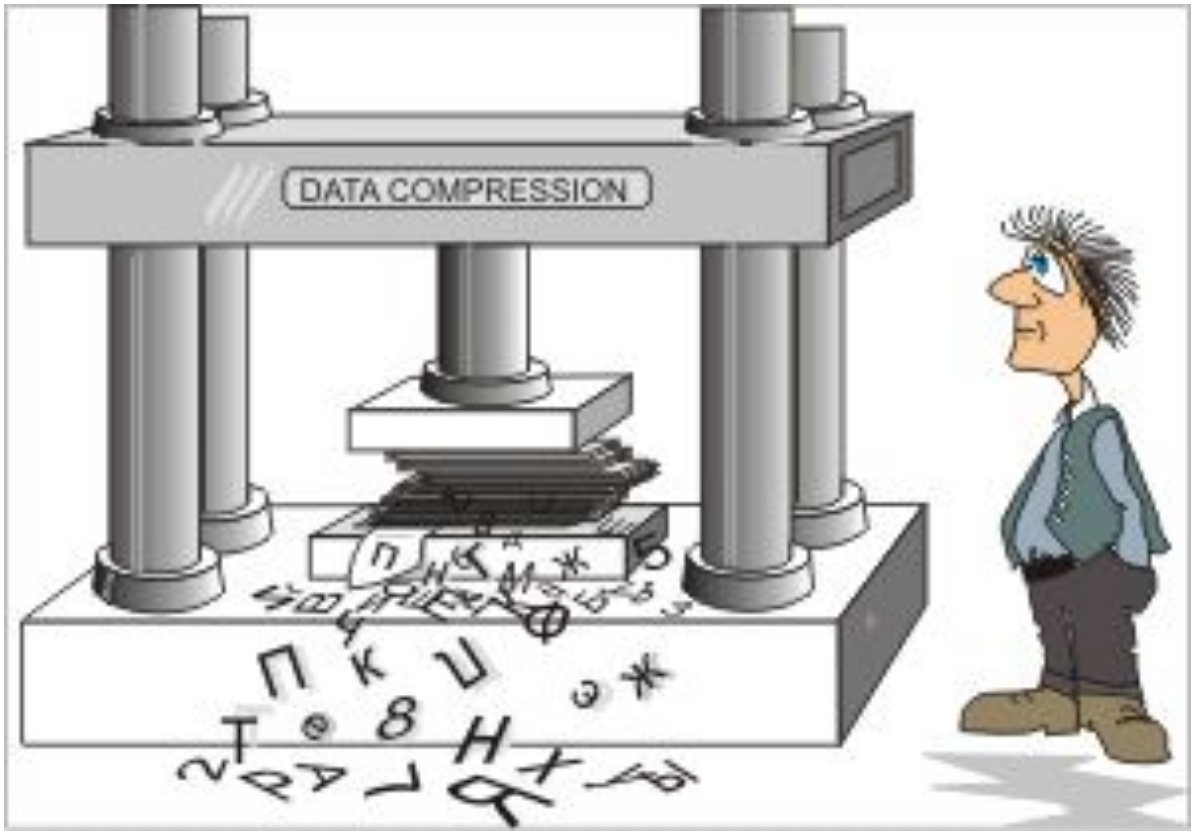
Сжатие данных

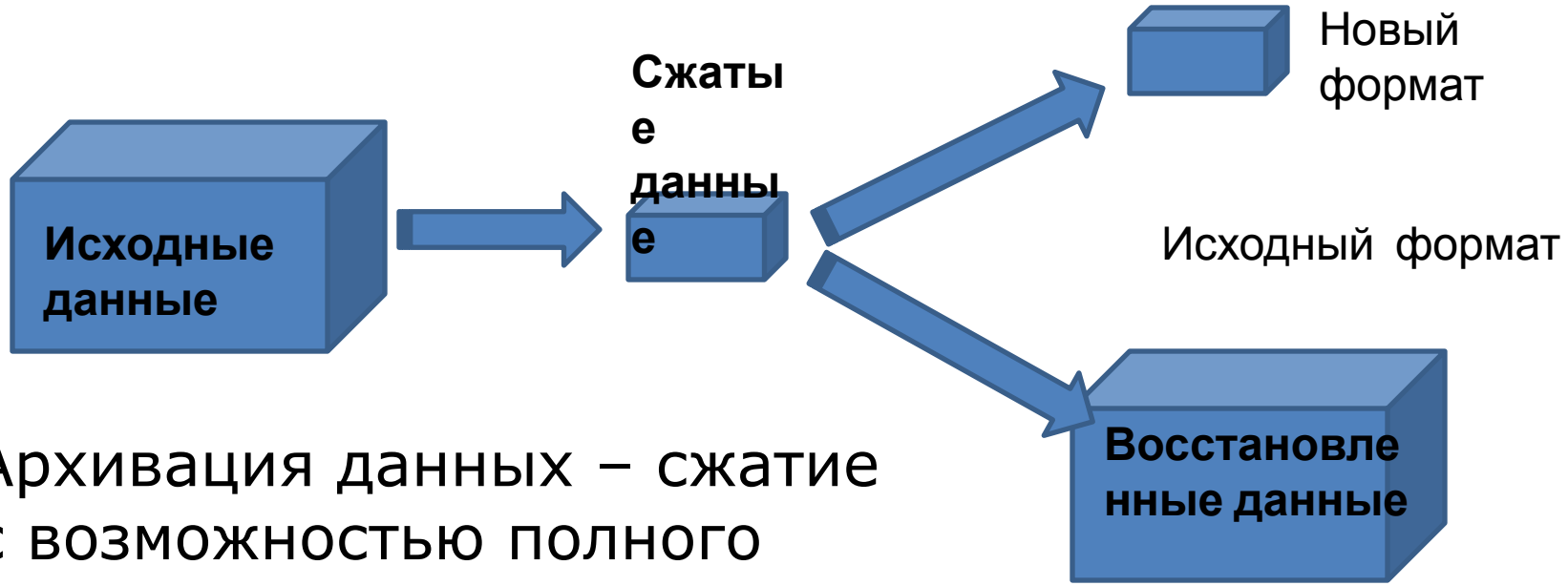


Сжатие данных это процесс, обеспечивающий уменьшение объема данных.

Способы сжатия

- Изменение содержания данных (уменьшение избыточности данных)
- Изменение структуры данных (эффективное кодирование)
- Изменение содержания и структуры данных

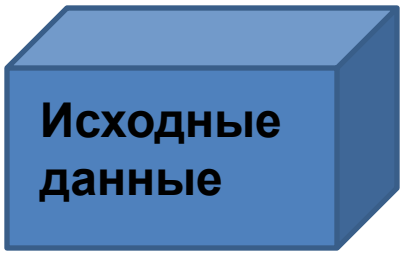
Цели сжатия данных – экономия ресурсов при хранении или передаче данных



Архивация данных – сжатие с возможностью полного восстановления данных

- **Коэффициент сжатия** – это величина для обозначения эффективности метода сжатия, равная отношению количества информации до и после сжатия

$$К_{сж} = 2\ МБ / 0,5\ МБ = 4$$



Размер файла 2МБ

Сжатые данные

Размер файла 512 КБ

Сжатие данных может происходить с потерями и без потерь

- **Сжатие без потерь (полностью обратимое)** – это методы сжатия данных, при которых данные восстанавливаются после их распаковки полностью без внесения изменений (используется для текстов, программ) Ксж до 50%
- **Сжатие с регулируемыми потерями** – это методы сжатия данных, при которых часть данных отбрасывается и не подлежит восстановлению

(используется для видео, звука, изображений)

Ксж до 99%

Сжатие с потерями

Тип данных	Тип файла после сжатия	Степень сжатия
Графика	.JPG	до 99%
Видео	.MPG	
Звук	.MP3	

Сжатие без потерь

Тип данных	Тип файла после сжатия	Степень сжатия
Графика	.GIF .TIF .PCX	До 50%
Видео	.AVI	
Любой тип	.ZIP .ARJ .RAR .LZH	

Алгоритмы сжатия символьных данных

- **Статистические методы** – это методы сжатия, основанные на статистической обработке текста.
- **Словарное сжатие** – это методы сжатия, основанные на построении внутреннего словаря.

