FINAL PROJECT REPORT v1.0

A Statistical analysis and extended implementation of

Compression algorithms

Guided By: Prof. Bhaskar Chaudhary

Prepared By: Rachit Mishra (201201092)

Dhrumil Shah (201201117)

Abstract

This project report presents a detailed analysis of some compression algorithms and their implementation with some new functionalities developed for them in an attempt to increase their efficiency. While conducting the research for this project, a systematic procedure of testing was done on these compression algorithms and the details on a comparative analysis of results and properties exhibited by different algorithms under different parameters has been tabulated in this report in both statistical and graphical forms.

INTRODUCTION

- Data Compression as the name suggests is a technique in by which the same amount of **data** is transmitted by using a smaller number of bits; for example, by replacing a string of twenty repeated digits with a command to repeat the digit twenty times.

- Data Compression can be either lossy or lossless. As the names suggest, in lossless data compression the information is not lost and all of the information at the beginning is as it is at the end. It reduces bits by identifying and eliminating statistical redundancy.

- Data Compression shrinks down a file so that it takes up less space. This is desirable for data storage and data communication.

- Storage space on disks is expensive so a file which occupies less disk space is "cheaper" than an uncompressed file. Smaller files are also desirable for data communication, because the smaller a file the faster it can be transferred.

- A compressed file appears to increase the speed of data transfer over an uncompressed file.

- Data Compression shrinks down a file so that it takes up less space. This is desirable for data storage and data communication.

- Storage space on disks is expensive so a file which occupies less disk space is "cheaper" than an uncompressed file. Smaller files are also desirable for data communication, because the smaller a file the faster it can be transferred.

- A compressed file appears to increase the speed of data transfer over an uncompressed file.

ALGORITHMS COVERED

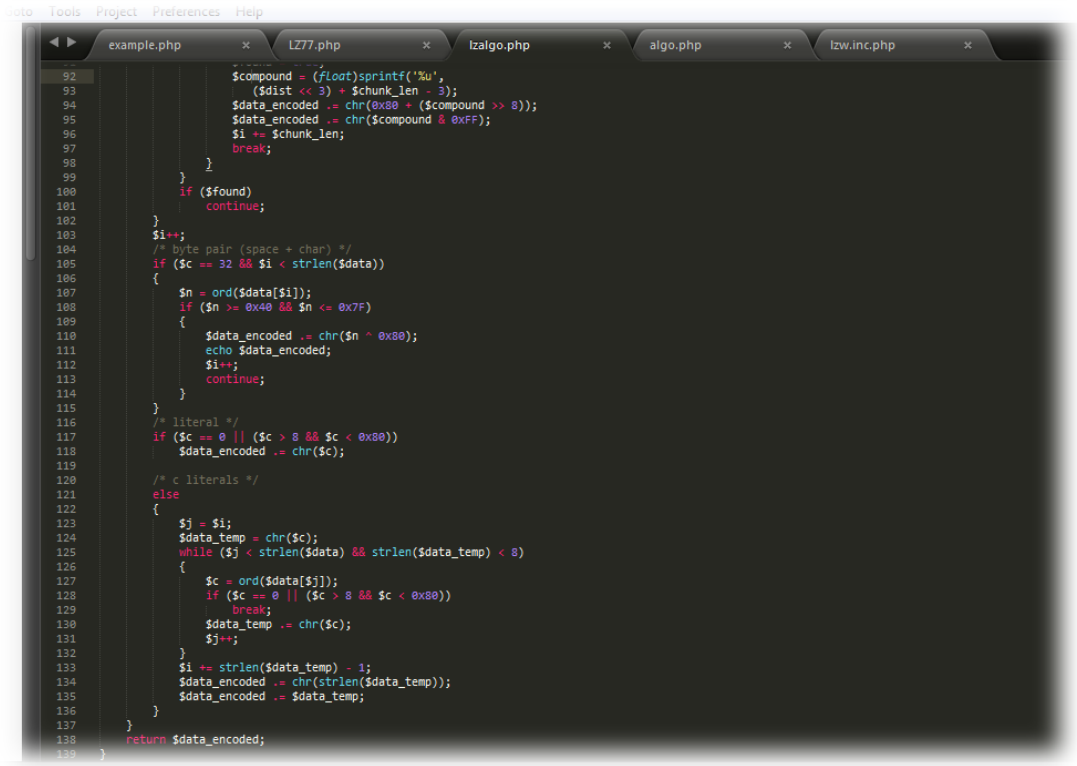
The algorithms we worked upon and which we have covered in these report are:

