

**ССЖКАТТИВЕЕ**

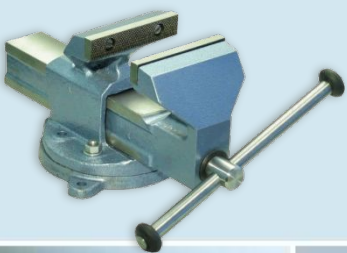
*информации*



# Сжатие текстовой информации

123	Использование цифр вместо слов
$a \parallel b, x \in [-2, 8]$	Математические символы
д.ф.-м.н., проф. ЯрГУ	Общепринятые сокращения
imo, plz, ruok	Молодежный язык SMS
	Стенография





# Основные понятия

Кодирование информации является **избыточным**, если количество бит в полученном коде больше, чем это необходимо для однозначного декодирования исходной информации.

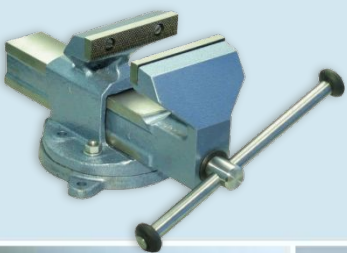
статья

Википедия  
Свободная энциклопедия

**Сжатие данных** — процедура перекодирования данных, производимая с целью уменьшения их объёма.

**Декомпрессия** - это способ восстановления сжатых данных в исходные.





# Архиваторы

Под архиватором понимается программа-архиватор, формат архива и метод сжатия в комплексе.

Лекции по информ...

Type: Документ Microsoft Word  
Author: Gygabite  
Title: http://it  
Date Modified: 27.09.2007 8:14  
Size: 1,45 MB

Лекции по информатике

Type: WinRAR archive  
Date Modified: 27.09.2007 16:25  
Size: 251 KB

Коэффициент сжатия:

$$\frac{1,45 \cdot 1024}{251} \approx 5,92$$

Имя и параметры архива

Общие | Дополнительно | Файлы | Резервные копии | Время | Комментарий

Имя архива: Family.rar

Метод обновления: Добавить с заменой файлов

Профили...

Формат архива:  
 RAR  
 ZIP

Метод сжатия:  
Обычный  
Без сжатия  
Скоростной  
Быстрый  
Обычный  
Хороший  
Максимальный

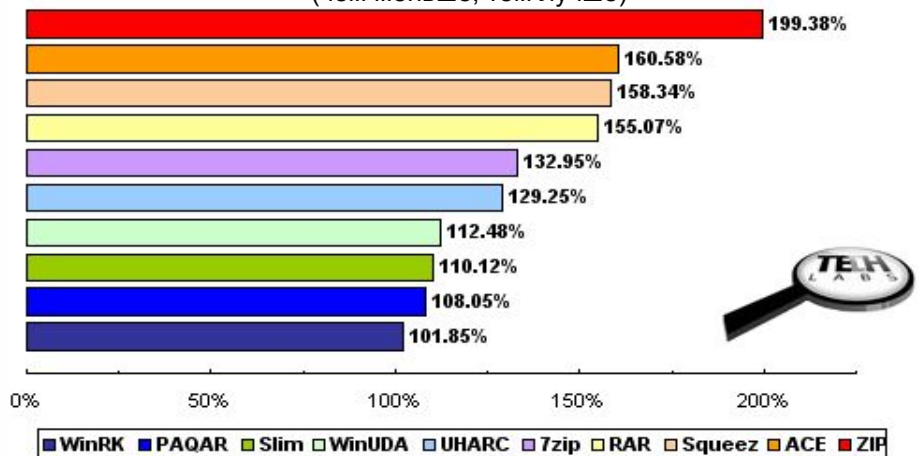
Параметры архивации:  
 Удалить файлы после упаковки  
 Создать SFX-архив  
 Создать непрерывный архив  
 Добавить электронную подпись  
 Добавить информацию для восстановления  
 Протестировать файлы после упаковки  
 Заблокировать архив

OK | Отмена | Справка



# Сравнительные характеристики

Средняя степень сжатия архиваторов  
(чем меньше, тем лучше)

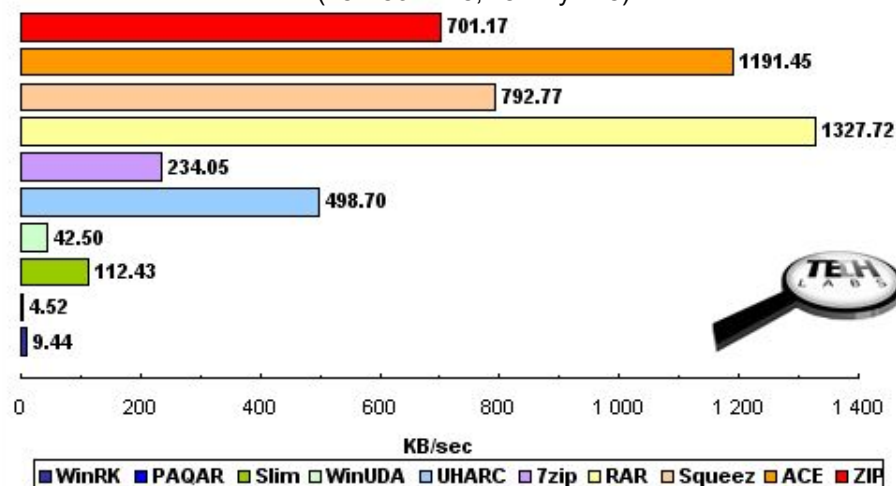


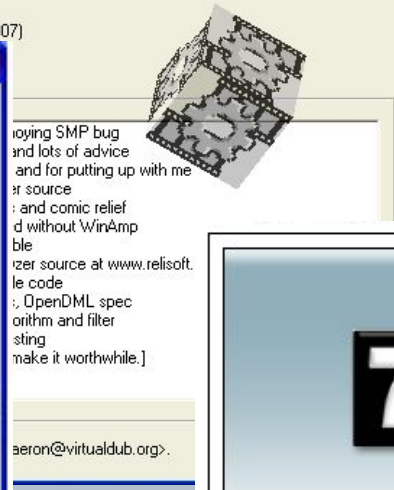
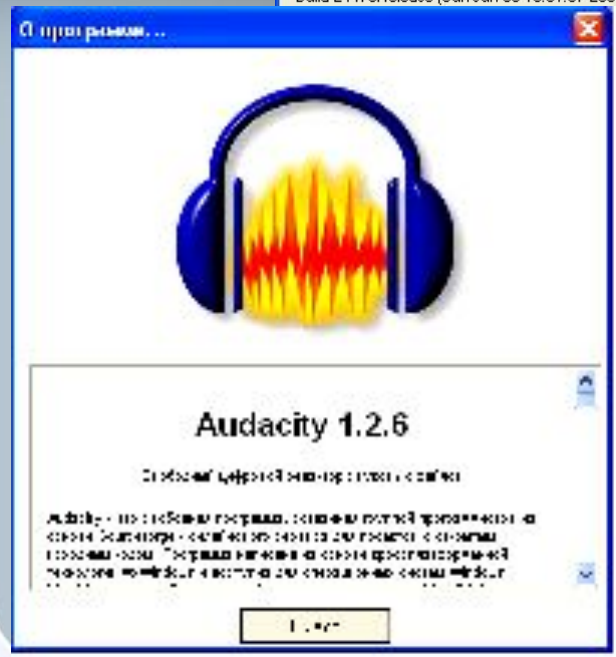
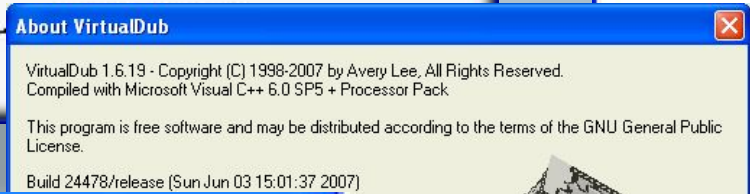
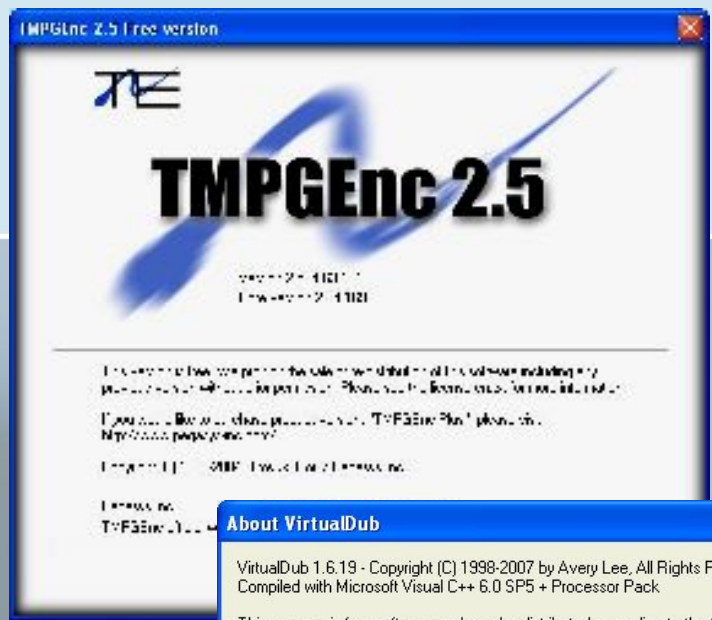
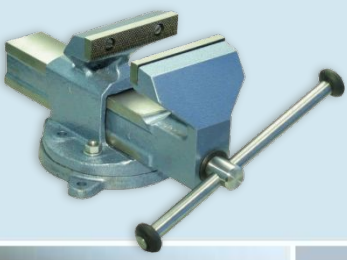
Степень сжатия зависит от

- используемого архиватора;
- метода сжатия;
- типа исходного файла.

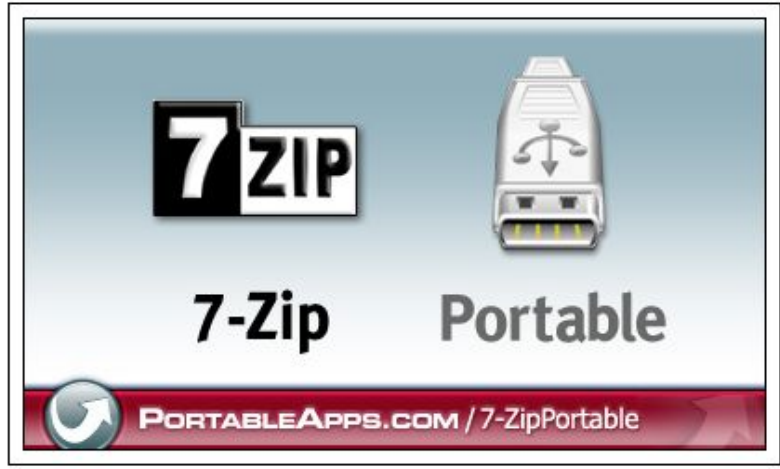
Архиваторы **ACE**, **RAR** и **Squeez** имеют близкие результаты с небольшим преимуществом по степени сжатия у RAR, и при высокой скорости сжатия у Squeez.

