

Data Compression

Compressing Data

Data Compression makes data smaller – which has the effect of reducing the amount of memory space required to store the data. Essentially, the file size is reduced.

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

There are reasons why this might be desirable:

- the file takes less space to store, saving storage space in RAM
- files can be accessed more quickly – there are less Bytes to open
- quicker to transfer files, for example over a network or 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 speeds will be quicker for other purposes

Although internet network speeds have increased, the types of data now being transferred means that compression is an essential part of modern life.

Different Compression Methods

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

A major distinction is between **lossy** and **lossless** compression methods. Lossy methods lead to data being lost; lossless methods don't.

MP3 compression is an example of a lossy 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.

The development of MP3 in the 1990s allowed large digital files stored on CDs to be massively reduced in size. This led to the rise of digital piracy – songs being made available illegally on the internet. A 50MB WAV files could be reduced to a 4MB MP3 file, allowing music to be shared. This led to record companies and artists losing money. More recently, streaming services such as Spotify have used MP3 compression whilst paying royalties to artists and record companies.

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, Gmail limits attachments to 25MB.

Another example of lossy compression is **JPEG**. File sizes can be reduced by up to 90% by grouping pixels together without a noticeable loss in image quality.

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.

Activity 1:

- a) Define the term **compression**
- b) Give three reasons why compression is used
- c) Using MP3 compression as an example, explain the impacts of data compression

Run Length Encoding (RLE)

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

RLE works by combining data items with the same value together in **frequency/data pairs**. In the sequence below there are repeated data values. This is where RLE can reduce file size:

RRRGRRRRBBBBBBGGGGGRRRRRRRRRB

When RLE is applied the data can be compressed to:

3 R 2 G 3 R 7 B 6 G 9 R 1 B

The frequency/data pair simply encodes the number of repeated data values followed by the data value itself. So 3 R means 3 repeated Rs.

Black and white bitmap images can be compressed using RLE.

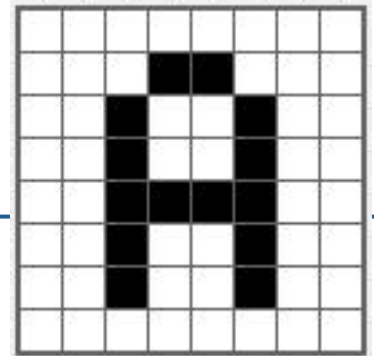
In the image on the right, the first line of pixels can be represented as 8 1 after applying RLE - 8 pixels all having the value 1 (for white).

The second line is 3 1 2 0 3 1

The third line is 2 1 1 0 2 1 1 0 2 1

Remember, in bitmap images a black pixel has the value 0, a white pixel by 1.

The standard bitmap binary coding for the first line of the image below is 11111111



Activity 2:

- 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
- Write down the Bitmap coding and RLE coding for lines 5, 6 and 8 of the A image above
- Explain how RLE uses **frequency/data pairs** to compress data

The Limits to RLE

RLE is good at compressing data when the same data item is repeated. This will almost always end up producing a smaller file when the RLE algorithm is applied.

For example, line 1 of the A image can be compressed from 11111111 to 8 1 – a much shorter representation.

When there is less repetition, however, the use of RLE can actually produce larger file sizes: line 3 of the bitmap is 11011011 which, when RLE is applied, becomes 2 1 1 0 2 1 1 0 2 1 – requiring more data to store.

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

The effects of compression are much more noticeable when applied to much larger files.

Activity 3:

- Apply the RLE method to each of the following data:
(i) RRRGGGGB (ii) 11110011 (iii) CYYYMMKKYYCCKKKKKK (iv) WYSIWYG
- Explain why applying RLE to data does not always produce a significantly smaller file size
- RLE is a lossless compression algorithm. Explain what this means