класса: сжатие без потерь информации (обратимый алгоритм) и сжатие с потерей информации (необратимый алгоритм). В первом случае исходную информацию можно точно восстановить по имеющейся упакованной информации. Во втором случае распакованное сообщение будет отличаться от исходного сообщения.

В настоящее время разработано много алгоритмов архивации без потерь. Однако все они используют две простые идеи.

Первая идея, основанная на учете частот появления символов в тексте, была разработана **Клодом Шенноном** и независимо от него **Робертом Фано**, а затем в 1952 г. развита **Дэвидом Хаффманом** - аспирантом Массачусетского технологического института при написании им курсовой работы. Идея базируется на том факте, что в обычном тексте частоты появления различных символов неодинаковые.

Вторая основная идея сжатия состоит в учете того факта, что в файлах часто встречаются несколько подряд идущих одинаковых байтов, а некоторые последовательности байтов повторяются многократно. При архивации такие места файла можно заменить командами вида «повторить данный байт  $n$  раз» или «взять часть данных длиной  $k$  байтов, которая встречалась  $m$  байтов назад». Такой алгоритм архивации носит имя RLE (**R**un **L**ength **E**ncoding — кодирование путем учета повторений или кодирование длин серий).

Рассмотрим детально метод сжатия RLE.

Упакованная методом RLE последовательность состоит из **управляющих**



---

**байтов**, за которыми следуют один или несколько байтов данных. При этом если старший бит управляющего байта равен 1, то следующий за ним байт данных нужно повторить при декодировании столько раз, сколько указано в оставшихся 7 битах управляющего байта.

Например, управляющий байт 10001001 говорит, что следующий за ним байт нужно повторить 9 раз, так как  $1001_2 = 9_{10}$ .

Если старший бит управляющего байта равен 0, то при декодировании нужно взять несколько следующих байтов без изменений. Число байтов, которые берутся без изменений, указывается в оставшихся 7 битах. Например, управляющий байт 00000011 говорит, что следующие за ним 3 байта нужно взять без изменений.

Рассмотрим пример сжатия методом RLE.

Пусть дана некоторая последовательность из 12 байтов:

```
11111111 11111111 11111111 11111111 11111111 11110000  
00001111 11000011 10101010 10101010 10101010 10101010.
```

В начале исходной двоичной последовательности 5 раз повторяется байт 11111111. Чтобы упаковать эти 5 байтов, нужно записать сначала управляющий байт 10000101, а затем повторяемый байт 11111111. В результате сжатия этого фрагмента данных выигрыш составит 3 байта. Далее идут 3 разных (неповторяющихся) байта: 11110000 00001111 и 11000011. Чтобы их «упаковать», нужно записать управляющий байт 00000011, а затем указать эти 3 неповторяющихся байта. В результате архивации этого фрагмента двоичной последовательности получается увеличение объема архива на 1 байт. Далее в последовательности 4 раза повторяется байт 10101010. Для архивации этого фрагмента двоичных данных нужно сформировать управляющий байт 10000100 и записать повторяемый байт 10101010. Сжатие последнего фрагмента даст выигрыш 2 байта.

В результате такой архивации получена новая последовательность данных (архив), состоящая из 8 байтов:

```
10000101 11111111 00000011 11110000  
00001111 11000011 10000100 10101010.
```

Таким образом, 12 байт исходной двоичной последовательности удалось сжать до 8 байт.

$$K_c = \frac{12}{8} = 1,5.$$

*Пример.*

Выполним сжатие сообщения методом RLE. Текст сообщения:  
ИНН 222221333.

Фраза	Десятичный код, таблица CP-1251	Двоичный код	Архив
И	200	11001000	00000001
Н	205	11001101	11001000
Н	205	11001101	10000010
	32	00100000	11001101
2	50	00110010	00000001
2	50	00110010	00100000
2	50	00110010	10000101
2	50	00110010	00110010
2	50	00110010	00000001
1	49	00110001	00110001
3	51	00110011	10000011
3	51	00110011	00110011
3	51	00110011	
КС двоичная			10010000В
КС шестнадцатеричная			90Н

Таким образом, тринадцать байт исходного текста сжаты до двенадцати байт.