Упаковка однородных данных

Закодируем сообщение длиной 16 символов

0,258-23,5+18,01

В кодировке ASCII сообщение составляет 16 байт.

Новая кодовая таблица для упаковки:

0 0000	**1** 0001	**2** 0010	**3** 0011	**4** 0100	**5** 0101	**6** 0110
7 0111	**8** 1000	**9** 1001	**_** 1010	**+** 1011	**-** 1100	**,** 1101

Код сообщения после упаковки составляет 8 байт:

000011010 01010101 00011000 0100011

110101011 01100011 00011010 0000001

$K_{сж} = 16 / 8 = 2$

Достоинства и недостатки метода

+ коэффициент сжатия увеличивается с увеличением размера символьного сообщения;

- необходимо указывать для распаковки новую кодовую таблицу;

- эффективен только для однородных сообщений, использующих ограниченный набор символов исходного алфавита;

Статистический метод сжатия
Алгоритм Хаффмана

Разные символы встречаются в сообщении с разной частотой, например для русского алфавита в среднем на 1000 символов:

символ	пробел	о	а	р	к	я	г	ю	ф
частота	175	90	62	40	28	18	13	6	2

Зададим коды символам согласно частоте их повторения:

чем чаще встречается символ, тем короче его код

(неравномерное кодирование)

Хаффмановское кодирование (сжатие) – это метод сжатия, присваивающий символам алфавита коды переменной длины, основы-ваясь на частоте появления этих символов в сообщении.

символ	код символа
пробел	00
о	01
р	101
к	110
ю	0110
ф	1001

Проблема декодирования

Пример : пусть коды символов **a**-10, **b** -101, **c**-1010

Декодировать сообщение 10101011010

10 10 101 1010 10 10 101 10 10 1010 101 1010

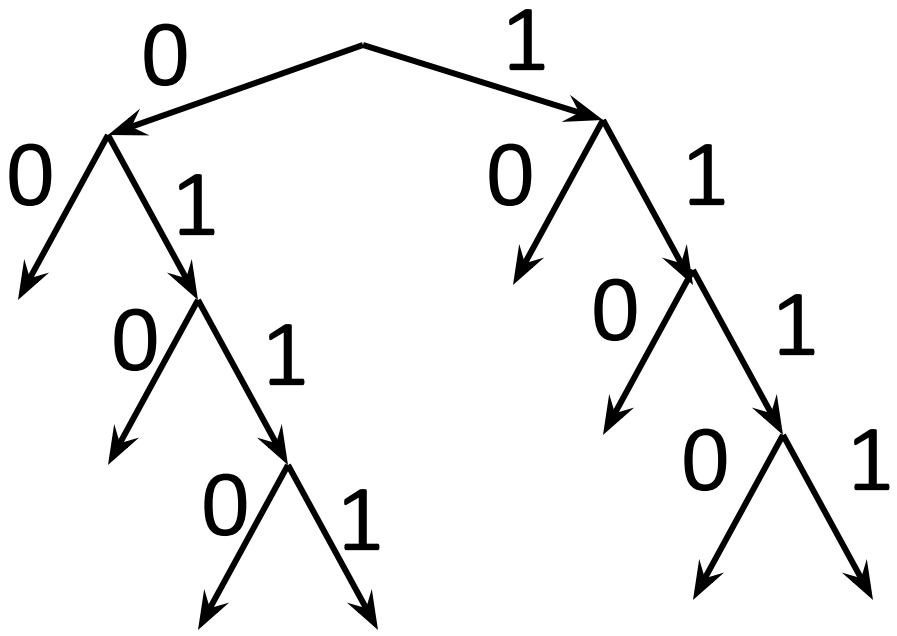
a a b c **a a b a a** **c b c**

Однозначное декодирование возможно при условии Фано: никакое кодовое слово не является началом (префиксом) другого кодового слова.

