

## mpel-Ziv coding

- uses strings of chars for compression.
- table shared b/w sender & receiver.
  - contains all possible char strings that occur in text to be transferred.
  - rather than sending ASCII codewords for the string, its index of location in the table is sent.
  - receiver uses this info to recover the text.
  - used as dictionary.
  - dictionary based compression.
- dictionary for spell check & compression.

$$\begin{array}{r} 347 \\ 15 \\ \hline 235 \\ 47 \\ \hline 705 \end{array}$$

eg: "multimedia"  $\rightarrow$  10 chars in word.  
using ASCII, no. of bits = 70 bits.  
using this approach, typically 25000 words,  
 $\therefore$  15 bits sufficient ( $2^{15} = 32768$ ).  
 $\therefore$  Compression ratio  $\div 70 : 15 = 4.7 : 1$

- shorter words have lower compression ratio
- & longer words have higher compression ratio.

Q. LZ algo is to be used to compress text prior to its transmission. If avg no. of chars per word is 6, & dictionary has 4096 words, derive avg. compression ratio achieved relative to using 7-bit ASCII codewords.

→ 4096 words →  $2^{12}$  ∴ 12 bits.

ASCII →  $7 \times 6 = 42$  bits

Ratio :-  $42 : 12 = 3.5 : 1$

X

### Lempel-Ziv-Welsh coding

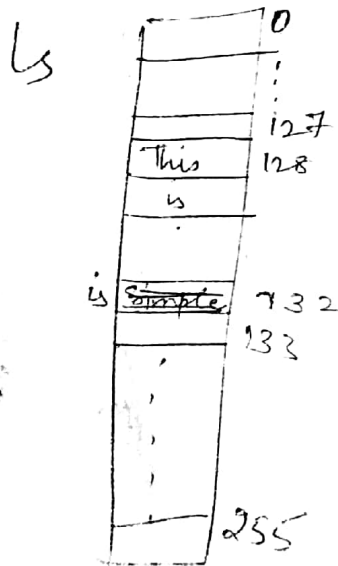
↳ build contents of dictionary dynamically as the text is transferred.

↳ initially contains only ASCII char set.

↳ remaining data is added as new text is encountered.

↳ example: char set has 128 chars & dictionary is limited to ~~4096~~ entries, then first 128 chars contain single chars, & remaining 3968 entries would contain strings of 2 or more chars that make up the words in text being transferred.

eg: "This is simple as it is ..."



As a char is encountered, 't', 'h', 'i', 's', next char is read as 'space' meaning a word just terminated, ∴ 'this' is stored as string in a fresh location & index 128.

↳ can be dynamically used.