Выводы по Лабораторной Работе №2 “Методы упаковки данных”

* В ходе выполнения лабораторной работы были реализованы два алгоритма сжатия данных: алгоритм LZW и алгоритм Хаффмана;
* Для алгоритма LZW была выбрана длина кода равная размеру типа данных ushort (2 байта), этот выбор сделан был вручную перебором различных типов данных, как размер, при котором книга «Евгений Онегин» в формате fb2 сжималась с максимальной степенью сжатия. (Несжатый файл 416 Кб, сжатый 180 Кб, коэффициент сжатия = 2,31);
* Исходя из вышесказанного следует, что эффективность алгоритма LZW на данном файле зависит от выбранного размера кода, эффективность будет максимальной для всех размеров файлов у реализации алгоритма, которая будет динамически подбирать подходящую длину кода;
* Для алгоритма Хаффмана так же была предложена своя реализация BitReader/BitWriter с буффером, т. к. .NET, на котором были выполнены лабораторные работы, поддерживает минимальную порцию данных для записи равную 1 байту, но код Хаффмана является неравномерным и может быть не равен какому-либо типу данных, поэтому было принято решения накапливать биты кода в буфер и при достижении размера в 8 бит, конвертировать их в байт и записывать на диск.
* Алгоритм Хаффмана так же проверялся при сжатии книги «Евгений Онегин» в формате fb2, коэффициент сжатия = 1,83 (Несжатый файл 416 Кб, сжатый 224 Кб, размер словаря кодов Хаффмана 3Кб);

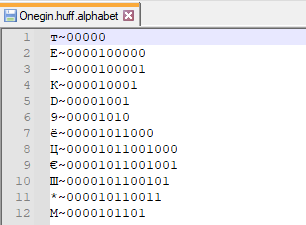


Рисунок . Часть файла словаря с кодами Хаффмана (~ разделитель символа и кода)

* К сожалению, целиком словарь слишком велик, чтобы вставить его в текущий документ, но он будет приложен к решению отдельным файлом. В словарь коды записывались символами, а не битами, чтобы можно было наглядно его посмотреть и отладить без использования hex-редактора, при бинарной записи можно словарь так же незначительно сократить;
* На файле с книгой эффективность сжатия по алгоритму LZW оказалась выше, но если взять файл большего размера, допустим, в несколько мегабайт, то эффективность алгоритма Хаффмана будет выше, так как не требуется вручную подбирать подходящую длину кода, что было сделано для алгоритма LZW на данном файле, поэтому можно назвать алгоритм Хаффмана более универсальным, хоть он и требует большее количество ресурсов, т.к. является двухпроходным;
* Так же одним из недостатков алгоритма LZW можно выделить необходимость первоначальной инициализации внутреннего словаря всеми возможными одно-символьными фразами, если текст будет содержать большое количество спец. символов, которыми располагает кодировка Юникод, то словарь в начале работы сжатия будет огромен и потребуется больший тип данных, чем при хранении только однобуквенных фраз (в предложенной реализации словарь инициализируется только первыми 256 символами кодовой страницы 1251, как содержащей русские и латинские буквы, чтобы начальные фразы не занимали большую часть возможных значений типа ushort).