1. Huffman encoding

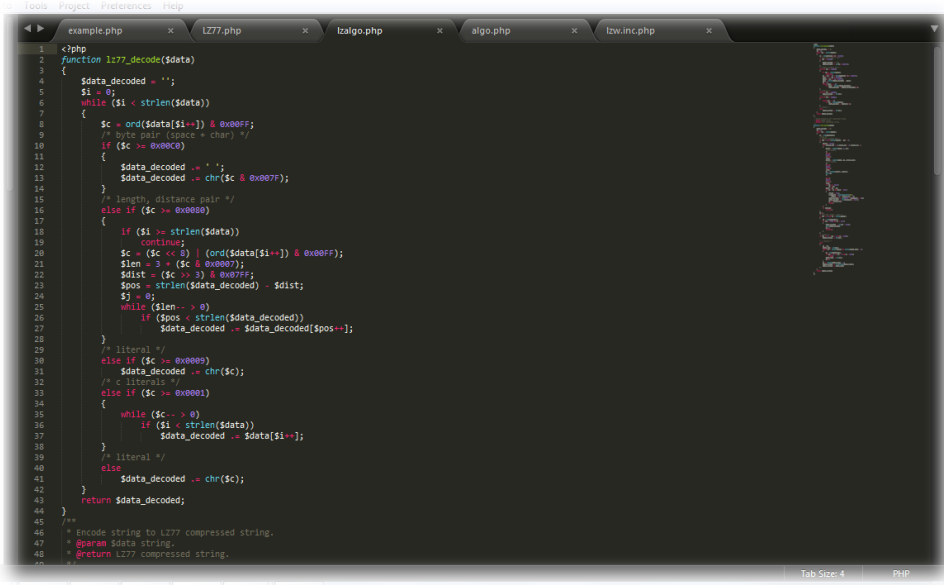
2. LZW algorithm (Precisely, we implemented LZ 77 which is an extended variation of the LZW algorithm.

- LZ77 and LZ78 are lossless data compression algorithms : Forms basis of **DEFLATE** used in PNG files) **and GIF**
- Lossless Data Compression algorithms: Original data is completely reconstructed from the compressed data. Lossy data compression allows compression of only an approximation of the original data.(although improves compression rates)
- LDC algorithms used in ZIP file formats. PNG and GIF use lossless compressions. 3D graphics (OpenCTM), Cryptography.
- Both LZ77 and LZ78 are **DICTIONARY CODERS.(Substitution coders) :** A class of LDC algorithms.
- Look for matches between data set to be compressed and a set of strings (contained in a data structure maintained by the encoder) called dictionary.
- When a match is found, a reference to the string's position in the data structure is given.
- In LZ77 compression MUST start at the beginning of the input itself.
- Typically every character is stored with 8 bits (up to 256 unique symbols for this data)..... Can be extended to 9 to 12 bits per character. New unique symbols made of combinations of symbols which occurred previously in the string.

Below 2 screenshots depicts the code for compressing strings via LZ77 algorithm.



The one screenshot shown below describes the method for decompressing strings via LZ77 algorithm.