Для упрощения проверки результата сжатия были вычислены контрольные суммы (КС) в двоичной и шестнадцатеричной системах счисления.

Коэффициент сжатия в данном случае составил:

$$K_c = \frac{13}{12} = 1,08.$$

## 4.2. Методические указания к заданию 3.2

Задачу экономного кодирования сообщений источника, имеющего отличное от равномерного распределение вероятностей появления символов его алфавита, позволяют решить неравномерные префиксные коды.

Рассмотрим принцип построения таких кодов методами Шеннона-Фано и Хаффмана [2, 3]. Оба этих кода основываются на статистических свойствах источника сообщений и ставят в соответствие часто встречающимся символам алфавита короткие кодовые комбинации.



Рис. 4.2.1. Абсолютные частоты появления букв в книге [1]

Из-за того, что разным символам алфавита соответствуют кодовые комбинации разной длины, такие коды называют неравномерными.

В качестве примера возьмем сообщение: «ИНН 637322757237». Данный текст содержит избыточность, которую невозможно устранить с помощью метода RLE. Избыточность определяется по формуле:

$$L = \left(1 - \frac{H}{n}\right) \cdot 100\% ,$$

где  $H$  - энтропия сообщения;

$n$  - длина кодовой комбинации при равномерном кодировании.

Энтропия сообщения вычисляется по формуле:

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

где  $N$  - объем алфавита источника (для русского алфавита  $N = 33$ );

$p_i$  - относительная частота (вероятность) появления символа в сообщении.

Относительная частота встречаемости символа определяется как отношение абсолютной частоты появления символа в сообщении к общей длине сообщения (числу символов в сообщении):

$$p_i = \frac{\omega_i}{m} ,$$

где  $\omega_i$  - абсолютная частота (частость) встречаемости  $i$ -ого символа алфавита источника;  $m$  - число символов в сообщении.

В данном случае энтропия сообщения равна:

$$H = -\left(\frac{4}{16}\log_2\frac{4}{16} + 2 \cdot \frac{3}{16}\log_2\frac{3}{16} + \frac{2}{16}\log_2\frac{2}{16} + 4 \cdot \frac{1}{16}\log_2\frac{1}{16}\right) = 2,781 \text{ бит/символ},$$

где  $p_1 = \frac{4}{16} = \frac{1}{4}$  - относительная частота появления символа «7»;

$p_2 = p_3 = \frac{3}{16}$  - относительная частота появления символов «2» и «3»;

$p_4 = \frac{2}{16} = \frac{1}{8}$  - относительная частота появления символа «Н»;

$p_5 = p_6 = p_7 = p_8 = \frac{1}{16}$  - относительная частота появления символов «5», «6», «И»,

Пробел.

При необходимости расчета логарифма по основанию два через логарифм по основанию десять можно воспользоваться соотношением:

$$\log_2 x = \frac{\lg x}{\lg 2}.$$

При использовании равномерного кода (например, СР-1251) длина кодовой комбинации определяется так:

$$n = \lceil \log_2 N \rceil,$$

$\lceil x \rceil$  - функция округления аргумента до ближайшего целого значения, не меньшего, чем  $x$ .

В данном примере  $n = \lceil \log_2 8 \rceil = 3$  бита.

Избыточность сообщения при кодировании равномерным кодом равна:

$$L = \left(1 - \frac{2,781}{3}\right) \cdot 100\% = 7,3\%.$$

На первом этапе построения кода Шеннона-Фано формируется таблица абсолютных частот символов.