Средняя скорость сжатия архиваторов  
(чем больше, тем лучше)





Бесплатные программы для обработки аудио- и видеoinформации





# Виды сжатий

## Сжатие

### с потерями

Восстановление возможно с искажениями, малозаметными для человеческого глаза или уха.

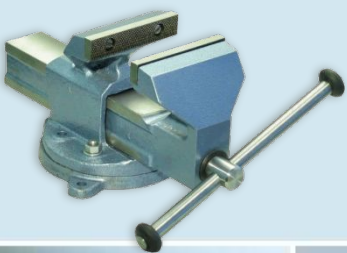
*Форматы файлов:  
jpg, tpeg, adpcm .*

### без потерь

Возможно восстановление исходных данных без искажений.

*Форматы файлов:  
gif, tif, png, psx,  
avi, zip, rar, arj.*

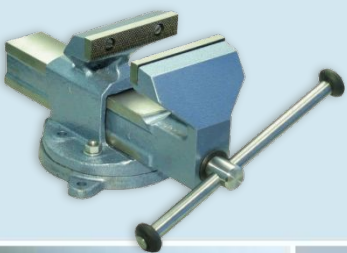




# Сжатие с потерей качества

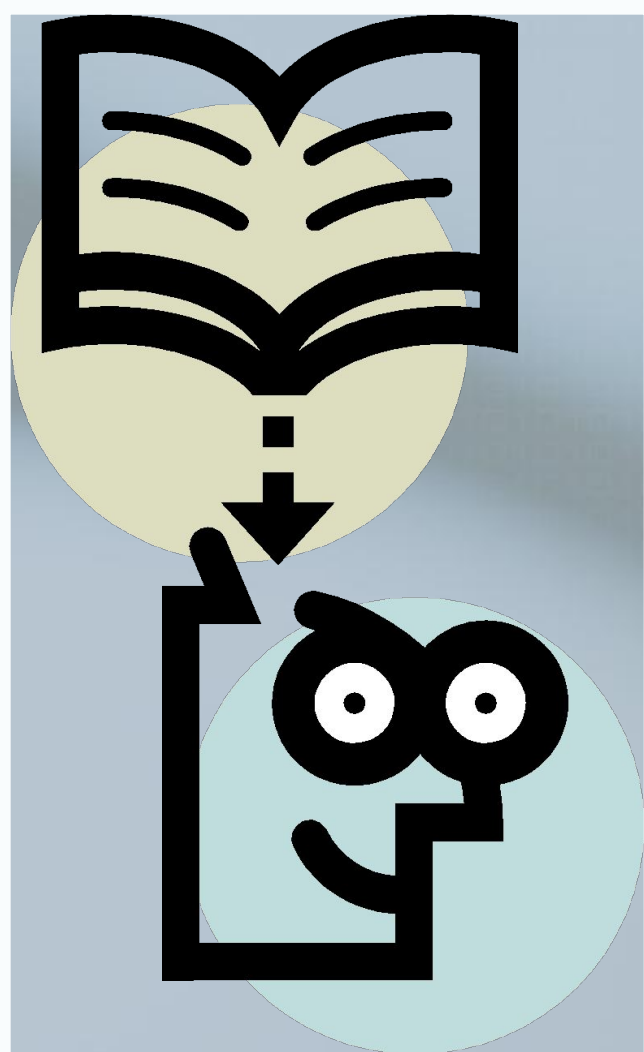






# Сжатие без потерь

## Алгоритм **RLE**



от англ. **R**un **L**ength **E**ncoding

В файле записывается,  
сколько раз повторяются  
одинаковые байты.

Схематично:

```
"RRRRRRGGGGBBBBBBRRRRB  
BRRRRRRR"
```

```
"5R3G6B3R2B7R".
```



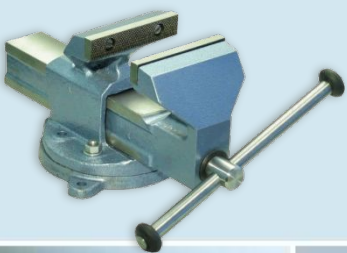
# ASCII-коды

В восьмиразрядной таблице символьной кодировки ASCII каждый символ кодируется восемью битами и, следовательно, занимает в памяти 1 байт.

<i>Знак, клавиша</i>	<i>Двоичный код</i>	<i>10-ый код</i>
пробел	<b>00100000</b>	32
А (лат.)	<b>01000001</b>	65
В (лат.)	<b>01000010</b>	66
Z	<b>01011010</b>	90
0	<b>00110000</b>	48
1	<b>00110001</b>	49
9	<b>00111001</b>	57
Клавиша Esc	<b>00011011</b>	27
Клавиша Enter	<b>00001101</b>	13

Некоторые  
ASCII-коды





# Сжатие без потерь

## Метод упаковки

### Пример №1

0	1	2	3	4	5	6	7	8	9
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

1010	1011	1100	1101	1110	1111

Двукратное сжатие.  
 Формат BCD –  
**B**inary **C**oded **D**ecimal

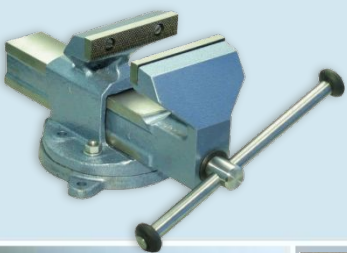
### Пример №2

TO BE OR NOT TO BE?

<b>T</b>	<b>O</b>	<b>B</b>	<b>E</b>	<b>R</b>	<b>N</b>	пробел	<b>?</b>
000	001	010	011	100	101	110	111

19 символов в предложении:  $3 \cdot 19 = 57$  бит = 8 байт  
 Коэффициент сжатия:  $19/8 \approx 2,4$





# Практическая работа

Снежная королева - Microsoft Word

Файл Правка Вид Вставка Формат Сервис Таблица Окно Справка

Правписание... F7  
Справочные материалы... Alt+щелчок  
Язык  
Статистика...  
Общая рабочая область...  
Письма и рассылки  
Настройка...  
Параметры...

Статистика

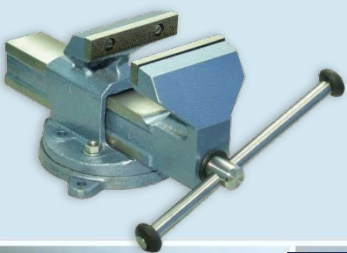
Статистика:	
Страниц	12
Слов	8 229
Знаков (без пробелов)	40 082
Знаков (с пробелами)	48 314
Абзацев	255
Строк	758

Учитывать все сноски

Панель Закреть

в которой рассказ  
Ну, начнем! Дойдя до кон  
сейчас. Так вот, жил-был тролль  
особенно хорошем расположении  
доброе и прекрасное уменьшалос  
выпирало, делалось еще гаже. П  
шпинатом, а лучшие из людей -  
ногами, а животов у них все  
у кого была веснушка, то уж бу  
губы. А если у человека являле  
ужимкой, что тролль так и пока  
Ученики тролля - а у него  
сотворилось чудо: теперь тольк  
их истинном свете. Они бежали  
страны, ни одного человека, которые не отразились бы в нем в искаженном виде.  
Напоследок захотелось им добраться и до неба. Чем выше они поднимались,  
тем сильнее кривлялось зеркало, так что они еле удерживали его в руках. Но во  
они взлетели совсем высоко, как вдруг зеркало до того перекорежило от гримас,  
что оно вырвалось у них из рук, полетело на землю и разбилось на миллионы,  
биллионы осколков, и оттого произошло еще больше бед. Некоторые осколки, с  
песчинку величиной, разлетаясь по белу свету, попадали людям в глаза, да так  
и оставались. А человек с таким осколком в глазу начинал видеть все навыворот  
или замечать в каждой вещи только дурное - ведь каждый осколок сохранял своис  
всего зеркала. Некоторым людям осколки попадали прямо в сердце, и это было





# Практическая работа

Снежная королева - Microsoft Word

Файл Правка Вид Вставка Формат Сервис Таблица Окно Справка

Отменить форматирование Ctrl+Z  
Вырезать Shift+Del  
Копировать Ctrl+Ins  
Буфер обмена Office...  
Вставить Shift+Ins  
Выделить все Ctrl+Num 5  
Найти... Ctrl+F  
Заменить... Ctrl+H

Снежная королева  
История первая,  
в которой рассказывается о зеркале и его осколках.

Ну, начнем! Дойдя до конца нашей истории, мы будем знать больше, чем сейчас. Так вот, жил-был тролль, злой-презлой, сущий дьявол. Раз был он в особенно хорошем расположении духа: смастерил такое зеркало, в котором все

Найти и заменить

Найти | Заменить | Перейти

Найти: к  
Параметры: Вперед

Выделить все элементы, найденные в: Найдено элементов: 1612

Текущий фрагмент Больше Найти все Закрыть

...ак и  
...сли  
...ой  
...ей в  
...ной  
...е.  
...от  
...и,  
...иона осколков, и оттого произошло еще больше бед: некоторые осколки, с  
...песчинку величиной, разлетаясь по белу свету, попадали людям в глаза, да так там  
...и оставались. А человек с таким осколком в глазу начинал видеть все наизуворот  
...или замечать в каждой вещи только дурное - ведь каждый осколок сохранял свойство  
...всего зеркала. Некоторым людям осколки попадали прямо в сердце, и это было



