LZW Coding

LZW coding is conceptually very simple. At the onset of the coding process, a codebook or "dictionary" containing the source symbols to be coded is constructed. For 8-bit monochrome images the first *256* words of the dictionary are assigned to the gray values *0, 1 , 2, ... , 255*. As the encoder sequentially examines the images pixels graylevel sequences that are not in the dictionary are placed in algorithmically determined (e.g. the next unused) locations. If the first two pixels of the image are white, for instance, sequence *"255-255"* might be assigned to location *256* the address following the locations reserved for graylevels 0 through *255* the next time that two consecutive white pixels are encountered, code word *256,* the address of the location containing sequence *255-255* is used to represent them. If a *9-bit* , *512* word dictionary is employed in the coding process, the original *(8+8)* bits that were used to represent the two pixels are replaced by a single *9-bit* code word. Cleary, the size of the dictionary is an important system parameter. If it is too small, the detection of matching gray-level sequences will be less likely, if it is too large the size of the code words will adversely affect compression performance.

Example: *Consider the following 4*4, 8-bit image (256 gray)*

39	*39*	*126*	*126*
39	*39*	*126*	*126*
39	*39*	*126*	*126*
39	*39*	*126*	*126*

Explain details the steps involved in coding its 16 pixels.

A 512 word dictionary with the following starting content is assumed:

Dictionary Location	***Entry***
0	*0*
1	*1*
.	.
.	.
.	.
255	*255*
256	-
.	.
.	.
.	.
511	-

Location 256 through 511 are initially unused

The image is encoded by processing its pixels in a left to right, top to bottom manner. Each successive graylevel value is concatenated.

Pixel being ***Processed***	***Encoded*** ***Output***	***Dictionary*** ***Location***	***Dictionary*** ***Entry***
39	-		
39	39	256	39-39
126	39	257	39-126
126	126	258	126-126
39	126	259	126-39
39	-		
126	256	260	39-39-126
126	-		
39	258	261	126-126-39
39	-		
126	-		
126	260	262	39-39-126-126
39	-		
39	259	263	126-39-39
126	-		
126	257	264	39-126-126
	126		

Example: *Use LZW algorithm to encode the following message " BET BE BEE BED BEG"*

Step	***Char***	***Code output***	***New Code of Dictionary***	
1	Space	-	already exist	
2	B	Space	SpaceB	256
3	E	B	BE	257
4	T	E	ET	258
5	Space	T	TSpace	259
6	B	-		
7	E	SpaceB	SpaceBE	260
8	Space	E	ESpace	261
9	B	nothing		
10	E	nothing		
11	E	SpaceBE	SpaceBEE	262
12	Space	nothing		
13	B	ESpace	ESpaceB	263
14	E	nothing	BE	already exit in table
15	D	BE	BED	264
16	Space	D	DSpace	265
17	B	nothing	SpaceB	already exist
18	E	nothing	SpaceBE	already exist
19	G	SpaceBE	SpaceBEG	266
20	End			