Символ	Абсолютная частота $\omega_i$	Символ	Абсолютная частота $\omega_i$
7	4	5	1
2	3	6	1
3	3	И	1
Н	2	Пробел	1

Для получения кодовых комбинаций строится кодовое дерево. При построении кода Шеннона-Фано дерево строится от корня к листьям (в отличие от настоящего дерева здесь корень располагается вверху, а листья – внизу). В качестве корня используется множество всех символов алфавита сообщения (рис. 1), упорядоченное по частоте встречаемости символов. Число сверху таблицы равно суммарной частоте символов в исходном сообщении.

$$^{16}$$

7	2	3	Н	5	6	И	
---	---	---	---	---	---	---	--

Рис. 1 - Корень кодового дерева Шеннона-Фано

Затем множество символов делят на два подмножества так, чтобы новые множества имели равные, насколько это возможно, суммарные частоты встре-

чаемости входящих в них символов. Эти подмножества соединяются с корнем дерева ветвями, становясь потомками. Левая ветвь дерева обозначается символом 1, а правая ветвь – символом 0 (рис. 2).

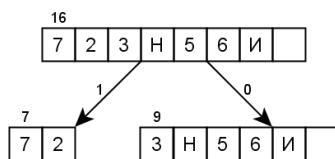


Рис. 2 - Деление на подмножества

Полученные подмножества также рекурсивно делятся до тех пор, пока не будут сформированы листья дерева – отдельные символы сообщения (рис. 3).

Кодовые комбинации (на рис. 3 они указаны в кавычках под соответствующими листьями) получаются при движении от корня дерева к кодируемому символу-листу путем сбора бит, присвоенных пройденным ветвям дерева. Запись кодовой комбинации ведут в направлении от старших разрядов к младшим. Например, при кодировании символа «3» сначала следует пройти по правой ветви к множеству {3, Н, 5, 6, И, Пробел} (к кодовой комбинации добавляется бит 0). Затем нужно пройти по левой ветви к множеству {3, Н} (к кодовой комбинации добавляется бит 1). Наконец, нужно пройти по левой ветви, чтобы достичь листа «3». Таким образом, получена кодовая комбинация «011».

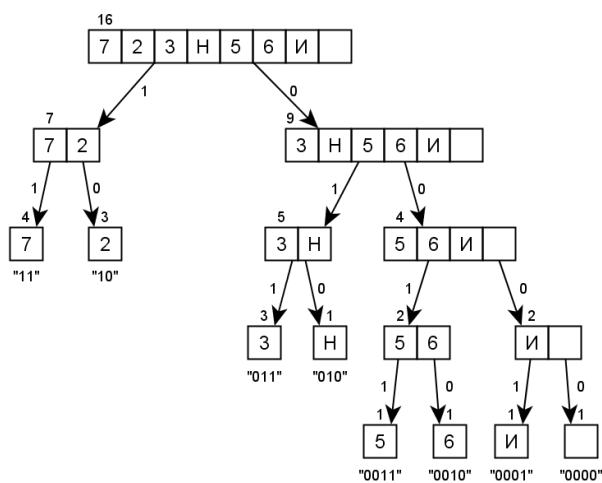


Рис. 3. Дерево кода Шеннона-Фано

При декодировании биты считываются из входного потока и используются, как указатели направления движения по кодовому дереву от корня к искомому листу. При достижении листа найденный символ записывается в выходной поток, а движение по кодовому дереву снова начинают от корня. Например, декодирование комбинации «010» происходит следующим образом. Из потока считывается бит 0, следовательно, нужно пройти по правой ветви от корня дерева к узлу {3, Н, 5, 6, И, Пробел}. Следующий бит единичный, что требует пройти по левой ветви к множеству {3, Н}. Наконец, следующий бит 0 приводит декодер по правой ветви к листу «Н».

В следующей таблице приведены все разрешенные комбинации полученного кода Шеннона-Фано.

Символ	Кодовая комбинация	Символ	Кодовая комбинация
7	11	5	0011
2	10	6	0010
3	011	И	0001
Н	010	Пробел	0000

Закодированное сообщение будет иметь вид:

000101001000000010011110111010110011111001111

Общая длина закодированного сообщения составляет 45 бит.

