

***СЖАТИЕ
ДВОИЧНОГО
КОДА***

**Сжатие данных – это процесс,
обеспечивающий
уменьшение объёма данных
за счёт изменения способа их
организации**

Возможны две ситуации при сжатии:

1. Потеря информации в результате сжатия недопустима.
2. Допустима частичная потеря информации в результате сжатия.

СЖАТИЕ С ЧАСТИЧНОЙ ПОТЕРЕЙ ИНФОРМАЦИИ

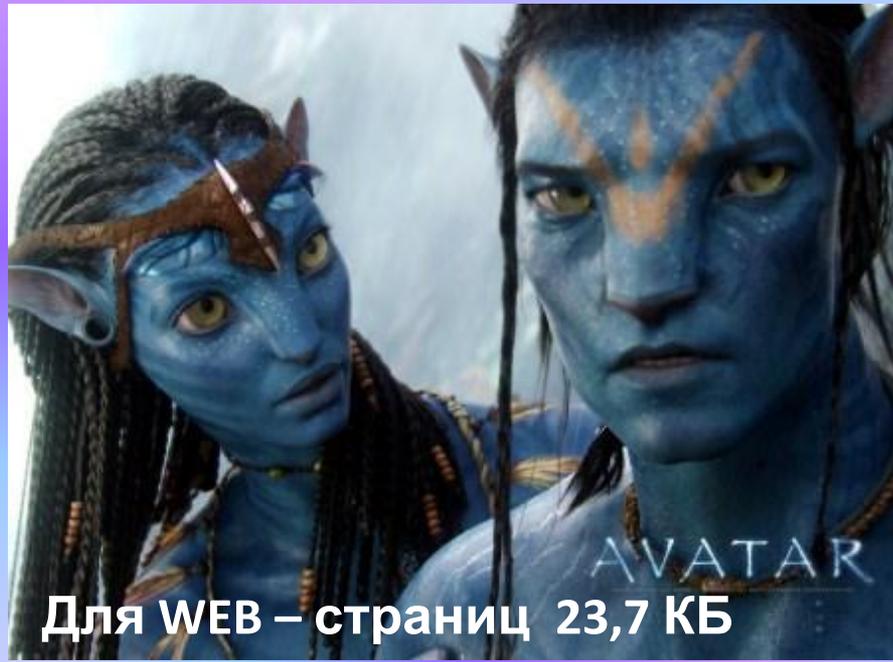
Графика, видео, звук

Связано с субъективными возможностями зрения человека.

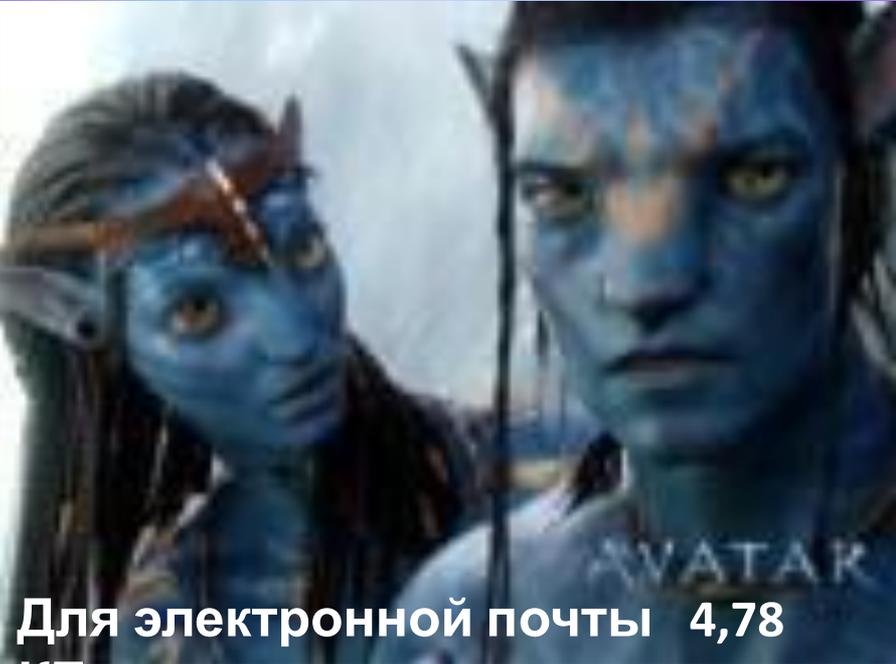
- Яркость важнее цвета. Объём сокращается за счёт того, что коды цвета хранятся не для каждой точки(через 1, через 2, ...). Чем больше сжатие, тем хуже качество
- При кодировании видеофильмов – свойство инерционности зрения (быстро меняющиеся фрагмента можно кодировать менее подробно, чем статические изображения)