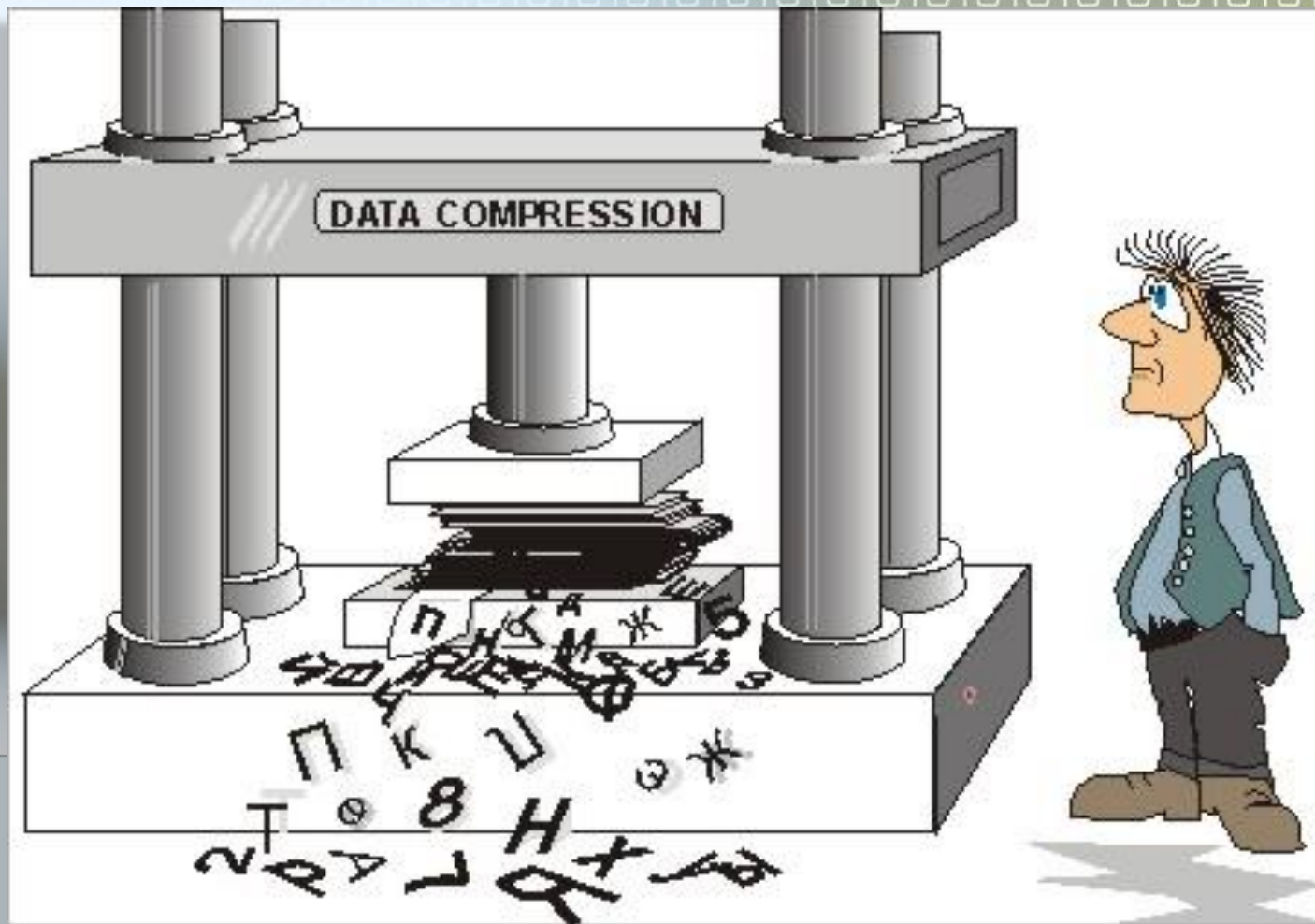


# Практическая работа

## “Частотный анализ букв русского языка”

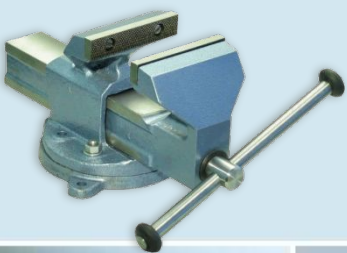
1. Открыть с помощью Microsoft Office Word документ Skazka.doc.
2. Подсчитать количество каждой из букв русского алфавита в тексте и заполнить таблицу Alphabet.xls в Microsoft Office Excel. Вычислить частоту встречаемости букв.
3. Упорядочить данные таблицы по убыванию частот.
4. Построить график распределения частот букв.





*АЛГОРИТМ*

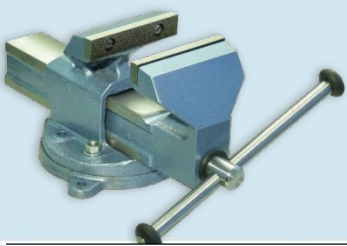
*Хаффманна*



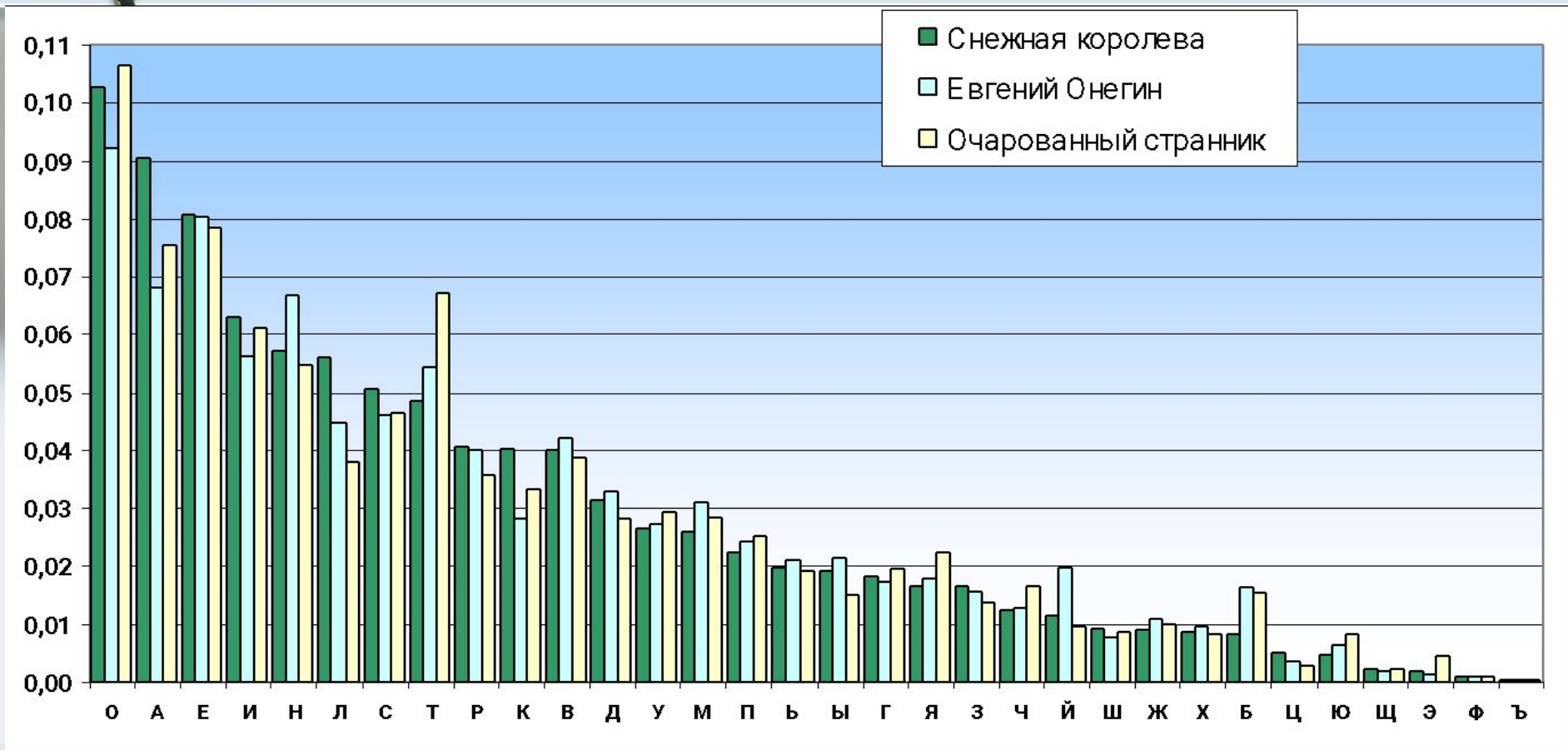
# Частотный анализ







# Сравнительный частотный анализ



«Анна Каренина»	оeanитслврқдмпупьяыгбчзжйшхэющцфь	280 тыс. слов
Солженицын А.И.	оeaинтсвлрлдмпупьяыгбзчйхжшюцщэфь	86 тыс. слов
Новости	оeaинтсрвлқдмпупьяыгзбчйжхющцщэфьё	25 тыс. слов



В тексте, написанном на русском языке, в каждой тысяче символов в среднем будет 90 букв "о", 72 - "е" и только 2 - "ф". Больше же всего окажется пробелов: 174.



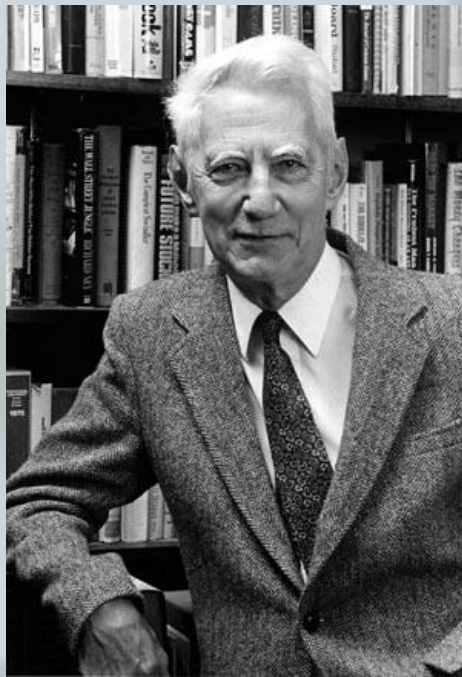
# Алгоритм

## Шеннона-Фано и Хаффмана

Оба этих алгоритма используют коды переменной длины: часто встречающийся символ кодируется двоичным кодом меньшей длины, редко встречающийся — кодом большей длины. Коды Шеннона-Фано и Хаффмана — префиксные, то есть никакое кодовое слово не является началом любого другого. Это свойство позволяет однозначно декодировать любую последовательность кодовых слов. В отличие от алгоритма Шеннона-Фано, алгоритм Хаффмана обеспечивает минимальную избыточность, то есть минимальную длину кодовой последовательности при побайтном кодировании.



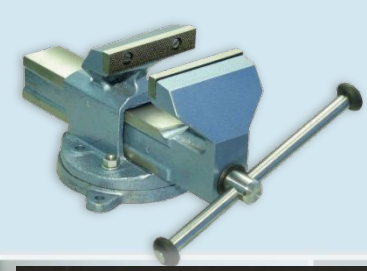
# Алгоритм Шеннона-Фано



К.Шеннон и Р.Фано  
сформулировали  
алгоритм сжатия,  
который использует  
коды переменной длины.

	%		%		%
<b>A</b>	8.1	<b>K</b>	0.4	<b>U</b>	2.4
<b>B</b>	1.4	<b>L</b>	3.4	<b>V</b>	0.9
<b>C</b>	2.7	<b>M</b>	2.5	<b>W</b>	1.5
<b>D</b>	3.9	<b>N</b>	7.2	<b>X</b>	0.2
<b>E</b>	13.0	<b>O</b>	7.9	<b>Y</b>	1.9
<b>F</b>	2.9	<b>P</b>	2.0	<b>Z</b>	0.1
<b>G</b>	2.0	<b>Q</b>	0.2		
<b>H</b>	5.2	<b>R</b>	6.9		
<b>I</b>	6.5	<b>S</b>	6.1		
<b>J</b>	0.2	<b>T</b>	10.5		





# Дэвид Хаффман

## David Huffman

(1925-1999)

В 18 лет Дэвид получил степень бакалавра электротехники в университете штата Огайо. Основную концепцию кодирования данных Хаффман разработал во время Второй мировой войны, когда служил на эсминце офицером-связистом. Изначально алгоритм предназначался для кодирования радиосообщений.

За свою деятельность он получил множество наград за исключительный вклад в теорию информации.



