

ОНЛАЙН КУРС:

# УПРАВЛЕНИЕ ДААННЫМИ

ОНЛАЙН КУРС:

# УПРАВЛЕНИЕ ДАННЫМИ



ТЕМА 6.2

Архивирование данных

**Вопрос 1**



# **Методы сжатия информации**

## Современные архиваторы

Принцип работы современных архиваторов основан на поиске в файле «избыточной» информации и последующем ее кодировании с целью получения минимального объема



# Методы сжатия информации

- Сжатие последовательности байтов, которые часто повторяются
- Алгоритм Хаффмана
- алгоритм Лемпела-Зива



## Пример 1



- Файл занимает 15 байт и состоит из следующей последовательности СИМВОЛОВ:
- В В В В В L L L L L А А А А  
А
- В шестнадцатиричной системе:
- 42 42 42 42 42 4C 4C 4C  
4C 4C 41 41 41 41 41

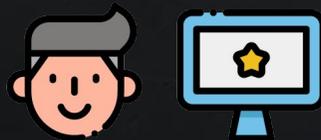
## Пример 2



- Архиватор может представить этот файл в виде (16-тиричном): 01 05 42 06 05 4C 0A 05 41
- Эти последовательности можно интерпретировать следующим образом: с первой позиции 5 раз повторяется знак B, с шестой позиции 5 раз повторяется знак L, с позиции 11 повторяется 5

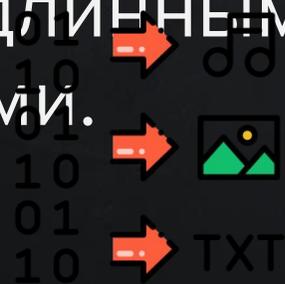
# Алгоритм Хаффмана

- **Оптимальный префиксный код или кодирование символами переменной длины** - более изощренный метод сжатия данных, используемый в том или ином виде практически любым архиватором.



# Алгоритм Хаффмана

- **Код переменной длины** позволяет записывать наиболее часто встречающиеся символы и фразы несколькими битами, в то время как редкие символы и фразы будут записаны более длинными битовыми строками.



## Пример



- Анализируя любой английский текст, можно установить, что буква E встречается гораздо чаще, чем Z, а X и Q относятся к наименее встречающимся.
- Используя специальную таблицу соответствия, можно закодировать каждую букву E меньшим числом бит, используя более длинный код для более редких букв, тогда как в обычных кодировках любому символу соответствует битовая последовательность фиксированной длины (как правило, кратной байту)

# Алгоритм Лемпела-Зива

- Популярные архиваторы работают на основе алгоритма Лемпела-Зива.
- Например, все слова книги могут быть представлены в виде номеров страниц и номеров строк некоторого словаря.



Вопрос 2



# Фрактальные методы архивации

# Фрактал

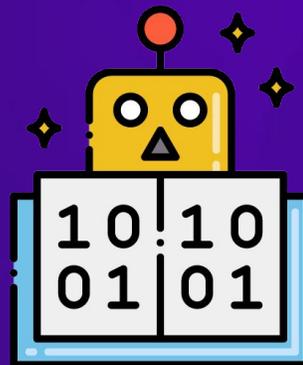


Понятия «**фрактал**» и «**фрактальная геометрия**» (fractus - состоящий из фрагментов, лат.) были предложены математиком Б. Мандельбротом в 1975 г. для обозначения нерегулярных, но самоподобных структур



# Фрактал

Это структура, состоящая из частей, которые в каком-то смысле подобны целому



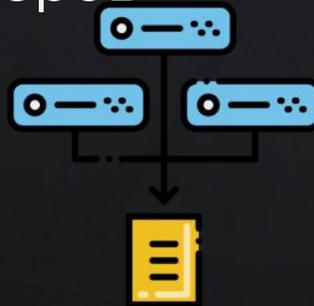
# Метод IFS

Применяется к построению фрактальных изображений, изобретённый большим их знатоком Майклом Барнсли и его коллегами из Технологического института шт. Джорджия, базируется на самоподобии элементов изображения и заключается в моделировании рисунка несколькими меньшими фрагментами его самого



# Главное преимущество IFS

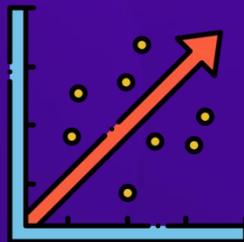
IFS-фракталы имеют одно вполне реальное и полезное применение: с их помощью можно сжимать большие растровые изображения до долей их нормальных размеров



# Алгоритм Лемпела-Зива



- Фрактальные методы сжатия позволяют сжать информацию в 10 000 раз.
- Все известные программы фрактальной компрессии базируются на алгоритме Джеквина



## Фрактальное сжатие



- Изображение  $R$  разбивают на кусочки  $r_i$ , называемые ранговыми областями. Далее для каждой области  $r_i$  находят область  $d_i$  и преобразование  $w_i$  такие, что выполняются следующие условия



# Фрактальное сжатие



- $d_i$  по размерам больше  $r_i$
- $w_i$  ( $r_i$ ) имеет ту же форму, размеры и положение, что и  $r_i$
- Коэффициент и преобразования  $w_i$  должен быть меньше единицы
- Значение должно быть как можно меньше

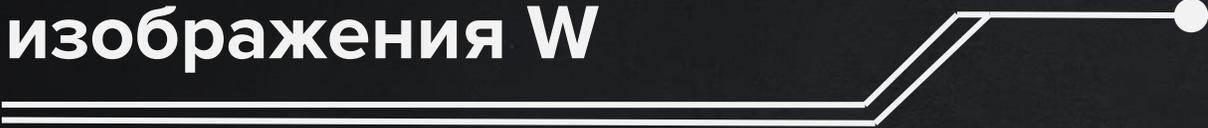


# Фрактальное сжатие

- Первые три условия означают, что отображение  $w_i$  будет сжимающим. А в силу четвёртого условия кодируемое изображение  $R$  и его образ  $W(R)$  будут похожи друг на друга. В идеале  $R = W(R)$ . А это означает, что наше изображение  $R$  и будет являться неподвижной  $W$



# Компрессия изображения $W$



- Разбить изображение на ранговые области  $r_i$
- Для каждой ранговой области  $r_i$  найти область  $d_i$ , и отображение  $w_i$ , с указанными выше свойствами
- Запомнить коэффициенты аффинных преобразований  $W$ , положения доменных областей  $d_i$ , а также разбиение изображения на домены

# Декомпрессия изображения

- Создать какое-то (любое) начальное изображение  $R_0$
- Многократно применить к нему отображение  $W$  (объединение  $w_i$ )
- Так как отображение  $W$  сжимающее, то в результате, после достаточного количества итераций, изображение придёт к аттрактору и перестанет меняться

# Декомпрессия изображения

- Именно это и позволяет при развертывании увеличивать его в несколько раз. Особенно впечатляют примеры, в которых при увеличении изображений природных объектов проявляются новые детали, действительно этим объектам присущие



# Выводы



- К сожалению, даже на современном ПК понадобится недопустимо много времени для того, чтобы сжать изображение размером всего 256 x 256 пикселей. Очевидно, что текущий алгоритм нуждается в оптимизации

**Благодарю  
за внимание!**