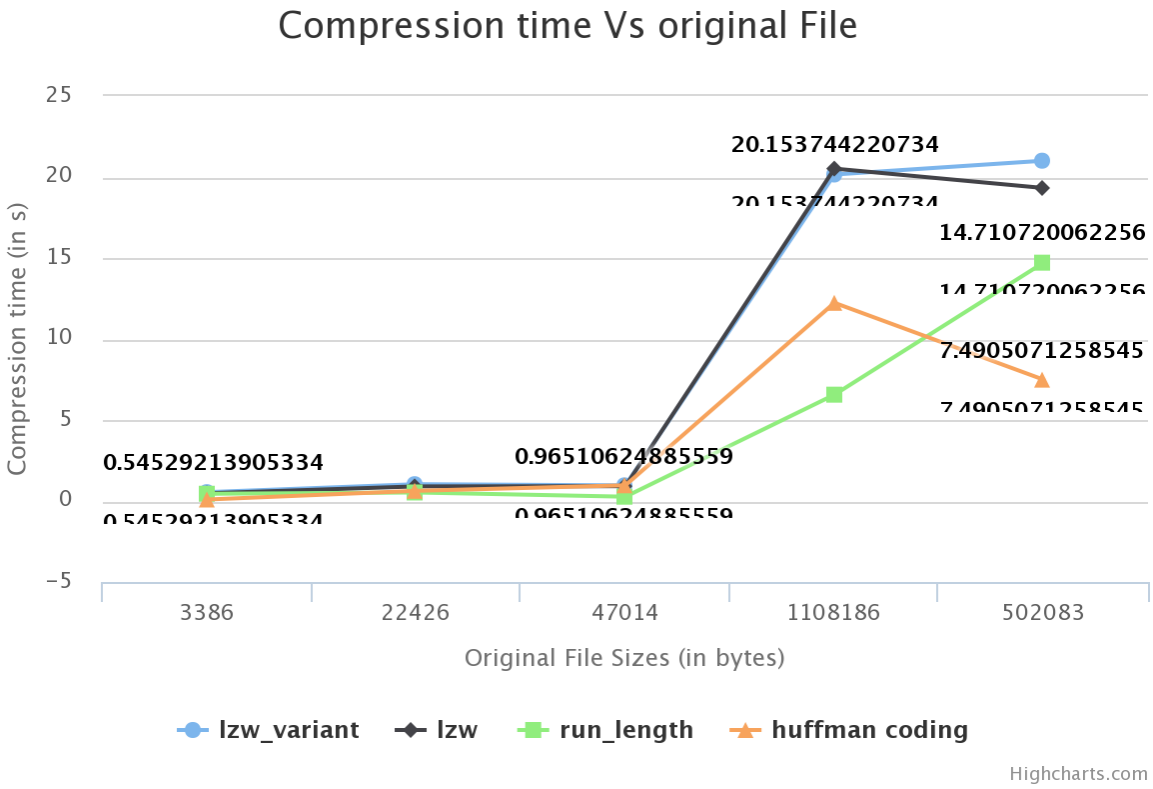
3. Run Length Encoding algorithm

4. Arithmetic Encoding algorithm

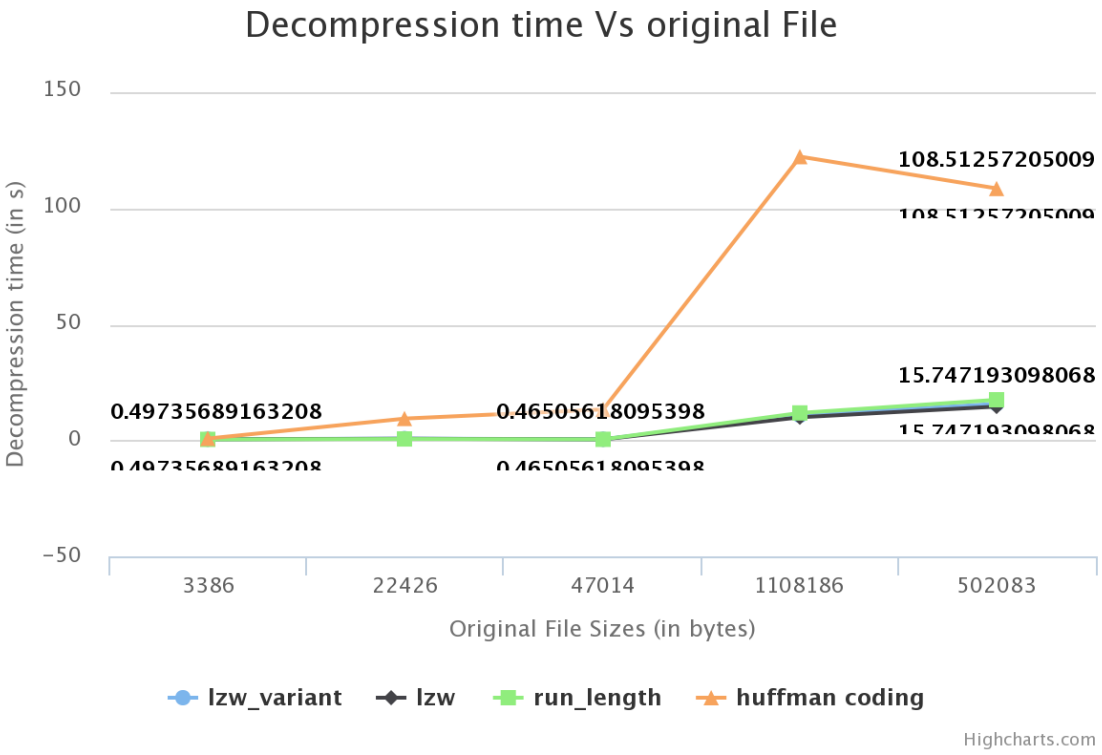
GRAPHICAL ANALYSIS

Based on few parameters, and after implementing all the algorithms mentioned above, we generated graphs highlighting the significant comparisons between these algorithms based on some important factors like compression rates, time taken to compress the files and some other factors.