Средняя длина кодовой комбинации равна (напомним, что число символов в сообщении – 16):

$$n = \frac{45}{16} = 2,813 \text{ бит/символ.}$$

Избыточность сообщения после применения кода Шеннона-Фано снизилась до значения:

$$L = \left(1 - \frac{2,781}{2,813}\right) \cdot 100\% = 1,13\% .$$

Несложно убедиться, что применение кода Шеннона-Фано позволило существенно уменьшить избыточность сообщения. При равномерном кодировании рассмотренного сообщения с помощью кодовой таблицы СР-1251 пришлось бы передать 128 бит.

### 4.3. Методические указания к заданию 3.3

В 1952 году Дэвид Хаффман предложил метод экономного префиксного кодирования с минимальной избыточностью. Как и код Шеннона-Фано, код Хаффмана требует получения априорных сведений о статистических свойствах источника сообщения, то есть необходима таблица абсолютных частот символов данного сообщения. На основе этих данных строится кодовое дерево, также называемое деревом Хаффмана или Н-деревом. В отличие от кода Шеннона-Фано, дерево Хаффмана строится в направлении от листьев к корню (в обратном направлении).

Построение кодового дерева начинают с того, что формируют набор листьев, имеющих веса, равные частотам появления символов в исходном (сжимаемом) сообщении. Листья ранжируют в соответствии с их весами (записывают веса справа налево в порядке их возрастания). Затем выбирают пару узлов (листьев), имеющих наименьший вес, которые соединяют дугами с новым узлом, вес которого равен сумме весов присоединенных к нему потомков. Образовавшийся узел называется родителем. Новый узел (родитель) участвует в

дальнейших построениях дерева. Процедура объединения свободных узлов ведется до тех пор, пока не останется единственный узел (корень дерева).

На рис. 4 показан первый этап построения дерева.

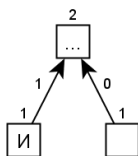


Рис. 4 - Первый этап формирования дерева Хаффмана

На втором шаге аналогично объединим общим родителем узлы, соответствующие символам «5» и «6». На третьем шаге, имея несколько возможностей для объединения узлов, выберем сформированные на первых двух шагах узлы с весом 2 (рис. 5).

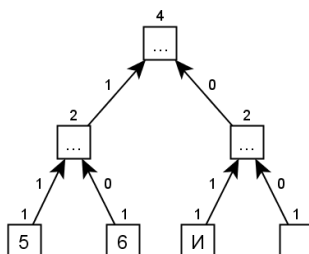


Рис. 5 - Третий шаг построения дерева Хаффмана

На четвертом шаге объединяем узлы «3» и «Н», получая новый узел с весом 5 (рис. 6).

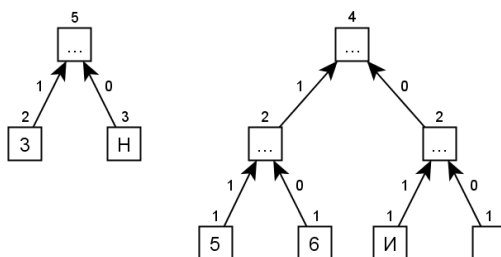


Рис. 6 – Четвертый шаг построения дерева Хаффмана

На пятом шаге узлами с наименьшими весами, используемыми для построения, выберем «2» с весом 3 и составной узел «5, 6, И, Пробел» с весом 4 (рис. 7).

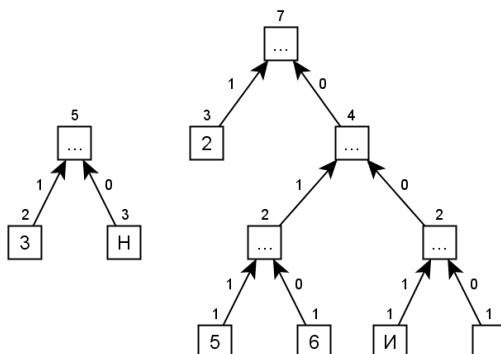


Рис. 7 – Пятый шаг построения дерева Хаффмана

На шестом шаге объединяем узел «7» с весом 4 и составной узел «3, Н» с весом 5 (рис. 8).