# Таблица кодов Хаффмана

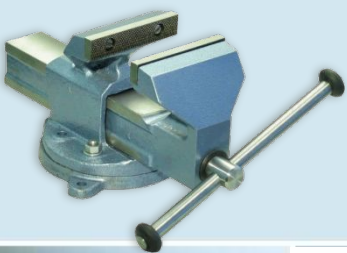
Буква	Код Хаффмана	Буква	Код Хаффмана	Буква	Код Хаффмана
<b>E</b>	100	<b>D</b>	11011	<b>W</b>	011101
<b>T</b>	001	<b>L</b>	01111	<b>B</b>	011100
<b>A</b>	1111	<b>F</b>	01001	<b>V</b>	1101001
<b>O</b>	1110	<b>C</b>	01000	<b>K</b>	110100011
<b>N</b>	1100	<b>M</b>	00011	<b>X</b>	110100001
<b>R</b>	1011	<b>U</b>	00010	<b>J</b>	110100000
<b>I</b>	1010	<b>G</b>	00001	<b>Q</b>	1101000101
<b>S</b>	0110	<b>Y</b>	00000	<b>Z</b>	1101000100
<b>H</b>	0101	<b>P</b>	110101		



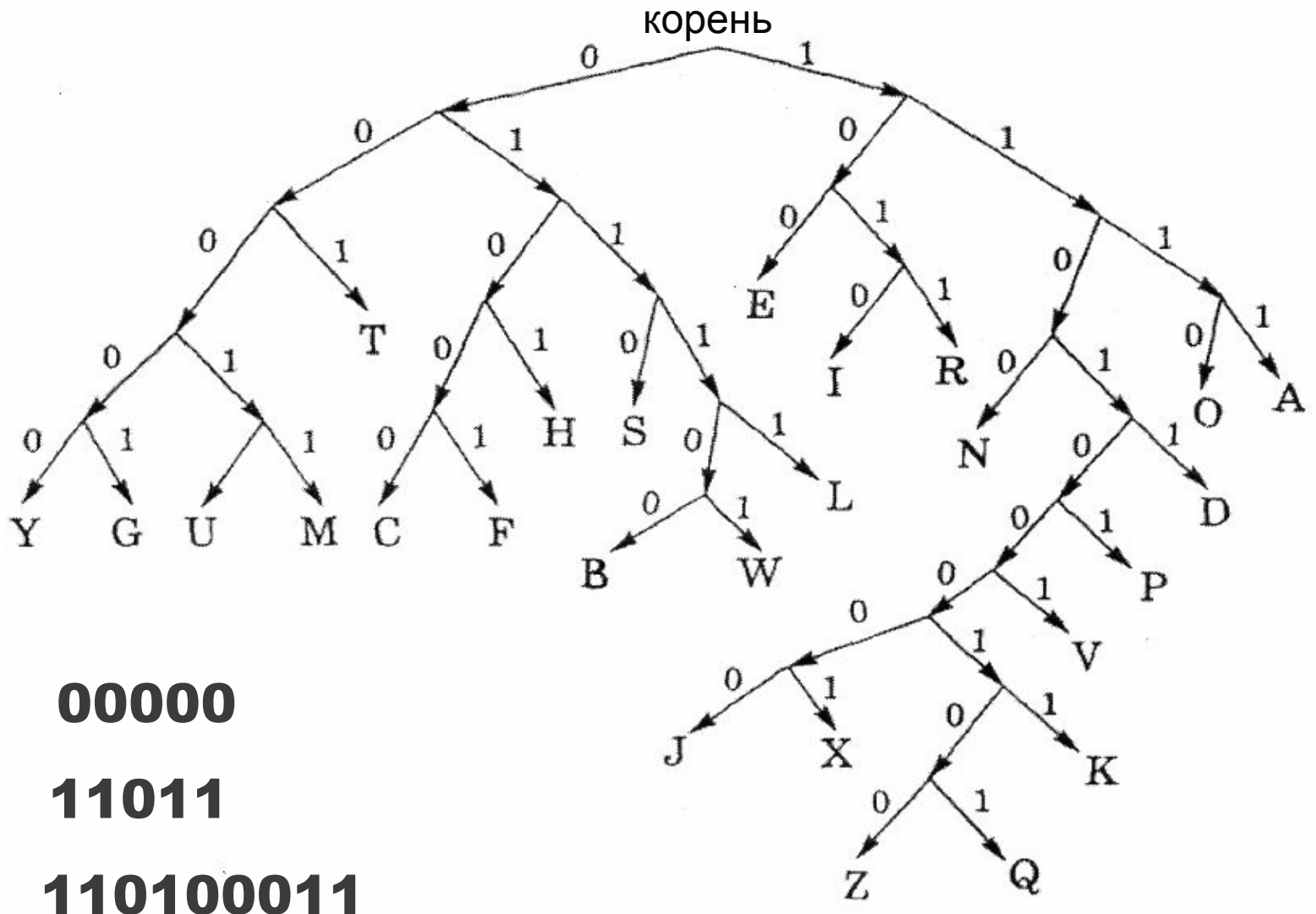






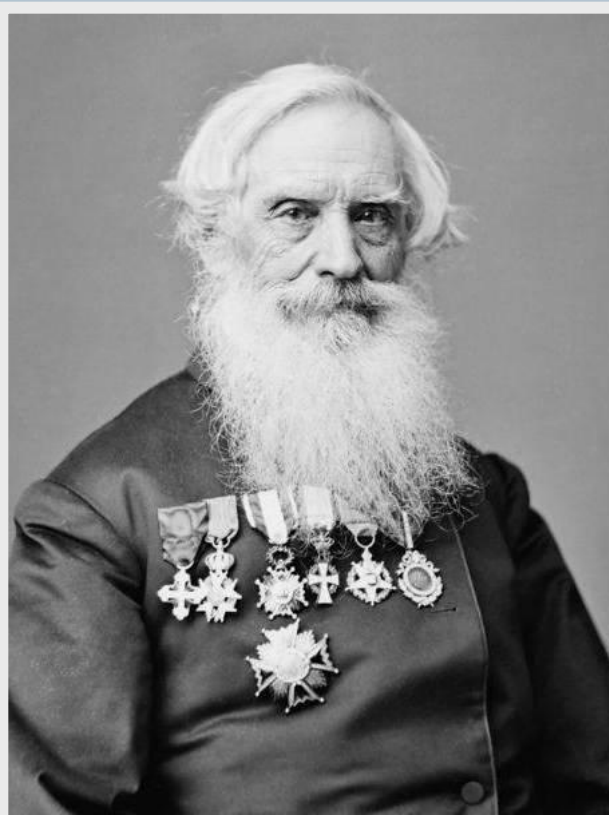


# Дерево Хаффмана





# Азбука Морзе

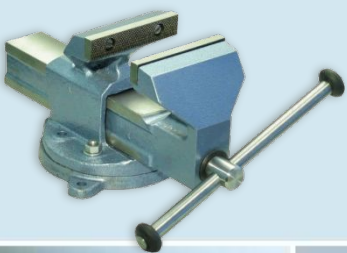


Сэмюэль Морзе  
(1791-1872) - американский  
изобретатель и художник

## INTERNATIONAL MORSE CODE

1. A dash is equal to three dots.
2. The space between parts of the same letter is equal to one dot.
3. The space between two letters is equal to three dots.
4. The space between two words is equal to five dots.

A	• —	U	• • —
B	— • • •	V	• • • —
C	— • — •	W	— • —
D	— • •	X	— • • —
E	•	Y	— • — —
F	• • — •	Z	— — • •
G	— — •		
H	• • • •		
I	• •		
J	• — — —		
K	— • —	1	• — — — —
L	• — • •	2	• • — — —
M	— —	3	• • • — —
N	— •	4	• • • • —
O	— — —	5	• • • • •
P	• — — •	6	— • • • •
Q	— — • —	7	— — • • •
R	• — •	8	— — — • •
S	• • •	9	— — — — •
T	—	0	— — — — —



# Азбука Морзе

А	• —	П	• — — •	Ь	— • • —
Б	— • • •	Р	• — •	Ы	— • —
В	• — — —	С	• • •	Й	• — — —
Г	— — — •	Т	—		
Д	— • •	У	• • — •	1	• — — — —
Е	•	Ф	• • — •	2	• • — — —
Ж	• • • —	Х	• • • •	3	• • • — —
З	— — — • •	Ц	— • — •	4	• • • • —
И	• •	Ч	— — — — •	5	• • • • •
К	— • — — •	Ш	— — — — —	6	— • • • •
Л	• — — • •	Щ	— — — • —	7	— — — • • •
М	— — —	Э	• • — — • •	8	— — — — • •
Н	— •	Ю	• • — — —	9	— — — — — •
О	— — — —	Я	• — — • —	0	— — — — —