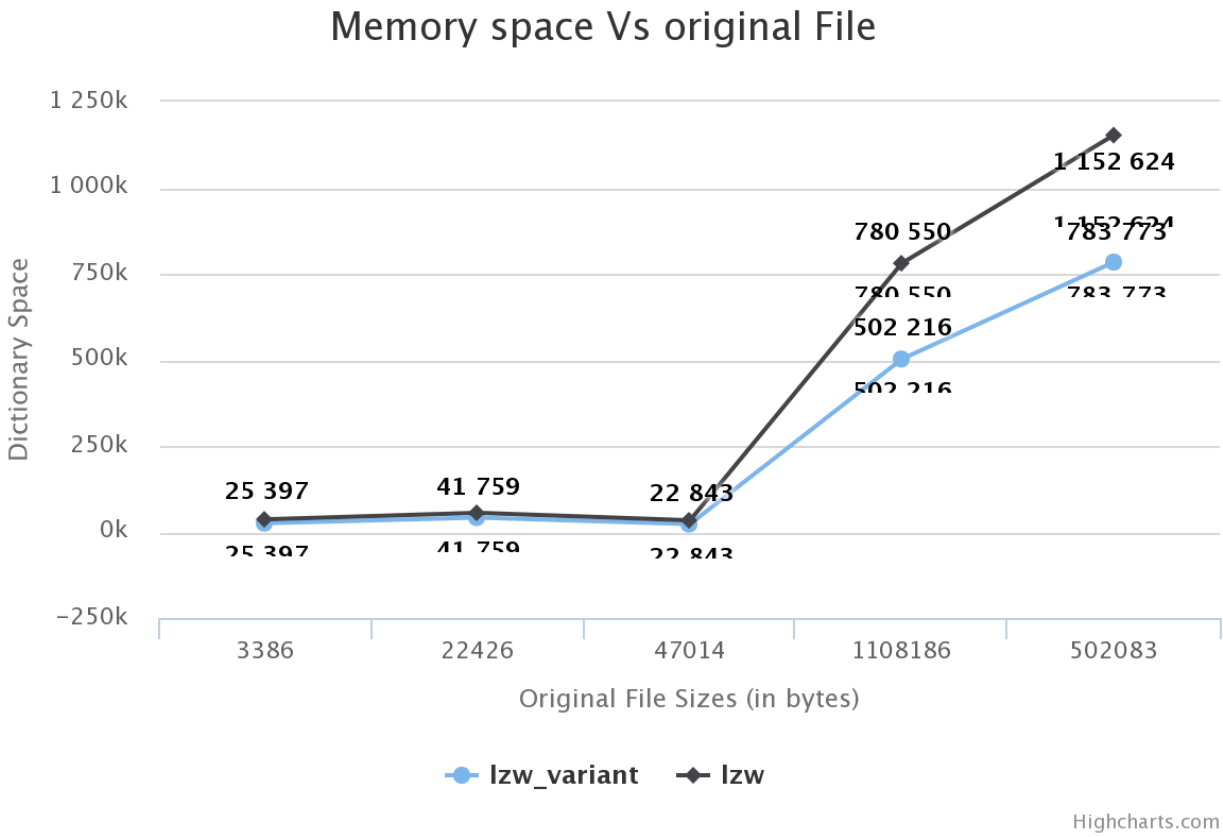
1. Factor describing the original file size versus the size of the compressed file in case of all four algorithms.



2. Factor describing the decompression time versus original file



3. Dictionary space utilized for to compress the original file with some file size.



4. Compressed files vs Original file size comparison for different algorithms.