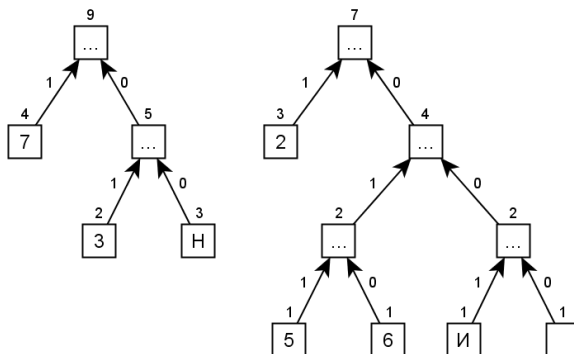


Рис. 8 – Шестой шаг построения дерева Хаффмана

В результате объединения двух составных узлов «7, 3, Н» с весом 9 и «2, 5, 6, И, Пробел» с весом 7 получаем итоговое дерево (рис. 9).

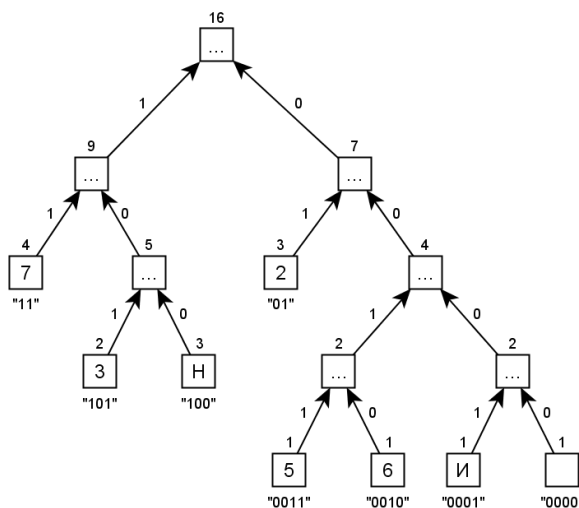


Рис. 9 - Дерево Хаффмана

Кодовые слова формируются так же, как и в случае кода Шеннона-Фано. Все разрешенные кодовые комбинации кода Хаффмана приведены в таблице.

Символ	Кодовая комбинация	Символ	Кодовая комбинация
7	11	5	0011
2	01	6	0010
3	101	И	0001
Н	100		0000

Закодированное сообщение выглядит так: «000110010000000010101111010101110011110110111». Общая длина закодированного сообщения равна 45 бит, средняя длина кодовой комбинации  $n = \frac{45}{16} = 2,813$  бит/символ. Избыточность сжатого сообщения равна:



---

$$L = \left(1 - \frac{2,781}{2,813}\right) \cdot 100\% = 1,13\% .$$

Средняя длина и избыточность для рассмотренного примера получились такими же, как у кода Шеннона-Фано.

Полученные коды Шеннона-Фано и Хаффмана обладают свойством префиксности. Это означает, что ни одна кодовая комбинация не является началом другой, что позволяет обеспечить однозначное декодирование и нет необходимости двоичные символы разделять пробелами.

Как уже отмечалось ранее, для кодирования и декодирования сообщений, сжатых по методам Шеннона-Фано и Хаффмана, кодек должен обладать априорной информацией о статистике сообщения. Поэтому кроме самого сообщения на приемную сторону необходимо передать таблицу частот символов данного сообщения, что увеличивает длину передаваемых данных и снижает фактическую эффективность сжатия. Тем не менее, этот недостаток нивелируется при сжатии больших объемов данных, например, при сохранении изображений в формате JPEG.

### Список литературы

1. Алексеев А.П. Информатика 2007.– СОЛОН-ПРЕСС, 2007. – 608 с.
2. Shannon C. E. «A mathematical theory of communication», Bell Sys. Tech. Jour., vol. 27, pp. 379-423; July, 1948.
3. Huffman D. A., «A method for the construction of minimum-redundancy codes», Proc. Inst. Radio Engineers, vol. 40, no. 9, pp. 1098-1101, Sep. 1952.

