**LZW Compression**

A popular dictionary based method

* Basis of compress in Unix also used in gif and tiff formats.
* Due to Lempel, Ziv and Welch hence LZW.
* Patented by Unisys(already expired)

Dictionary a collection of strings

* Codeword represents each string
* Codeword is a bit pattern
* Can be interpreted as a non-negative integer

Whenever a codeword is output during compression, what is written to the compressed file is the bit pattern

* Using a number of bits determined by current codeword length

Dictionary built dynamically during compression and also during decompression

Initially dictionary contains all possible strings of length 1

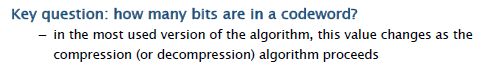
Throughout the dictionary is closed under prefixes

* If string s is represented in dictionary so is every prefix of s

Follows that a trie is an ideal representation of dictionary

* Every node in trie represents a word in dictionary
* Trie is effective and efficient for other reasons too.

Key question how many bits are in a codeword?



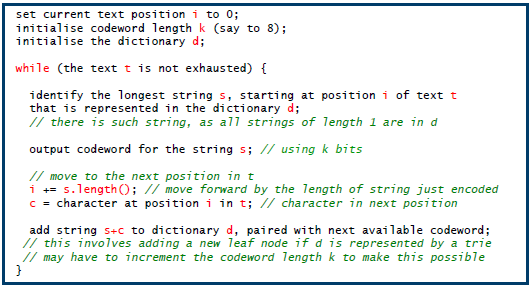
At any given time during compression or decompression

* Current codeword length K
* Exactly 2^k distinct codewords available

All possible bit strings of length K

* Limit size of dictionary
* Codeword length however can be incremented when necessary
* Thereby doubling number of available codewords
* Initial value of K should be large enough to encode all strings of length 1

Pseudo code



**Variants of LZW compression**

Constant codeword length: fix the codeword length for all time

* Fixed capacity, full just stop adding to it.

Dynamic codeword length

* Start with shortest reasonable codeword length, 8 for normal text
* Whenever dictionary becomes full

Add 1 to current codeword length(double number of codewords)

Does not affect sequence of codewords already output

* May specify max codeword length as increasing size may become counter productive

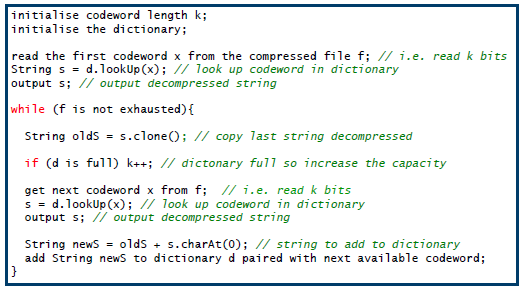
LRU version when dictionary is full and codeword length max

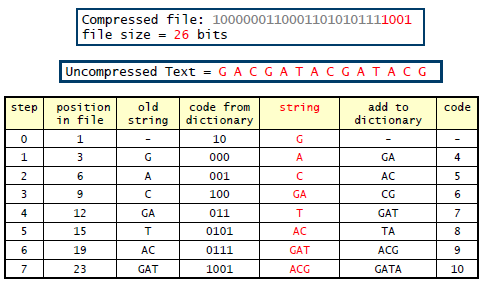
* Current string replaces least recently used string in dictionary.

**Difference between LZW decompression and compression?**

Same dictionary build but step out of phase for decompression

Decompression pseudo code

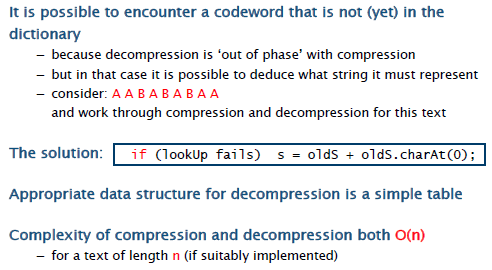




**Special case of decompression?**

Possible because a codeword is not yet in the dictionary.

Watch youtube for the example.



**String comparison**

String comparison

How similar or different are 2 strings??

* Application include:

Biology(dna and protein sequences)

File comparison(diff in unix and other similar file utilities)

Spelling correction, speech recognition,…

A more precise formulation:

* Given strings s=s0s2..sm-1 and t0t2…tn-1 of lengths m and n. what is the smallest number of basic operations needed to transform s to t??

Basic operations for transforming strings:

* Insertion, deletion of single character and substitution of one character by another

Distance between s and t is defined to be smallest number of basic operations needed to transform s to t.

