Data Compression

Compressing Data

Data Compression makes the file size needed to store a digital file smaller.

This means that the amount of memory required to store the data is less. The file size is reduced.

This means that a compressed file will be smaller in terms of **Bytes**.

There are reasons why this is a useful thing to do:

- the file takes less space to store, saving storage space on a hard drive
- files are smaller so they open more quickly – there are less Bytes to open
- quicker to transfer files, for example it's quicker to download a file on a network or to send it by e-mail
- less bandwidth is required to transfer the file over the internet, meaning audio and video files are more suitable for streaming
- reduction of network congestion – there are less Bytes being transferred so network speeds will be quicker for other purposes

Compression was really important when internet speeds were much slower than they are today. Although data transfer speeds have increased, the amount of data now being transferred means that compression remains an essential part of modern life.

At least 1.8 billion images are estimated to be uploaded to social media sites every day. This is a huge amount of data. It's useful if the amount of data needing to be transferred and stored can be reduced.

In some cases there are limits on the file size which can be transferred by, for example, e-mail. For example, as of December 2022 Gmail limits attachments to 25MB.

Different Compression Methods

There are different ways to compress a file. Each method uses a distinct **algorithm** to reduce the amount of data required.

MP3 compression is an example of a compression algorithm. Audio file sizes can be reduced by 75% to 95% using MP3 by lowering the sample rate to between 16kHz and 24kHz and by discarding data considered to be beyond the range of human hearing – losing data but with little impact on sound quality.

This means that MP3 is an example of a **lossy compression algorithm**. Some data is lost permanently when the audio is compressed. JPEG is another type of lossy compression algorithm.

MP3 compression can reduce an audio file from a 50MB WAV file to a 4MB MP3 file

Because of their reduced file size, MP3 files are very portable. They can be downloaded much more quickly and many more can be stored on a media player or phone.

Activity 1:

a) Define the term **data compression**
b) Give **four** reasons why compression is used. Include some detail about why each is useful
c) Explain how audio files can be compressed using the MP3 compression algorithm
d) What does **lossy compression** mean?

Run Length Encoding (RLE)

Run Length Encoding is a **lossless compression algorithm**. It can be used to compress data, including **bitmap image files**.

RLE works by using **frequency/data pairs** to represent data. For example:

RRRGGBBBBGRRRR

becomes

3 R 2 G 4 B 1 G 4 R

Each **frequency/data pair** represents a **run** of identical data items. For examples, in a large bitmap image file there may be lots of identical blue pixels in the sky. Rather than record each blue pixel separately, RLE can record a **run** of pixels in a way that is often shorter, compressing the data.

> **For example:** RRRRRR is six values. Using RLE this becomes 6R – just two values are needed to represent the same data, four less.

The idea of a **run** of identical values is an important part of RLE. It works well when there are lots of runs of identical values.

RLE doesn't always reduce the file size. If there aren't lots of runs of identical values, RLE can actually increase the size of file:

> **For example:** RGBRGB is six values. Using RLE this becomes 1R1G1B1R1G1B – taking 12 values to store the same data.

Lossless compression algorithms reduce the file size without needing to destroy any of the data. PNG and GIF are examples of lossless image file algorithms.

Repeated data values are called a **run** – hence **Run Length Encoding**.

In these examples, think of R as red, G as green and B as blue. The number of different colours available in any bitmap image depends on the colour depth.

Activity 2:

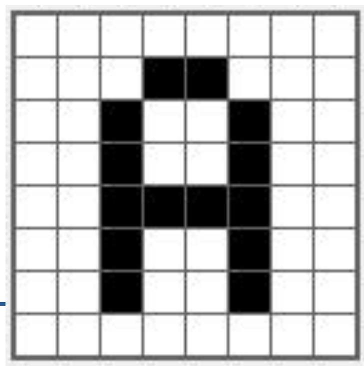
a) Explain how RLE uses **frequency/data pairs** to encode data. An example will probably help
b) Explain why **runs** of identical values are so important when using RLE
c) True of false: RLE always produces a smaller file size. Explain your answer.

RLE and Bitmap Images:

Black and white bitmap images can be compressed using RLE.

In the image on the right, the first line of pixels can be represented as 8 1 after applying RLE – 8 pixels all having the value 1 (for white).

The second line is 3 1 2 0 3 1. The third line is 2 1 1 0 2 1 1 0 2 1



Activity 3:

a) Apply the RLE method to each of the following data:
   (i) RRRGGGGB   (ii) 11110011   (iii) CYYYMMKKYYCCKKKKKK   (iv) WYSIWYG
b) Data has been encoded using RLE. Decode the data:
   (i) 4 G 2 R 2 G   (ii) 5 0 2 1 1 0 3 1   (iii) 1 G 2 O 1 D 1 B 2 O 1 K
c) Write down the RLE coding for lines 5, 6 and 8 of the A image above. Remember, 1 is a white pixel, 0 is a black pixel.
d) RLE is a **lossless compression algorithm**. Explain what this means