## Приложение 1

### Таблица CP-1251

<b>пробел</b>	32	<b>!</b>	33	<b>"</b>	34	<b>#</b>	35	<b>\$</b>	36
<b>%</b>	37	<b>&amp;</b>	38	<b>'</b>	39	<b>(</b>	40	<b>)</b>	41
<b>*</b>	42	<b>+</b>	43	<b>,</b>	44	<b>-</b>	45	<b>.</b>	46
<b>/</b>	47	<b>0</b>	48	<b>1</b>	49	<b>2</b>	50	<b>3</b>	51
<b>4</b>	52	<b>5</b>	53	<b>6</b>	54	<b>7</b>	55	<b>8</b>	56
<b>9</b>	57	<b>:</b>	58	<b>;</b>	59	<b>&lt;</b>	60	<b>=</b>	61
<b>&gt;</b>	62	<b>?</b>	63	<b>@</b>	64	<b>A</b>	65	<b>B</b>	66
<b>C</b>	67	<b>D</b>	68	<b>E</b>	69	<b>F</b>	70	<b>G</b>	71
<b>H</b>	72	<b>I</b>	73	<b>J</b>	74	<b>K</b>	75	<b>L</b>	76
<b>M</b>	77	<b>N</b>	78	<b>O</b>	79	<b>P</b>	80	<b>Q</b>	81
<b>R</b>	82	<b>S</b>	83	<b>T</b>	84	<b>U</b>	85	<b>V</b>	86
<b>W</b>	87	<b>X</b>	88	<b>Y</b>	89	<b>Z</b>	90	<b>[</b>	91
<b>\</b>	92	<b>]</b>	93	<b>^</b>	94	<b>_</b>	95	<b>`</b>	96
<b>a</b>	97	<b>b</b>	98	<b>c</b>	99	<b>d</b>	100	<b>e</b>	101
<b>f</b>	102	<b>g</b>	103	<b>h</b>	104	<b>i</b>	105	<b>j</b>	106
<b>k</b>	107	<b>l</b>	108	<b>m</b>	109	<b>n</b>	110	<b>o</b>	111
<b>p</b>	112	<b>q</b>	113	<b>r</b>	114	<b>s</b>	115	<b>t</b>	116
<b>u</b>	117	<b>v</b>	118	<b>w</b>	119	<b>x</b>	120	<b>y</b>	121
<b>z</b>	122	<b>A</b>	192	<b>Б</b>	193	<b>B</b>	194	<b>Г</b>	195
<b>Д</b>	196	<b>Е</b>	197	<b>Ж</b>	198	<b>З</b>	199	<b>И</b>	200
<b>Й</b>	201	<b>К</b>	202	<b>Л</b>	203	<b>М</b>	204	<b>Н</b>	205
<b>О</b>	206	<b>П</b>	207	<b>Р</b>	208	<b>С</b>	209	<b>Т</b>	210
<b>У</b>	211	<b>Ф</b>	212	<b>Х</b>	213	<b>Ц</b>	214	<b>Ч</b>	215
<b>Ш</b>	216	<b>Щ</b>	217	<b>Ъ</b>	218	<b>Ы</b>	219	<b>Ь</b>	220
<b>Э</b>	221	<b>Ю</b>	222	<b>Я</b>	223	<b>а</b>	224	<b>б</b>	225
<b>в</b>	226	<b>г</b>	227	<b>д</b>	228	<b>е</b>	229	<b>ж</b>	230
<b>з</b>	231	<b>и</b>	232	<b>й</b>	233	<b>к</b>	234	<b>л</b>	235
<b>м</b>	236	<b>н</b>	237	<b>о</b>	238	<b>п</b>	239	<b>р</b>	240
<b>с</b>	241	<b>т</b>	242	<b>у</b>	243	<b>ф</b>	244	<b>х</b>	245
<b>ц</b>	246	<b>ч</b>	247	<b>ш</b>	248	<b>щ</b>	249	<b>ъ</b>	250
<b>ы</b>	251	<b>ь</b>	252	<b>э</b>	253	<b>ю</b>	254	<b>я</b>	255