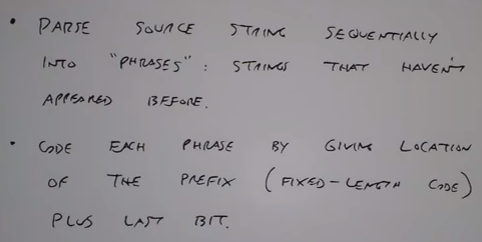
Lempel Ziv Coding:



Parsing symbols that are never seen before in a sequence:



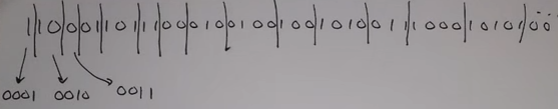




So on…



Now assign each of these phrases to a fixed length code:

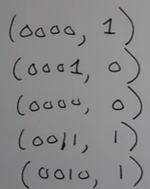


Consider a null set to be:



We will now assign the location of the prefix (phrase seen before) and the new bit (that makes it a new phrase).

Result for the first 5 phrases:



Doesn’t look like a good compression algorithm but it is excellent at squeezing down information.

* But it works really well for extremely long strings of data.
* And there is no pmf as in Huffman coding.
* It exploits how symbols occur together.
* For strings having redundancy each of them do not have to coded individually.

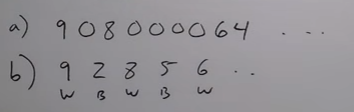
***Run Length Coding:***

Here the idea is to code the number of 0’s between successive 1’s. In other words code the lengh of continuous 0/1 runs.

Consider an example:



There are two ways of solving it:



Works well for line art and encoding fax documents.

To get something better we can apply Huffman code to this run length result