Префиксный код – это код, в котором никакое кодовое слово не является префиксом любого другого кодового слова.

Пример префиксного кода :

00 10 010 110 0110 0111 1110 1111



Префиксный код задается орграфом с размеченными листьями

Пример: построить код Хаффмана для фразы

ОТ_ТОПОТА_КОПЫТ_ПЫЛЬ_ПО_ПОЛЮ_ЛЕТИ

1. Определим частоту вхождения символов в фразу:

символ	А	Е	И	К	Л	О	П	Т	Ы	Ь	Ю	_
частота	1	1	1	1	3	6	5	6	2	1	1	6

2. Строим орграф Хаффмана:
   -символ задает вершину- лист орграфа;
   -вес вершины равен частоте вхождения символа;
   -соединяются пары вершин с наименьшим весом:
   -левые ветви обозначаем 0;
   -правые ветви обозначаем 1;

Достоинства и недостатки метода

+ Алгоритм Хаффмана универсальный, его можно применять для сжатия данных любых типов;

- Классический алгоритм Хаффмана требует хранения кодового дерева, что увеличивает размер файла.

Метод словарей
Алгоритм сжатия LZ

Этот алгоритм был впервые описан в работах А. Лемпеля и Дж. Зива (Abraham Lempel, Jacob Ziv) в 1977-78 гг., поэтому этот метод часто называется Lempel-Ziv, или сокращенно LZ.

В его основе лежит идея замены наиболее часто встречающихся цепочек символов (строк) в файле ссылками на «образцы» цепочек, хранящиеся в специально создаваемой таблице (словаре).

Алгоритм разработан израильскими математиками Якобом Зивом и Аб рахам ом Лемпелем.

Словарь содержит, кроме многих других, такие цепочки:
1-ра 2-аб 3-ат 4-мат 5-ми_ 6-ам 7-бо 8-ом_ 9-бом 10-ем 11-лем

Алгоритм раз1ботан из1ильскими мате4ика5Яко7ом Зив821х68 Л10пе11

Чем длиннее цепочка,заменяемая ссылкой в словарь, тем больше эффект сжатия.

Достоинства и недостатки метода

+

-применим для любых данных;

- очень высокая скорость сжатия;

- высок коэффициент сжатия;

-

-словарь настроен на тип текста;

- словарь может быть очень большим;

Вопросы по теме:

1. Что такое архивирование данных? Для данных каких типов возможно применять архивирование?
2. Для каких данных допустимо сжатие с потерями?
3. При каких условиях метод упаковки неэффективен?
4. Что такое префиксный код?
5. Для каких данных метод Хаффмана эффективен?
6. На каких принципах основан метод словарного сжатия?
7. Назовите известные вам программы для сжатия данных.
8. Есть ли эффект от архивирования сжатых данных? Почему?
9. Изменилось ли количество информации в звукозаписи после сжатия с потерями? Поясните свой ответ.
10. Изменилось ли количество информации в изображении после его архивирования? Поясните свой ответ.

Домашнее задание: используя любые данные указанного типа, проведите эксперименты по архивированию. Результаты занесите в таблицу и поясните полученный эффект сжатия.

Тип данных	Исходный формат	Исходный размер	Формат архива	Размер архива	Абсолютная величина сжатия(вМБ)	Коэффициент сжатия	Пояснение эффекта сжатия
Текст	.doc						
	.pdf						
Видео	.avi						
	.mpg						
Изображение	.bmp						
	.jpeg						
Звук	.mp3						
	.midi						

$К_{сж} = (\text{исходный размер файла} - \text{размер файла архива}) / \text{исходный размер файла}$