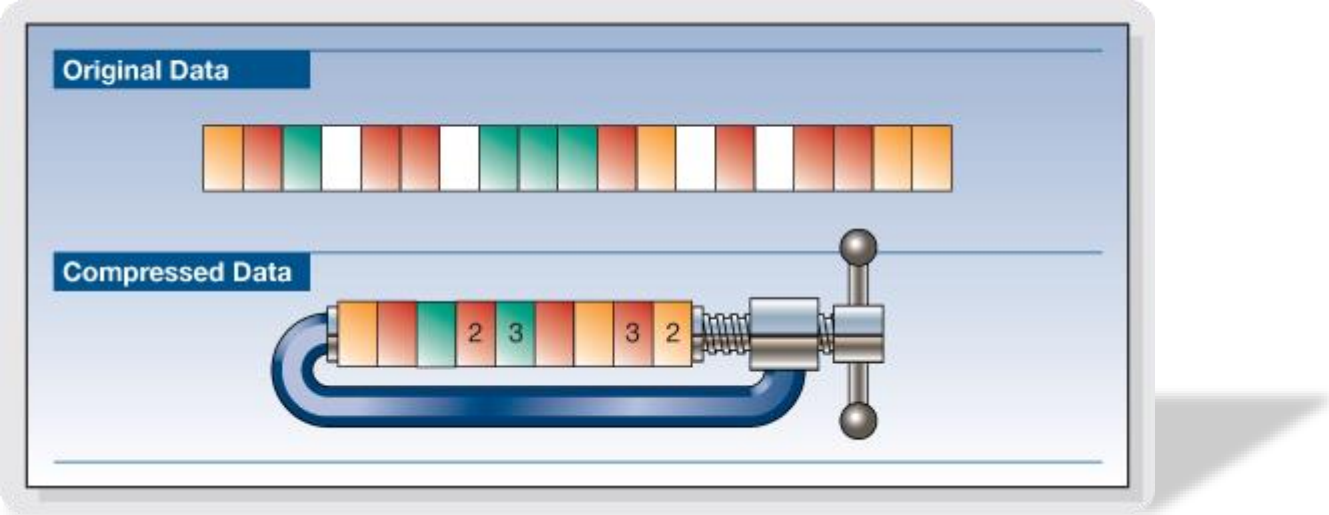
LOSSLESS DATA COMPRESSION ALGORITHM

- *Implement and analyze the Lossless Data Compression Algorithm*



- The project aims at the implementing the existing algorithm and analyze the complexity of the same. Also, we intend to improve the existing algorithms.

Sparsh Gupta || Tanmay Singh

170001049 || 170001051

Introduction

Data compression is a process by which a file (Text, Audio, and Video) can be compressed, such that the original file may be fully recovered without any loss of actual information. This process may be useful if one wants to save the storage space. The exchanging of compressed file over internet is very easy as they can be uploaded or downloaded much faster. Data compression has important application in the area of file storage and distributed system.

In short, Data Compression is the process of encoding data to fewer bits than the original representation so that it takes less storage space and less transmission time while communicating over a network. Data Compression is possible because most of the real-world data is very redundant. A compression program is used to convert data from an easy-to-use format to one optimized for compactness. Likewise, an uncompressing program returns the information to its original form. Various lossless data compression algorithms have been proposed and used. Some of main techniques are Shannon-Fano, Huffman Coding, Run Length Encoding, and Arithmetic Encoding.

Types of Data Compression:

1) Lossy:

2) Lossless:

Huffman Coding:

A binary code tree is generated in Huffman Coding for given input. A code length is constructed for every symbol of input data based on the probability of occurrence. Huffman codes are part of several data formats as ZIP, GZIP and JPEG. Normally the coding is preceded by procedures adapted to the particular contents. For example, the wide-spread DEFLATE algorithm as used in GZIP or ZIP previously processes the dictionary based LZ77 compression. It is a sophisticated and efficient lossless data compression technique. The symbol with highest probability has the shortest binary code and the symbol with lowest probability has the longest binary code.

Shannon Fano Coding:

This is one of an earliest technique for data compression that was invented by Claude Shannon and Robert Fano in 1949.In this technique, a binary tree is generated that represent the probabilities of each symbol occurring. The symbols are ordered in a way such that the most frequent symbols appear at the top of the tree and the least likely symbols appear at the bottom.

Run Length Encoding:

Data often contains sequences of identical bytes. Replacing these repeated byte sequences with the number of occurrences, a reduction of data can be achieved. RLE basically compresses the data by reducing the physical size of a repeating string of characters.

Arithmetic Coding:

Arithmetic encoding is the most powerful compression techniques. This converts the entire input data into a single floating-point number. A floating-point number is similar to a number with a decimal point, like 4.5 instead of 41/2. However, in arithmetic coding we are not dealing with decimal number so we call it a floating point instead of decimal point.

Project Objective

- To analyze the existing data compression algorithms.
- To implement these algorithms.
- To improve the available algorithm for better efficiency and lesser time complexity (if possible).

References

- https://ieeexplore.ieee.org/document/6824486
- https://www.ripublication.com/irph/ijict_spl/07_ijictv3n3spl.pdf
- http://www.ijirst.org/articles/IJIRSTV4I1078.pdf