**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**«Российский университет дружбы народов имени Патриса Лумумбы»**

Инженерная академия

*Кафедра механики и процессов управления*

**Курсовая работа**

По информатике и программированию

**Направление:** Прикладная математика и информатика

**Профиль:** Математические методы механики космического полёта и анализ геоинформационных данных

**Тема: Сжатие без потерь: от Хаффмана до LZW**

Выполнено студентами: Фирсов Егор Дмитриевич

Иркитова Элина Эркеновна

Группа: ИПМбд-01-23

№ студенческого: 1132233511

№ студенческого:1132231889

**Москва, 2025**

**Цель работы:** изучить оба метода кодирования и сделать выводы о их эффективности

**Хаффман**

Кодирование Хаффмана – это алгоритм сжатия данных, основанный на статистической частоте символов в сообщении. Он использует коды переменной длины, где часто встречающиеся символы кодируются короткими последовательностями, а менее часто встречающиеся – длинными. Это позволяет уменьшить общий объем данных, представленных в битах.

**Основные принципы кодирования Хаффмана:**

Статистический анализ:

Алгоритм анализирует частоту появления каждого символа в исходном сообщении.

Формирование дерева Хаффмана:

На основе частот строится бинарное дерево, где часто встречающиеся символы располагаются ближе к корню, а менее часто встречающиеся – дальше.

Префиксное кодирование:

Каждому символу присваивается уникальный префиксный код, то есть ни один код не является префиксом другого кода.

Сжатие:

После построения дерева и присвоения кодов, исходное сообщение кодируется, используя эти коды.

**Преимущества кодирования Хаффмана:**

- Эффективное сжатие:

Уменьшает объем данных, особенно для текстов и изображений, где некоторые символы встречаются гораздо чаще других.

- Оптимальность:

Алгоритм Хаффмана является одним из самых эффективных префиксных кодов для сжатия данных.

- Простота реализации:

Алгоритм относительно простой для реализации и быстрого кодирования и декодирования данных.

**Применение кодирования Хаффмана:**

JPEG и MPEG-2:

Использование кодирования Хаффмана в этих форматах сжатия изображений и видео.

Сжатие архивов:

Часто используется в архиваторах данных, таких как ZIP, чтобы уменьшить размер файлов.

Трансляция данных:

Применяется для эффективной передачи информации по сетям.

**LZW**

**LZW (Lempel-Ziv-Welch)** - это алгоритм сжатия данных без потерь, который используется для сжатия различных типов данных, в том числе изображений, текстов и программного кода. Он основан на концепции замены повторяющихся последовательностей символов кодами, что позволяет уменьшить объём данных.

**Основные характеристики LZW:**

**- Сжатие без потерь:**

После сжатия и последующего восстановления данных, они будут идентичны исходным, без потери информации.

**- Использование словаря:**

LZW использует динамически расширяемый словарь, который содержит информацию о найденных повторяющихся последовательностях символов.

**- Замена строк символов кодами:**

Алгоритм заменяет повторяющиеся строки символов их уникальными кодами, что позволяет сократить объём данных.

**- Применение в различных форматах:**

LZW используется в графических форматах, таких как GIF и TIFF, а также в других областях, где требуется сжатие данных без потерь.

**Принцип работы LZW:**

Инициализация словаря:

Алгоритм начинает с начального словаря, содержащего все отдельные символы, которые могут встречаться в данных.

Поиск повторяющихся строк:

Алгоритм ищет последовательности символов, которые уже существуют в словаре.

Замена строк кодами:

Если последовательность найдена, она заменяется своим кодом.

Расширение словаря:

Если последовательность не найдена, она добавляется в словарь с новым кодом.

Повторение шагов 2-4:

Алгоритм продолжает процесс обработки данных до тех пор, пока не будут обработаны все символы.

Декомпрессия:

Для восстановления данных после сжатия необходимо знать начальный словарь и код каждой замененной последовательности.

**Преимущества LZW:**

- Высокая эффективность:

LZW может значительно уменьшить объём данных, особенно в случаях, когда в данных есть много повторяющихся последовательностей.

- Поддержка различных типов данных:

LZW подходит для сжатия различных типов данных, включая изображения, тексты и программы.

- Простота реализации:

Алгоритм LZW относительно прост в реализации.

**Недостатки LZW:**

1. Возможный рост объёма данных:

В некоторых случаях LZW может не дать желаемого эффекта сжатия, а даже увеличить объём данных.

1. Чувствительность к начальным данным:

Эффективность LZW может зависеть от структуры исходных данных.

В целом, LZW - это эффективный и широко используемый алгоритм сжатия без потерь, который подходит для различных задач, требующих сжатия данных без потери качества.

**Чем они отличаются?**

Кодирование Хаффмана и LZW – это два различных метода сжатия данных без потерь, используемые для уменьшения размера файлов. Основное отличие заключается в том, что Хаффман использует статические частотные таблицы, а LZW – адаптивные.

Кодирование Хаффмана:

- Статическое: основано на предварительно установленных частотах символов.

- Простота: легко реализовать и декодировать.

- Не адаптируется: не учитывает частоты, меняющиеся в процессе сжатия.

LZW (Lempel-Ziv-Welch):

- Адаптивное: динамически создает таблицу кодов, основанную на частотах встречающихся последовательностей символов в данных.

- Более эффективное: в некоторых случаях может обеспечить более высокую степень сжатия по сравнению с Хаффманом, особенно для текстов с повторяющимися фрагментами.

- Более сложное: имеет более сложную реализацию и декодирование по сравнению с Хаффманом.

**Выводы:**

**Хаффман -** когда данные имеют относительно стабильную частотную структуру и не требуется высочайшая степень сжатия.

**LZW -** когда данные имеют большие повторяющиеся фрагменты и требуется более высокая степень сжатия, например, при сжатии текстовых файлов или изображений в формате GIF.