• • • — • — • — •

• • • — • — • — •

• • • — • — • — •

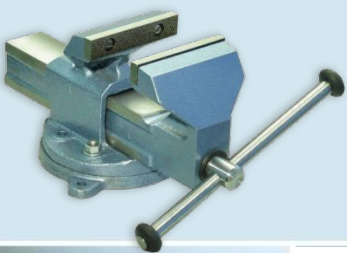
• • • — • — • — •

с т е к

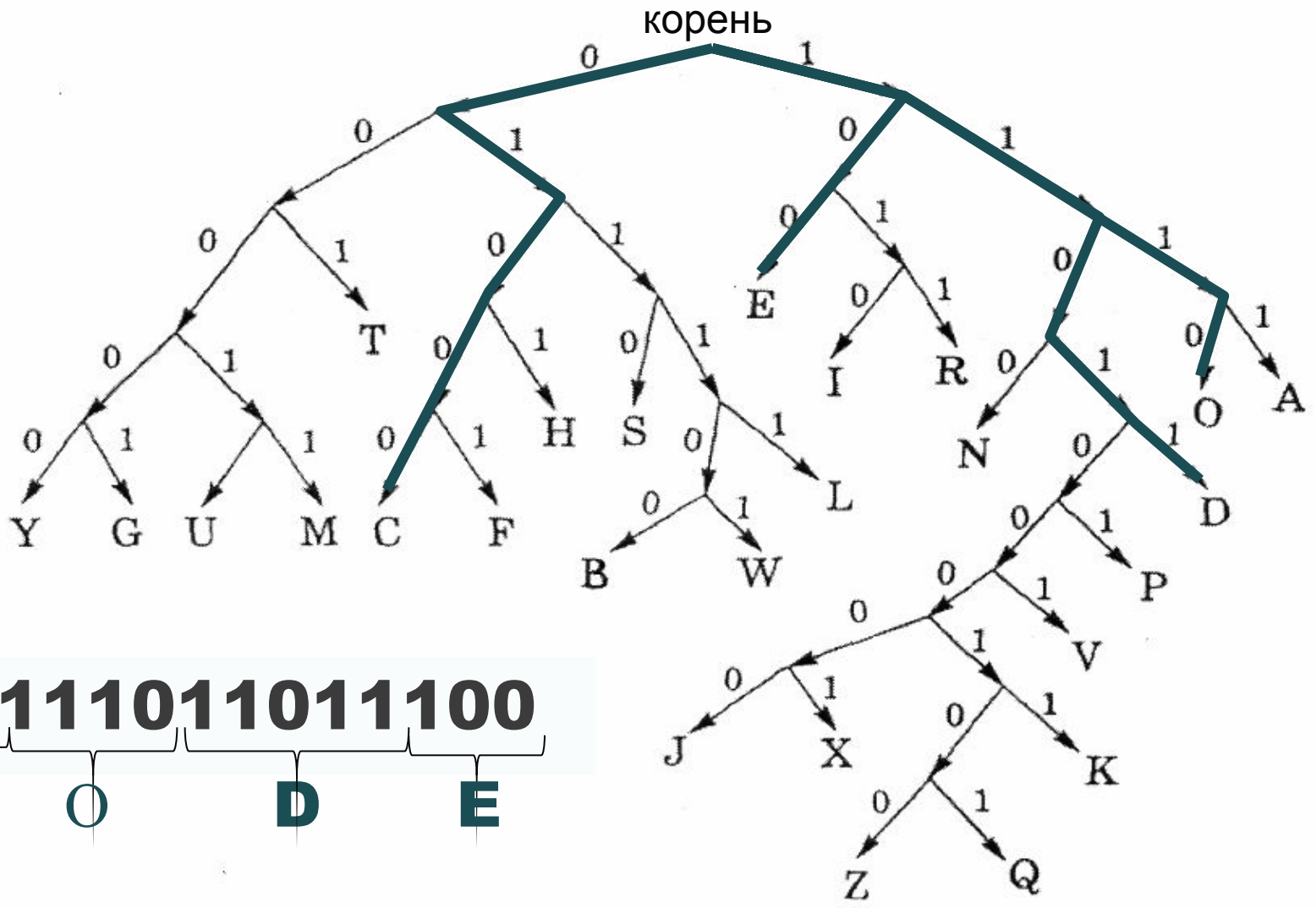
ж а р

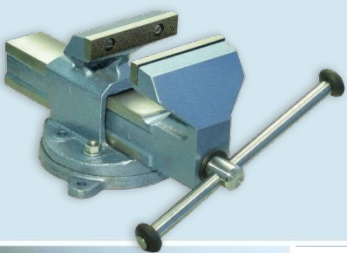
е ф т а е



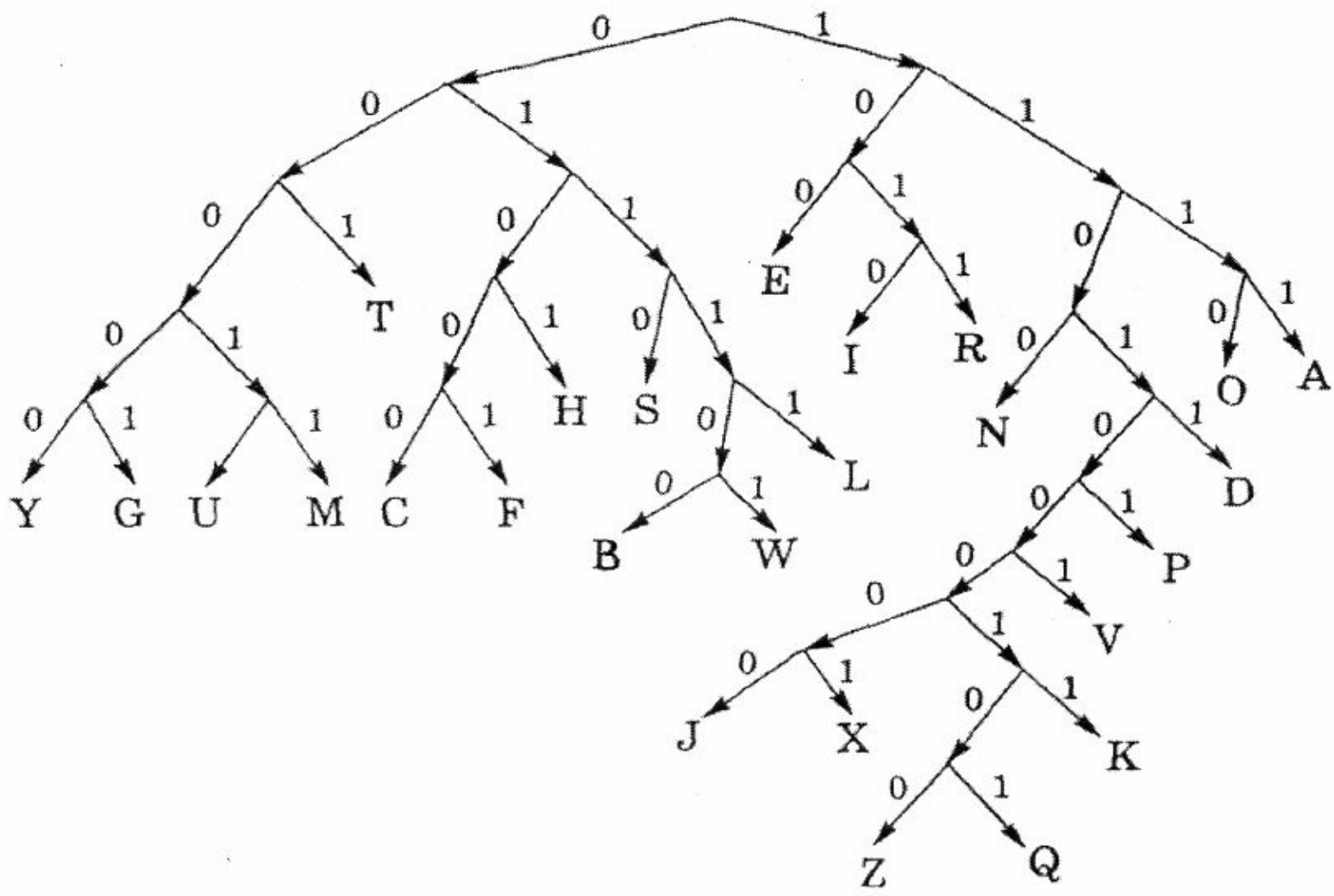


# Дерево Хаффмана



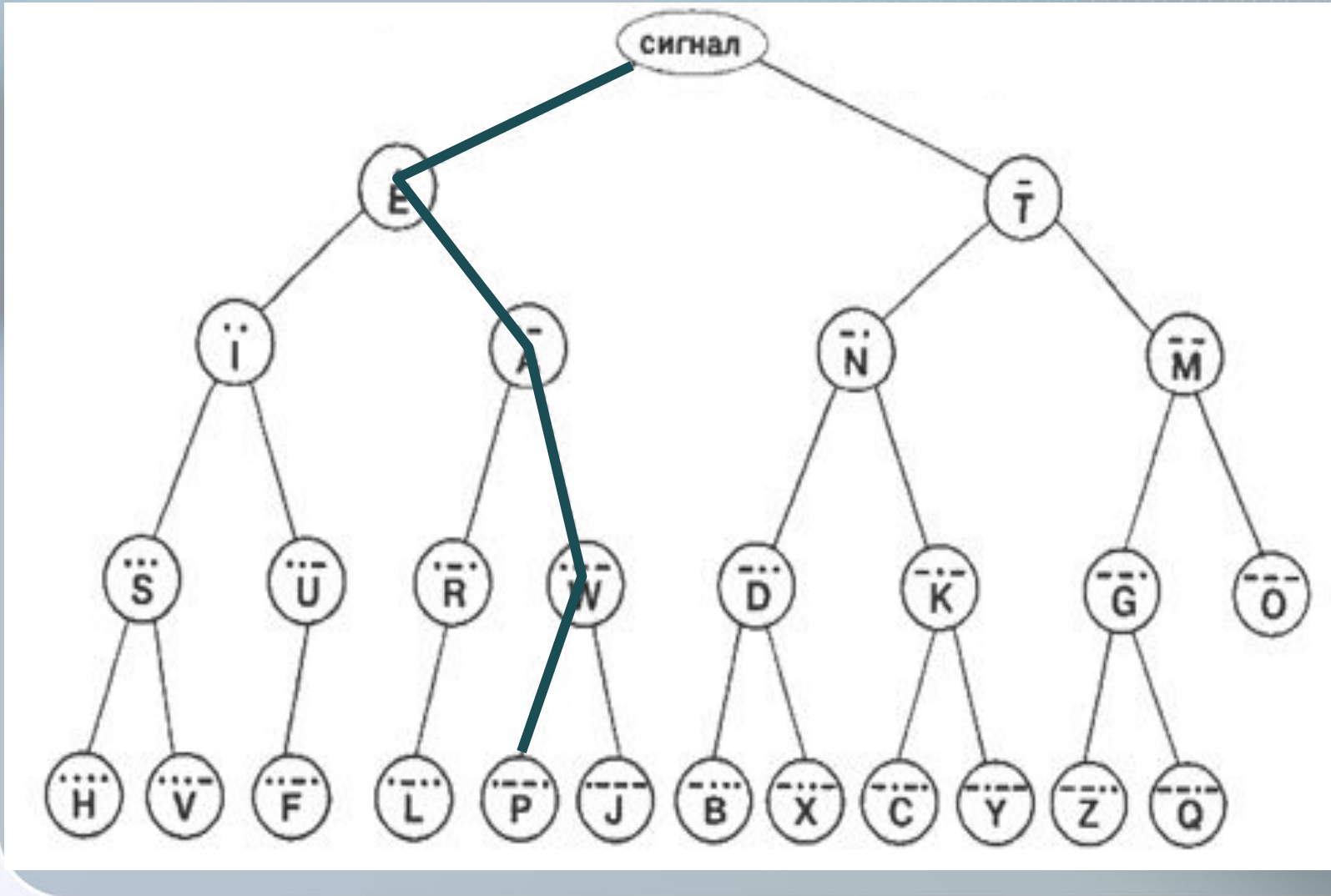


Код Хаффмана обладает свойством **префиксности**, то есть код никакого символа не является началом кода какого-либо другого символа.





# Дерево азбуки Морзе





# Алгоритм Хаффмана

Алгоритм Хаффмана двухпроходный:

на **первом** проходе строится частотный словарь и генерируются коды;

на **втором** проходе происходит непосредственно кодирование.

НА ДВОРЕ ТРАВА, НА ТРАВЕ ДРОВА

1. Подсчитать частоту встречаемости (вес) каждого символа.