Исходное 419 КБ



Для WEB – страниц 23,7 КБ



Для электронной почты 4,78 КБ

Связано с субъективными возможностями слуха человека.

- Учитывается восприимчивость слуха. Слабо воспринимаемые гармоники отфильтровываются путём математической обработки.

**СЖАТИЕ БЕЗ
ПОТЕРИ
ИНФОРМАЦИИ**

Использование неравномерного кода для сжатия текста

- В компьютере 1 символ – 8 бит (1 байт)
- Частота встречаемости символов

Таблица 1

Буква	пробел	о	е,ё	а	и	т	н	с	р	в	л	к	м	д	п	у
Частота	175	90	72	62	62	53	53	54	40	38	35	28	26	25	23	21
Буква	я	ы	з	ь,ъ	б	г	ч	й	х	ж	ю	ш	щ	э	ф	
Частота	18	16	16	14	14	13	12	10	9	7	6	6	4	3	3	2

- Чем чаще встречается символ, тем меньше его информационный вес.
- **Часто встречающиеся символы кодируют более коротким кодом.**

Алгоритм Дэвида Хаффмана

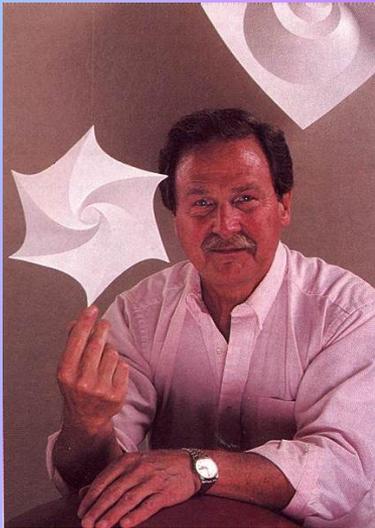


Таблица 5. Вариант кодовой таблицы Хаффмана

Буква	Код Хаффмана	Буква	Код Хаффмана	Буква	Код Хаффмана	Буква	Код Хаффмана
Е	100	Ѕ	0110	И	00010	К	110100011
Т	001	Н	0101	Г	00001	Х	110100001
А	1111	Д	11011	У	00000	Ж	110100000
О	1110	Л	01111	Р	110101	Ц	1101000101
Н	1100	Ф	01001	В	011101	З	1101000100
Р	1011	С	01000	Б	011100		
И	1010	М	00011	У	1101001		

WENEEDMORESNOWFORBETTERSKIING

Закодируем строку:

011101100110010010011011000111110101110001101100111001
110101001111010110111001000010011001011011011010001110
101010110000001

Переведём в шестнадцатеричный код, разместив побайтно (по 8)

84218421 84218421 84218421

01110110 01100100 10011011 00011111 01011100 01101100

76

64

9B

1F

5C

6C

11100111 01010011 11010110 11100100 00100110 01011011

E7

53

D6

E4

26

5B

01101000 11101010 10110000 00100000

68

EA

B0

20

Текст, занимающий в кодировке ASCII 29 байтов , в кодировке Хаффмана займёт 16 байтов.

