LZW (Lempel-Ziv-Welch) Compression

- TEAM O

INTRODUCTION

LZW compression is the compression of a file into a smaller file using a table-based lookup algorithm invented by Abraham Lempel, Jacob Ziv, and Terry Welch. Two commonly used file formats in which LZV compression is used are the GIF image format served from Web sites and the TIFF image format. LZW compression is also suitable for compressing text files. No data is lost when compressing. The algorithm is simple to implement and has the potential for very high throughput in hardware implementations. It is the algorithm of the widely used Unix file compression utility compress and is used in the GIF image format.

CHARACTERISTIC FEATURES OF LZW INCLUDES

I. *LZW compression uses a code table, with 4096 as a common choice for the number of table entries. Codes 0-255 in the code table are always assigned to represent single bytes from the input file.*
II. *When encoding begins the code table contains only the first 256 entries, with the remainder of the table being blanks. Compression is achieved by using codes 256 through 4095 to represent sequences of bytes.*
III. *As the encoding continues, LZW identifies repeated sequences in the data, and adds them to the code table.*
IV. *Decoding is achieved by taking each code from the compressed file and translating it through the code table to find what character or characters it represents.*

COMPRESSION

The idea of the compression algorithm is the following: as the input data is being processed, a dictionary keeps a correspondence between the longest encountered words and a list of code values. The words are replaced by their corresponding codes *and so the input file is compressed. Therefore, the efficiency of the algorithm increases as the number of long, repetitive words in the input data increases.*

```
PSEUDOCODE
Initialize table with single character strings
P = first input character
WHILE not end of input stream
     C = next input character
     IF P + C is in the string table
       P = P + C
     ELSE
       output the code for P
     add P + C to the string table
       P = C
     END WHILE
output code for P
```

DECOMPRESSION

The LZW decompressor creates the same string table during decompression. It starts with the first 256 table entries initialized to single characters. The string table is updated for each character in the input stream, except the first one.Decoding achieved by reading codes and translating them through the code table being built.

```
PSEUDOCODE
Initialize table with single character strings
OLD = first input code
output translation of OLD
WHILE not end of input stream
    NEW = next input code
    IF NEW is not in the string table
           S = translation of OLD
           S = S + C
   ELSE
           S = translation of NEW
    output S
    C = first character of S
    OLD + C to the string table
    OLD = NEW
END WHILE
```

EXTENT OF COMPRESSION

Compression ratio of this algorithm completely depends on the extent of repeating string in the file.

More the repeating strings more is the compression ratio. Compression ratio may come less than 1 if there are nearly no repeating strings.

SUMMARY

The other team decreased the file size more by using the upper decompression method as in this we do not need to add dictionary in the file explicitly as we can make dictionary once again while decoding the file and adding every newly received word as codes from number 257 to the dictionary size defined initially.