СИМВОЛ	А	В	Д	,	Е	Н	Р	О	Т	_
вес	6	4	2	1	2	2	4	2	2	5

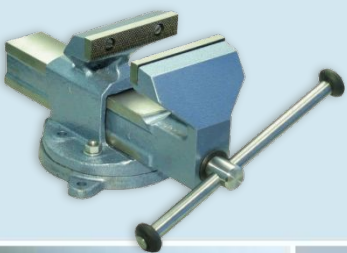


# Алгоритм построения дерева Хаффмана

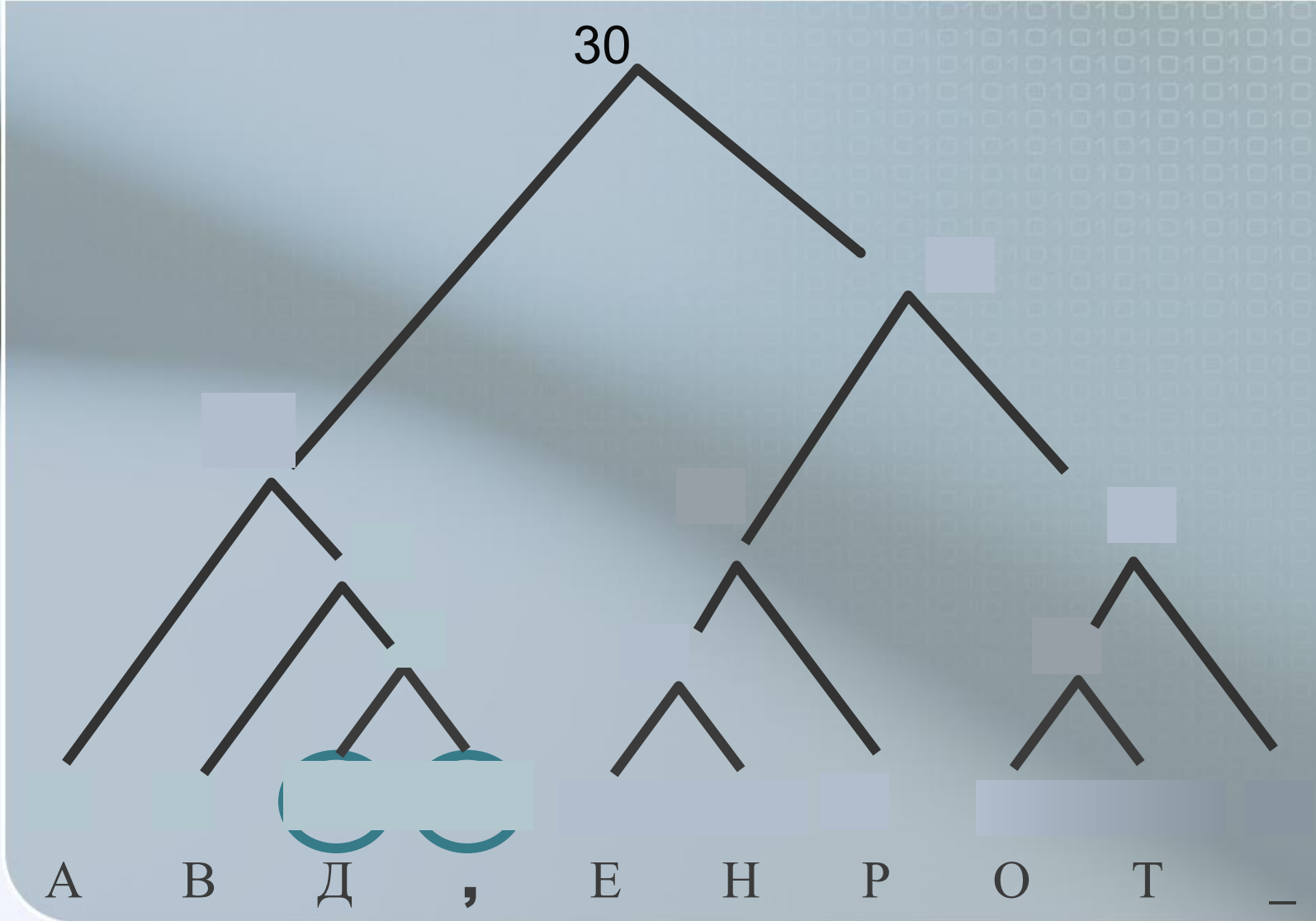
1. Среди символов выбрать два с наименьшими весами (если таких пар несколько, выбирается любая из них).
2. Свести ветки дерева от этих двух символов в одну точку с весом, равным сумме двух весов, при этом веса самих элементов стираются.
3. Повторять пункты 1 и 2 до тех пор, пока не останется одна вершина с весом, равным сумме весов исходных символов.
4. Пометить одну ветку нулём, а другую - единицей (по собственному усмотрению).

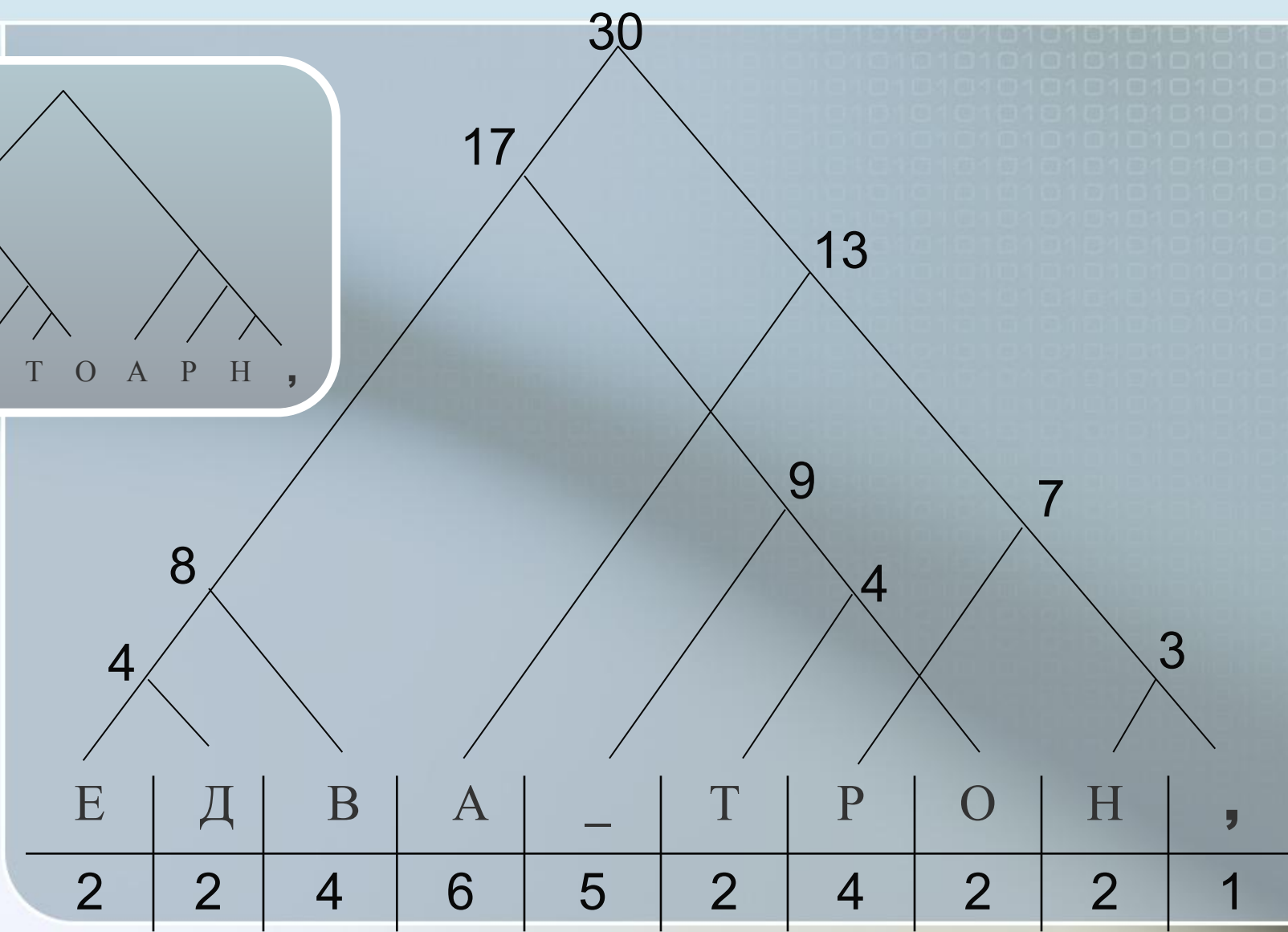


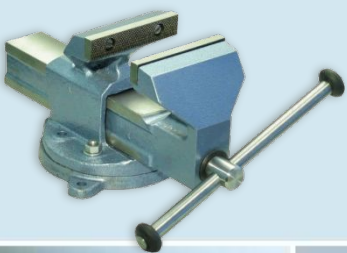




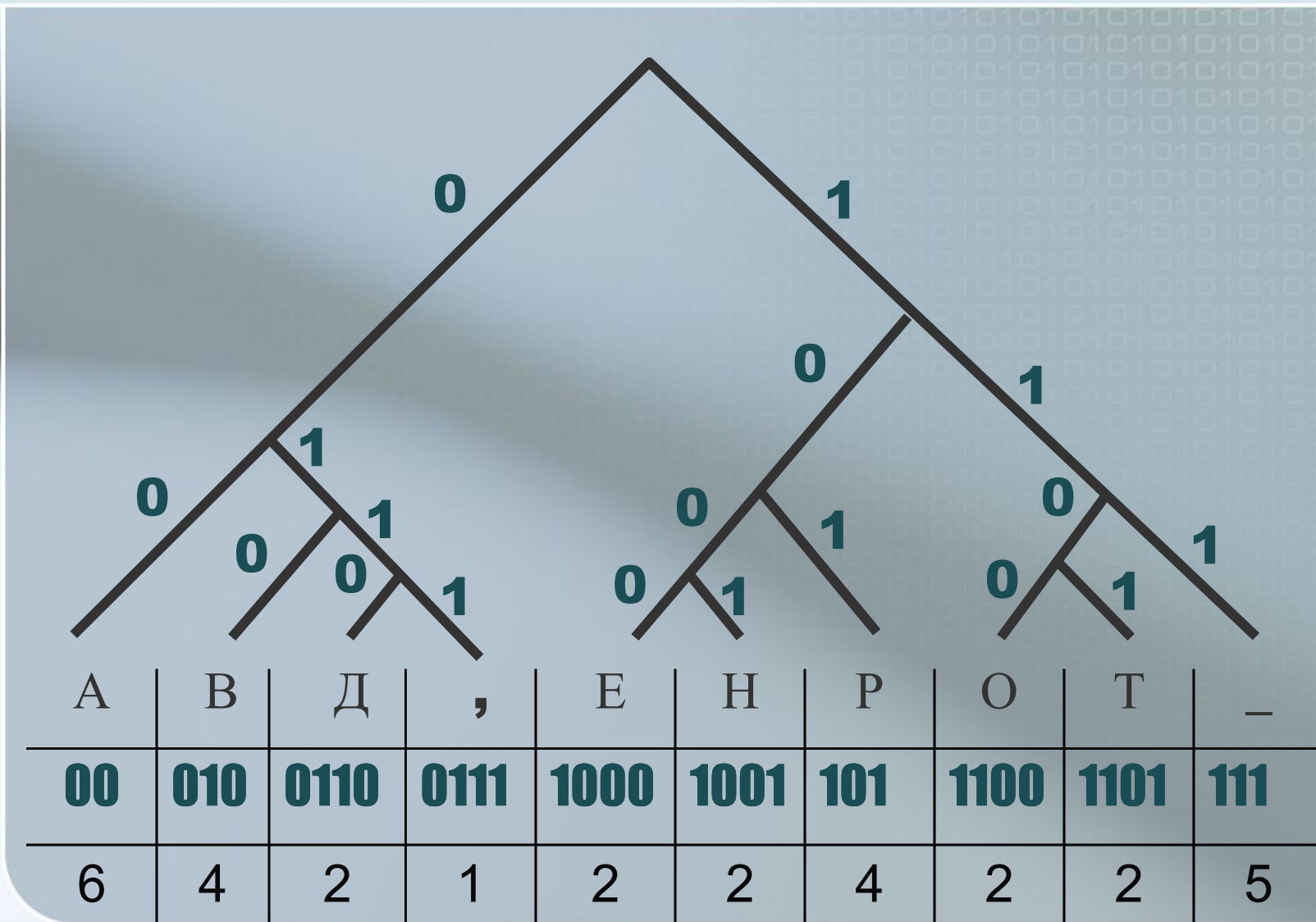
# Построение дерева Хаффмана

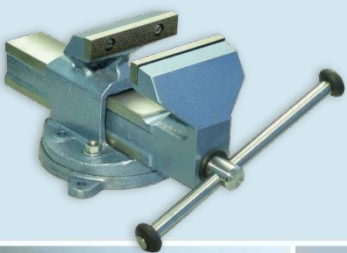






# Построение дерева Хаффмана





# Кодирование текста

А	В	Д	,	Е	Н	Р	О	Т	_
<b>00</b>	<b>010</b>	<b>0110</b>	<b>0111</b>	<b>1000</b>	<b>1001</b>	<b>101</b>	<b>1100</b>	<b>1101</b>	<b>111</b>

НА ДВОРЕ ТРАВА, НА ТРАВЕ ДРОВА

**1001001110110010110010110001111101**

Н А \_ Д В О Р Е \_ Т

**10100010000111111001001111101101**

Р А В А , \_ Н А \_ Т Р

**0001010001110110101110001000**

А В Е \_ Д Р О В А





# Кодирование текста

А	В	Д	,	Е	Н	Р	О	Т	_
<b>00</b>	<b>010</b>	<b>0110</b>	<b>0111</b>	<b>1000</b>	<b>1001</b>	<b>101</b>	<b>1100</b>	<b>1101</b>	<b>111</b>

НА ДВОРЕ ТРАВА, НА ТРАВЕ ДРОВА

**10010011101100101100101100011111**

**01101000100001111111001001111101**

**1010001010001110110101110001000**





# Коэффициент сжатия

Коэффициентом сжатия называется отношение объема исходного сообщения к объему сжатого.