Коэффициент сжатия = $\frac{\text{длина кода после сжатия}}{\text{длина кода до сжатия}}$

$$\frac{16}{29} \approx 0,55$$

Раскодирование происходит при помощи двоичного дерева Хаффмана

Дерево (граф) – графическое представление структуры связей между элементами некоторой системы.

Состоит из вершин и линий связи.

Если линии связи имеют направление, они называются дугой.

Двоичное дерево – любая вершина имеет не более двух потомков.

Корень дерева – единственная вершина, не имеющая родителей.

Листья – вершины, не имеющие потомков.

Дерево Хаффмана

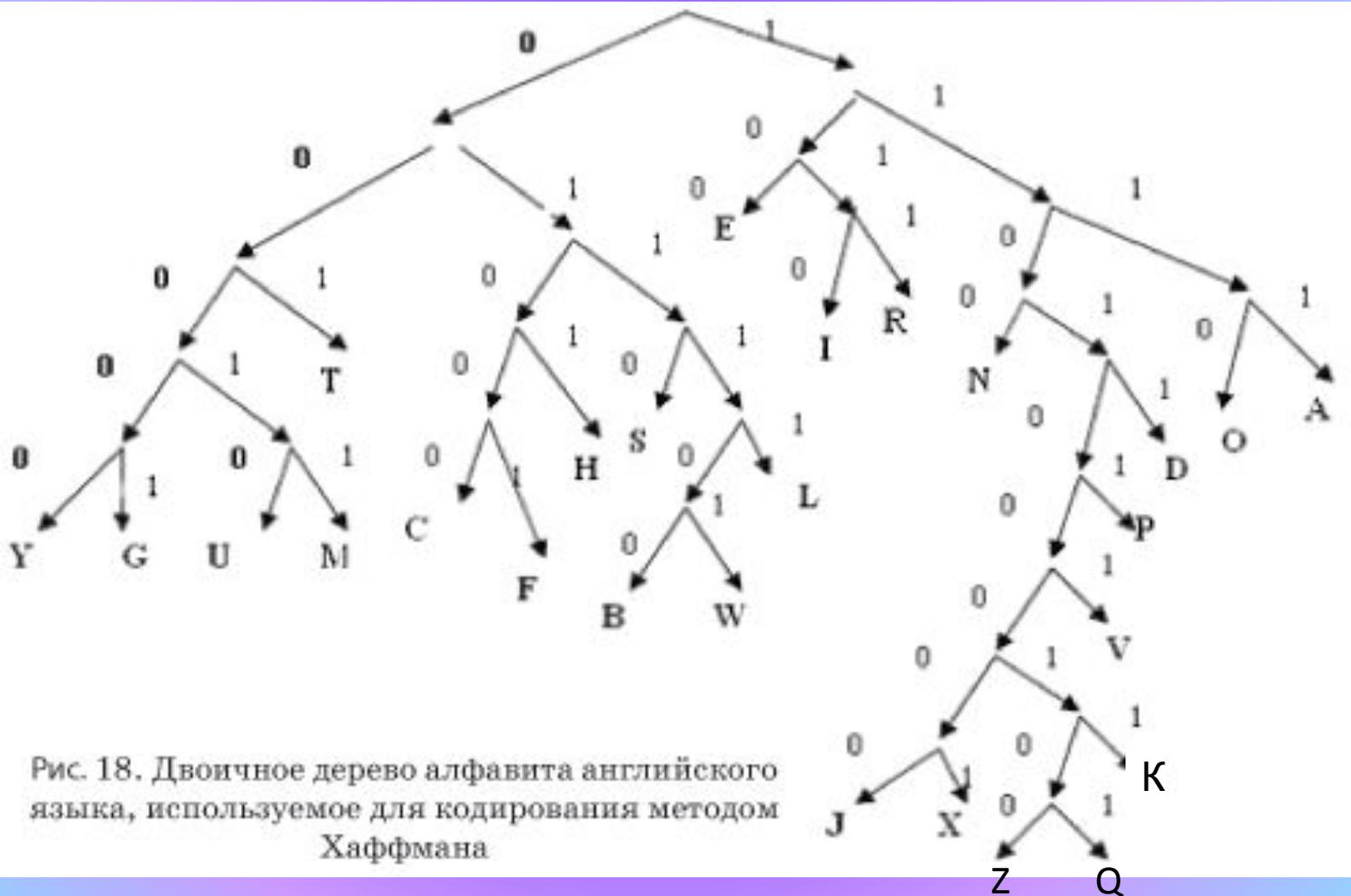
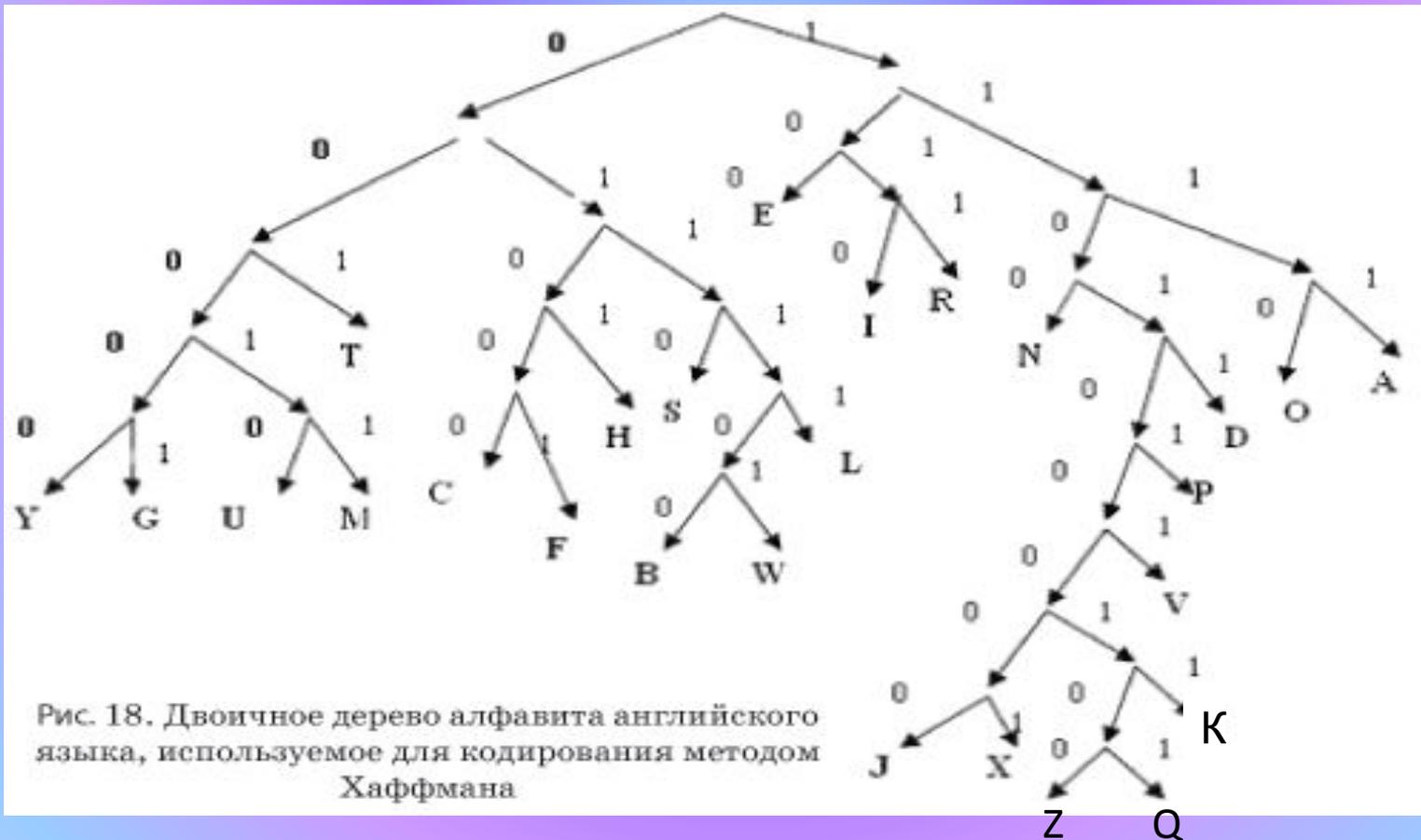


