Compression

**Compression**

Compression software reduces the size of files. There are various reasons why the size of a file might need to be reduced:

* Less storage space required
* Faster download times – improving online experience
* Faster steaming speeds of video/audio files

There are two ways in which compression software might reduce the size of a file.

***Lossy Compression***

This is when unrequired data is removed from a file. MP3s are an example of this where sound quality may reduce but not to a point which is noticeable by the listener.

***Lossless Compression***

This is when data is temporarily removed from the file, but added back (rebuilt) when the file is to be used again. Zip files are an example of this. They will need to be unzipped (extracted) to be useable again.

**Lossy Compression**

As already stated, this is when unrequired data is removed from a file.

Regarding MP3s, certain frequencies that are not noticeable are removed reducing the data that the file contains but not giving any ‘noticeable’ drop in sound quality.

When chatting online or via mobile phone networks, lossy compression is used to ensure that only a small amount of bandwidth is used and although sound quality reduces, it doesn’t affect the ability to understand the other person.

Images are often compressed using lossy compression techniques. Details of the image may be lost but not impairing the overall quality. Lossy compression is especially important on websites where page load speeds can be seriously affected with image files that are large in size.

Lossy compression results in a much smaller file, compared to the lossless compression method.

**Lossless Compression**

Lossless compression although removes data temporarily, will later recreate the file exactly as it was.

This type of compression looks at the sequencing of the data stored within the file and constructs an algorithm that can later reconstruct the file by reproducing the sequence of data.

This type of compression doesn’t remove as much data at lossy compression, but does allow the size to reduce and still retain the file’s data.

**Run Length Encoding (lossless example)**

Run length encoding is a simple form of lossless compression and demonstrates the nature of how lossless compression seeks to find patterns / sequences of data that can be reconstructed at a later date.

Consider the following terrible attempt at a 1 bit image of a smiley:



If we take the 2nd row for example, the data would ordinarily be 5 bits in length - 00000.

However, run length encoding would focus on the pattern of 5 white pixels in a row and so would store the data (whilst the file is compressed) as 5W or 101 0, which uses 1 fewer bits.

This is the principle of Run Length Encoding. As images increase in size, generally the patterns of similar shades increase in length and so the data required to store the pattern gets much smaller in relation to the actual file’s data.

**Dictionary Coding (lossless example)**

Dictionary coding is another form of lossless compression which looks to replace the file’s data with a reference to what the data is.

Imagine a dictionary which contains all of the words in the English language. Imagine this dictionary was contained in a table with every word having a unique indexed location.

Consider the following string, 23 bytes in size: “I love Computer Science”

With dictionary coding, these words could be replaced with their reference (index location) in the dictionary. So the data would be stored (whilst being compressed) as 11000 110110 1010101 1100011 which is just 25 bits. Which is far smaller. Yes, the dictionary would have to accompany the compressed file, however, for large files, the size of the dictionary doesn’t have a massive impact on the size of the compressed file.

***Keywords / Key Terms:***

**Compression:** The process of reducing the size of a file*.*

**Lossy Compression:** A compression method which reduces the size of a file by permanently removing some of the file’s data*.*

**Lossless Compression:** A compression method, which reduces the size of a file by temporarily removing some of the file’s data.