СИМВОЛ	А	В	Д	,	У	Н	Р	О	Т	_
КОД	00	010	0110	0111	1000	1001	101	1100	1101	111
ВЕС	6	4	2	1	2	2	4	2	2	5

Объем **сжатого** сообщения:

$$6*2+4*3+2*4+1*4+2*4+2*4+4*3+2*4+2*4+5*3=95 \text{ бит}=12 \text{ байт.}$$

Объем **исходного** сообщения в ASCII равен 30 байт.

$$\text{Коэффициент сжатия составляет } 30 / 12 = \mathbf{2,5}$$





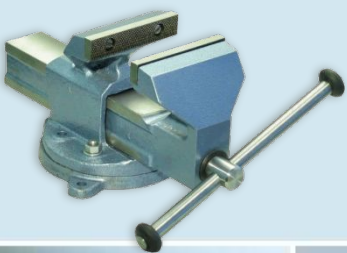
# Декодирование

Восстановить исходный текст:

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

СИМВОЛ	А	В	Д	,	У	Н	Р	О	Т	_
КОД	00	010	0110	0111	1000	1001	101	1100	1101	111

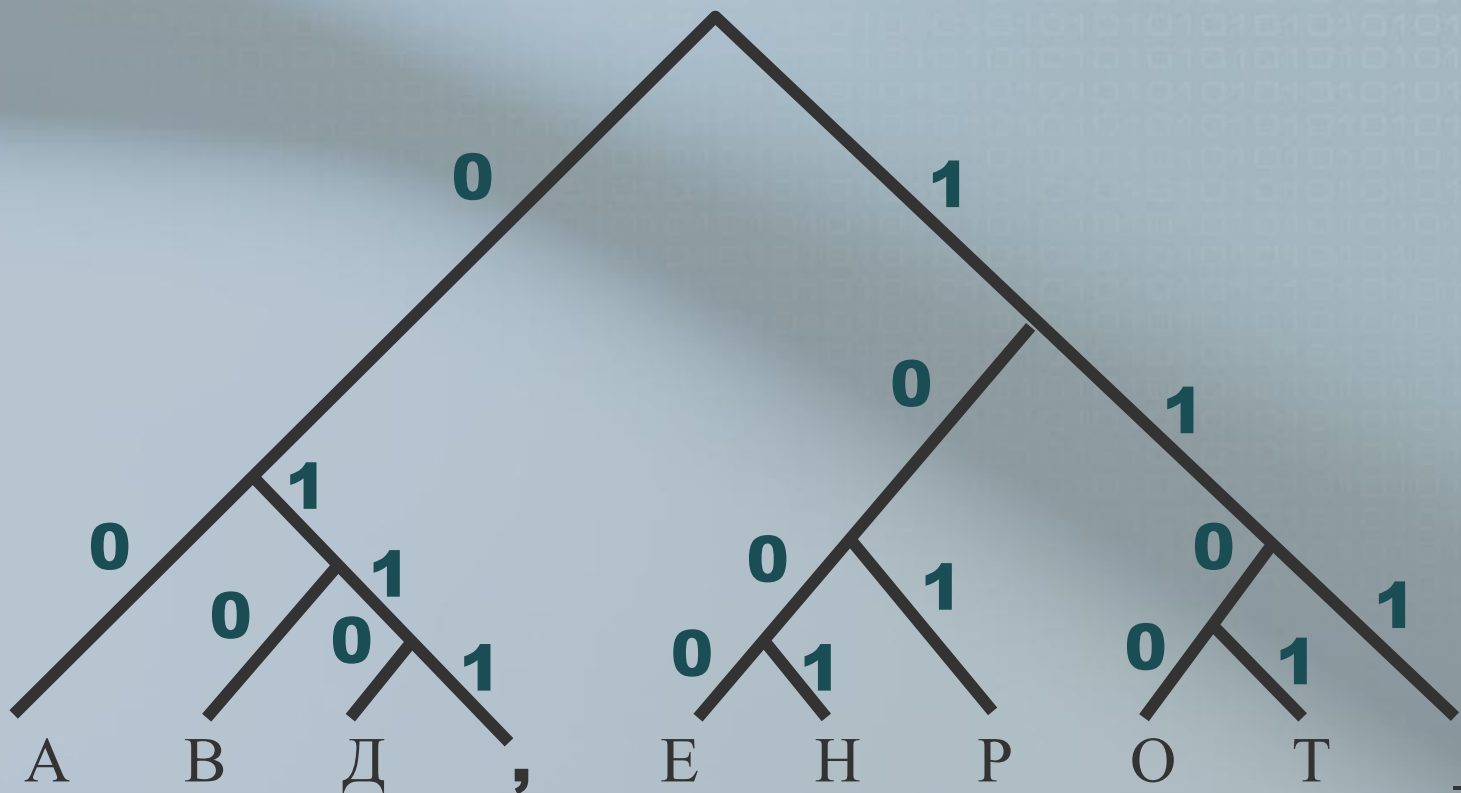




# Декодирование

1 0 0 1 0 0 1 0 1 1 1 0 0 0 1 1 0

Н А Р О Д







# Самостоятельная работа

Постройте код Хаффмана для предложения:

**TO BE OR NOT TO BE?**

Определите коэффициент сжатия для данной фразы, если каждый символ кодируется в ASCII.

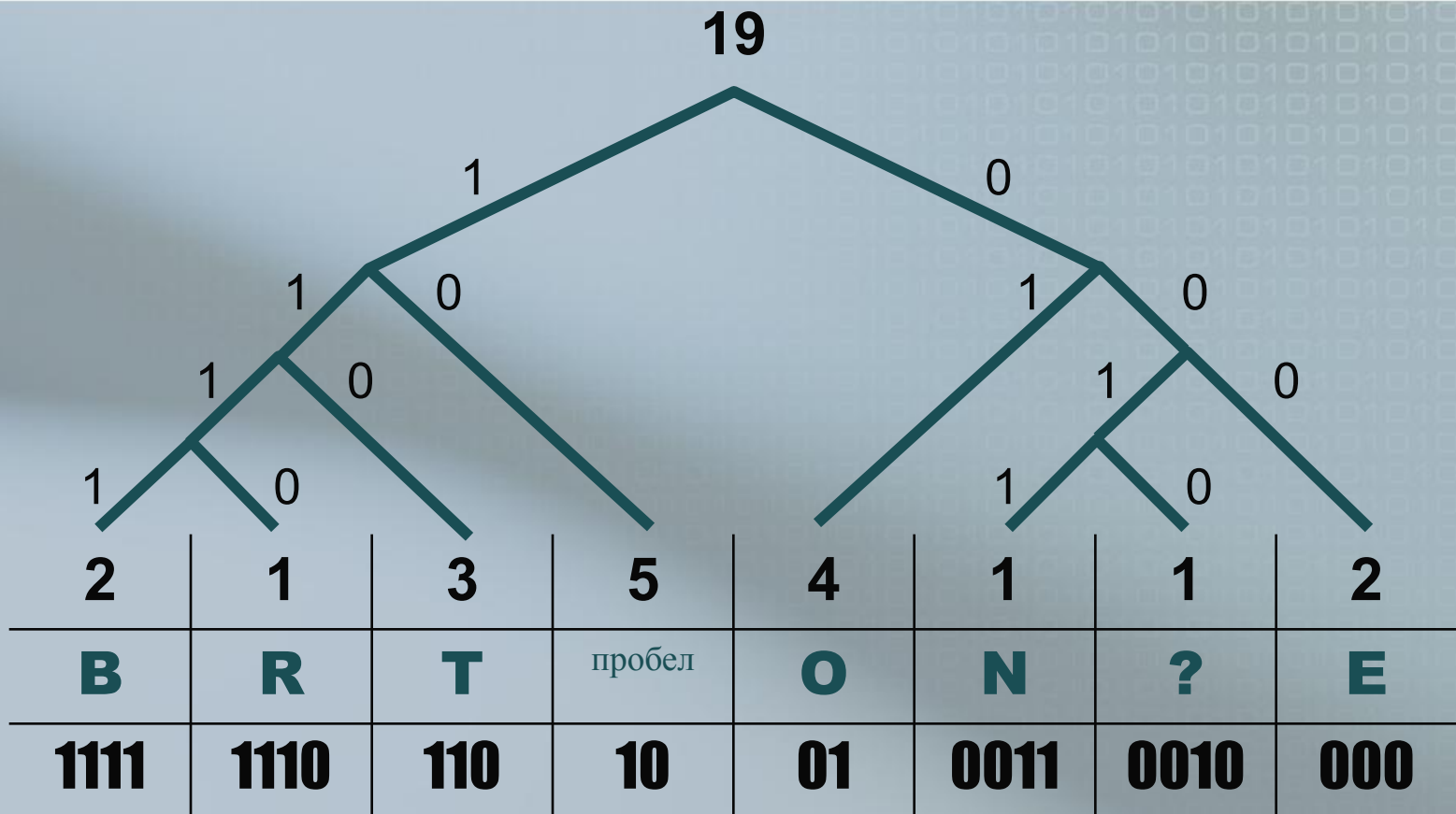
Сравните результат с тем, что был получен при сжатии фразы методом упаковки, сделайте выводы.

Проверьте правильность выполнения задания: закодируйте предложение, используя полученный код, а затем попробуйте его восстановить.