Рис. 18. Двоичное дерево алфавита английского языка, используемое для кодирования методом Хаффмана

Раскодировать двоичный код

01010001 00100101 00100011

11111100. Рассчитать коэффициент сжатия

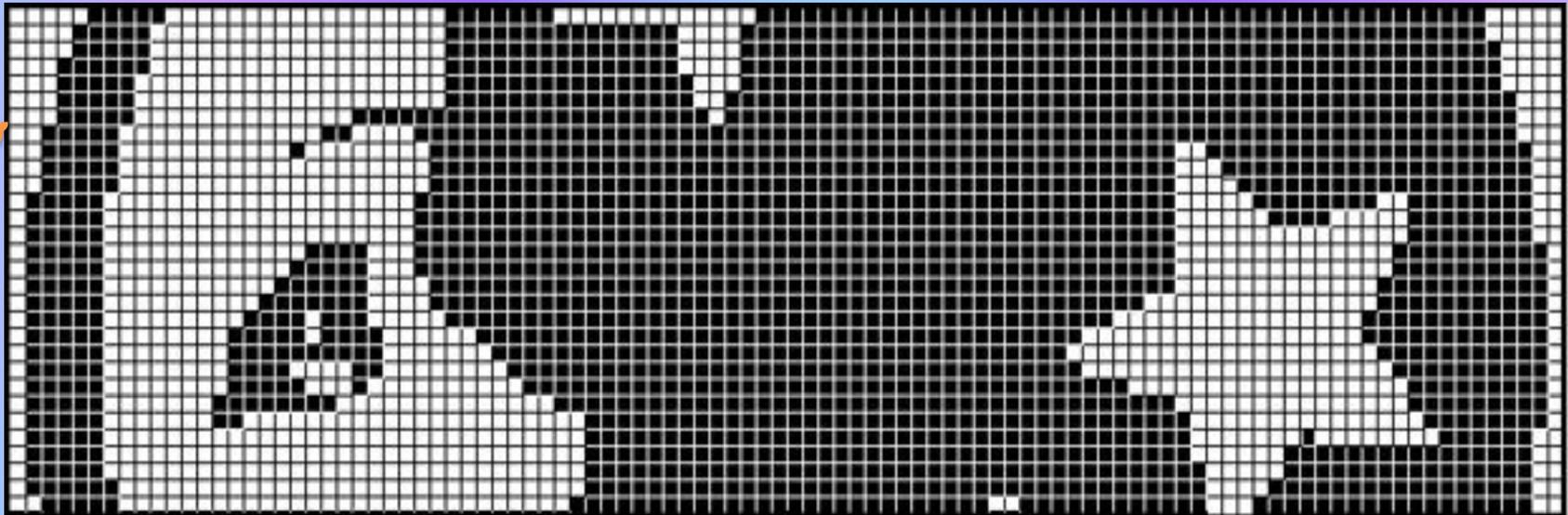


Сжатие путём учёта числа повторений.

- **Алгоритм RLF.** Выявляются группы идущих подряд одинаковых однобайтовых кодов. Группа заменяется на два байта: число повторений (≤ 127), повторяющийся байт

Эффективен для графики с большими областями равномерной закрашки.





RLE – метод сжатия без потерь (Строка 23: 2 нуля, 5 единиц, 25 нулей, 43 единицы и т. д.)

Сжатие путём учёта числа повторений.

- **Алгоритм Лемпеля – Зива (LZ77,LZ78).**
При обнаружении слова, которое уже встречалось, на него формируется ссылка в виде смещения назад относительно текущей позиции и длины слова в байтах.

Дома:
§1.4.5,
стр. 75 №3,4 в тетради