

## O algoritmo LZW

O algoritmo LZW é um algoritmo de codificação de palavras cuja idéia principal é ir construindo um dicionário de símbolos ou palavras conforme o texto ou a informação vai sendo processado pelo algoritmo. No início da codificação de um texto (ou imagem), o dicionário LZW possui inicialmente como palavras apenas os 256 caracteres da tabela ASCII (ou, no caso de uma imagem, a quantidade de tons que compõe a paleta da imagem).

Durante o processo de análise do texto, o dicionário vai criando novas palavras a partir de combinações de pelo menos 2 caracteres (ou tons) encontrados no texto (ou na imagem). A cada nova combinação encontrada, é criada uma nova entrada no dicionário que será utilizada para codificações futuras. Ao final do processo de codificação, teremos um dicionário de combinações de palavras ou tons que poderá ser utilizado para codificar o texto ou a imagem, possivelmente reduzindo a quantidade de espaço gasto para seu armazenamento.

O algoritmo LWZ funciona da seguinte forma:

```
p = nulo;
enquanto ( leia um caracter c )
{
    se pc existir no dicionário
        p = pc;
    senão
        adicione pc ao dicionário;
        exiba o código para p;
        p = c;
}
```

Onde p é a palavra ou combinação de tons a ser codificada e c o caracter ou tom do pixel atual analisado.

Exemplificando, se quiséssemos codificar a seqüência de caracteres "**^WED^WE^WEE^WEB^WET**", por exemplo, teríamos ao final da execução do algoritmo o seguinte dicionário de palavras:

p	c	saída	índice	símbolo
nulo	^			
^	W	^	256	<b>^W</b>
W	E	W	257	<b>WE</b>
E	D	E	258	<b>ED</b>
D	^	D	259	<b>D^</b>
^	W			
^W	E	256	260	<b>^WE</b>
E	^	E	261	<b>E^</b>
^	W			
^W	E			
^WE	E	260	262	<b>^WEE</b>
E	^			
E^	W	261	263	<b>E^W</b>
W	E			
WE	B	257	264	<b>WEB</b>
B	^	B	265	<b>B^</b>
^	W			
^W	E			
^WE	T	260	266	<b>^WET</b>
T	EOF	T		

Ou seja, após a execução do algoritmo, teríamos um dicionário composto dos símbolos de 0 a 255 da tabela de caracteres ASCII e mais algumas entradas correspondendo a combinações de caracteres encontradas no texto, indo de 256 (correspondendo à combinação  $\text{^w}$ ) até a entrada 266 (correspondendo à combinação  $\text{^wET}$ ). Desta forma, ao utilizar este novo dicionário para codificar o texto, estaríamos reduzindo uma entrada de 17 caracteres (o texto) para uma entrada de 7 códigos, sendo que cada código precisaria apenas de 1 bit a mais (9 bits) para ser armazenado, ao invés de 8 bits. Também seria necessário algum espaço para armazenar as novas entradas criadas no dicionário.

#### Referências:

- <http://homepages.inf.ed.ac.uk/rbf/HIPR2/> (tutorial com tudo sobre análise de imagens)
- <http://www.cs.cf.ac.uk/Dave/Multimedia/node214.html> (algoritmo de compressão LZW)
- <http://www.dogma.net/markn/articles/lzw/lzw.htm> (compressão LZW)