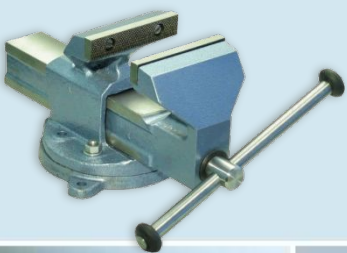
# Одно из решений



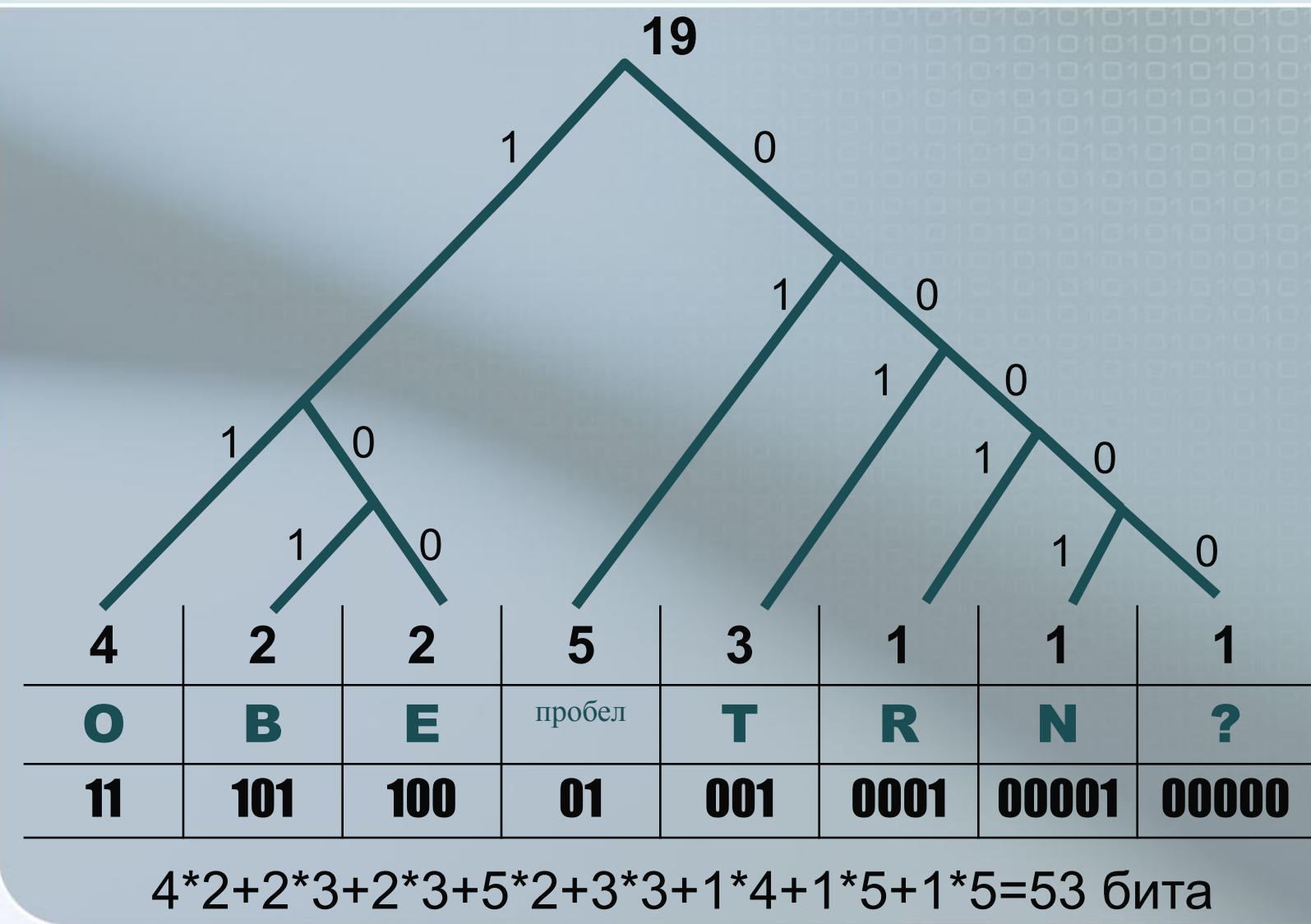
$2*4+1*4+3*3+5*2+4*2+1*4+1*4+2*3=53$  бита=7 байт

Коэффициент сжатия:  $19/7 \approx 2,7$

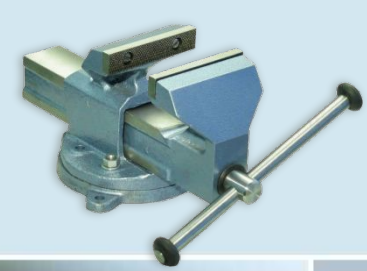




# Другое решение







# задание

## Задание №1

Постройте код Хаффмана для фраз:

**1) Человек как музыкальный инструмент, как настроишь, так и живет.**

**2) Музыка показывает человеку те возможности величия, которые есть в его душе.**

Определите коэффициент сжатия для данной фразы, если каждый символ кодируется в ASCII.





# Домашнее задание

## Задание №2

На языке Си++ напишите программу, реализующую алгоритм RLE для текстовых данных.

Исходные данные в виде строки, содержащей латинские буквы и пробелы, находятся в текстовом файле input.txt (длина строки не более 255). Результат должен выводиться в текстовый файл output.txt, первая строка которого содержит сжатую строку, вторая – коэффициент сжатия с точностью до сотых.

Пример файла:

```
LLLLLLLLLEESSSSSSSSSSoooooooooooooNN    one
```

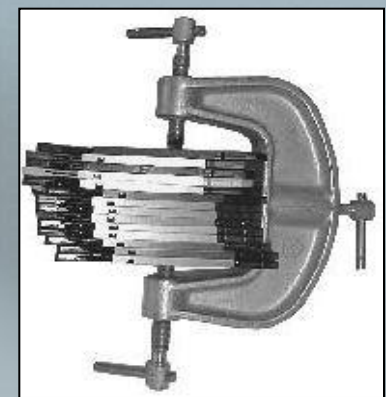
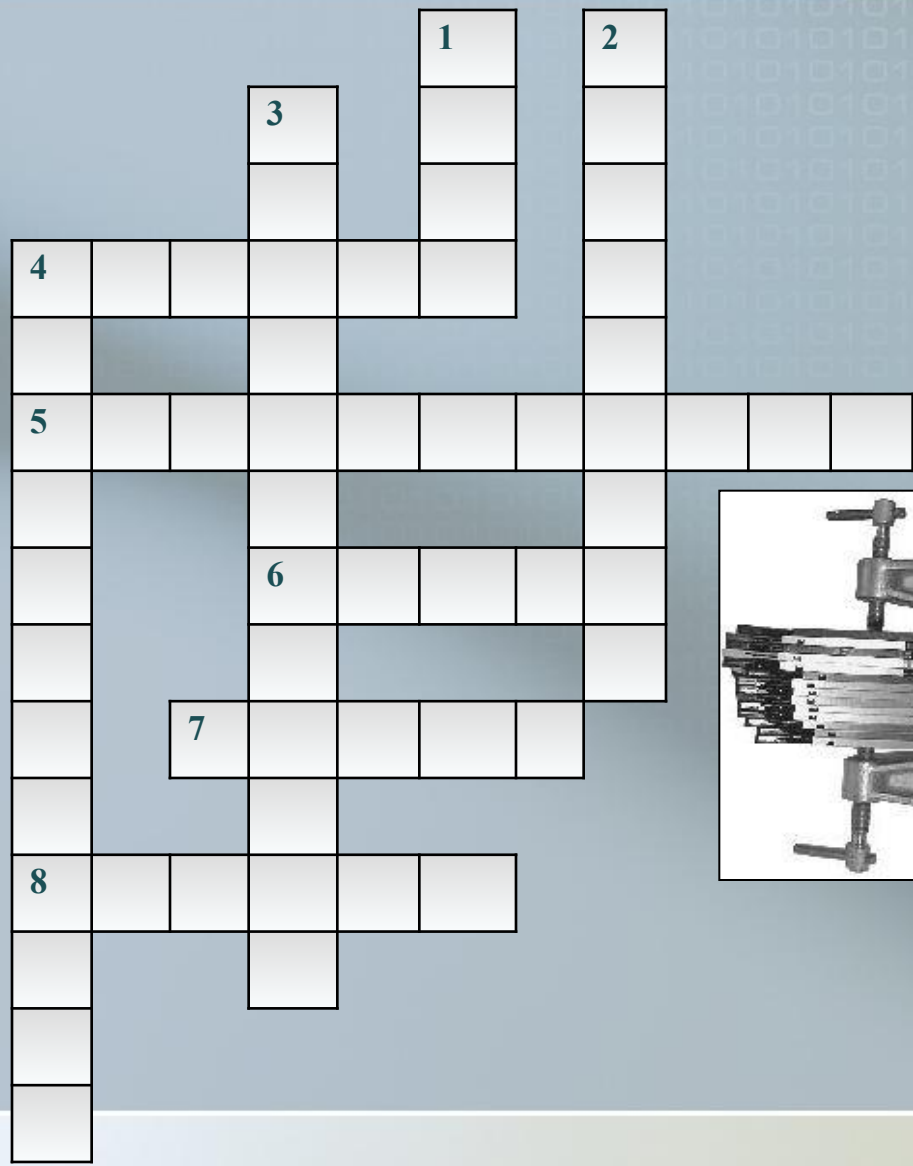
Ответ:

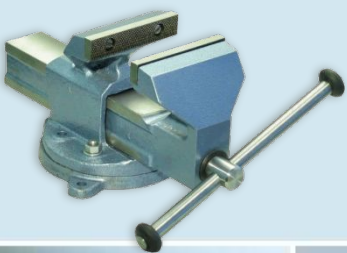
```
8L2E9S12o2N8 1o1n1e  
2.32
```





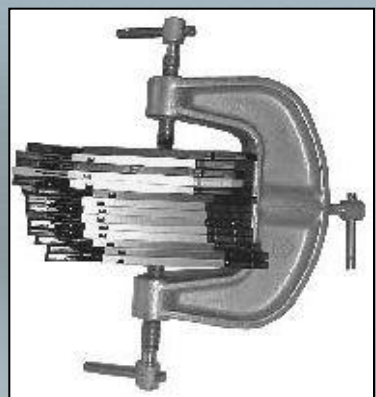
# Кроссворд





# Кроссворд

					ф	а				
			п		а	р				
			р		н	х				
д	е	р	е	в	о	и				
е			ф			в				
к	о	д	и	р	о	в	а	н	и	е
о			к				т			
м			с	л	о	в	о			
п			н				р			
р		м	о	р	з	е				
е			с							
с	ж	а	т	и	е					
с			ь							
и										
я										







# Литература

1. *Андреева Е.В.* Математические основы информатики. Элективный курс: учебное пособие / Е.В.Андреева, Л.Л.Босова, И.Н.Фалина – М.:БИНОМ. Лаборатория знаний, 2005.
2. *Гейн А.Г.* Математические основы информатики. / «Информатика» №17 / 2007
3. *Семакин И.Г.* Информатика. Базовый курс. 7-9 классы / И.Г.Семакин, Л.А.Залогова, С.В.Русаков, Л.В.Шестакова. – 2-е изд., испр. И доп. – М.: БИНОМ. Лаборатория знаний, 2004.
4. *Семакин И.Г.* Информатика и ИКТ. Базовый уровень : практикум для 10-11 классов / И.Г.Семакин, Е.К.Хеннер, Т.Ю.Шеина – М. : БИНОМ. Лаборатория знаний, 2007.
5. <http://www.compression.ru/> сайт «Все о сжатии»
6. *Устинов П.С.* Архиватор собственными руками!  
[http://vio.fio.ru/vio\\_09/resource/Print/art\\_1\\_9.htm](http://vio.fio.ru/vio_09/resource/Print/art_1_9.htm)

