Compressing Data

There be LOTS of data

As the world around us becomes increasingly computerized, digital data is constantly being created and stored.

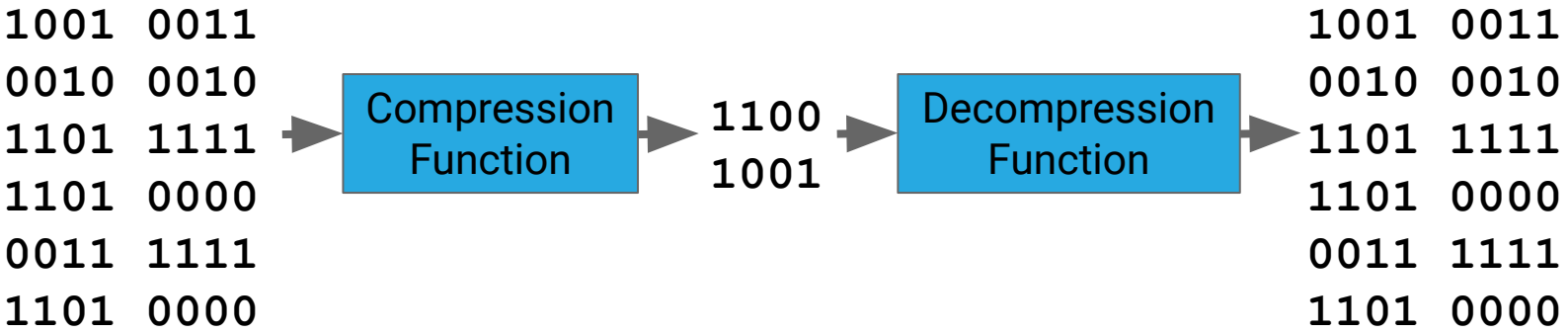
Physical media is always improving - more efficient ways to store digital data are always being developed and improved upon. We are able to store increasingly large amounts of data in decreasingly large areas, which is always great.

We need to have more than just hardware solutions to this problem of increasing data, though - we need to use software to shrink down our data so that it takes up less space!

Data Compression

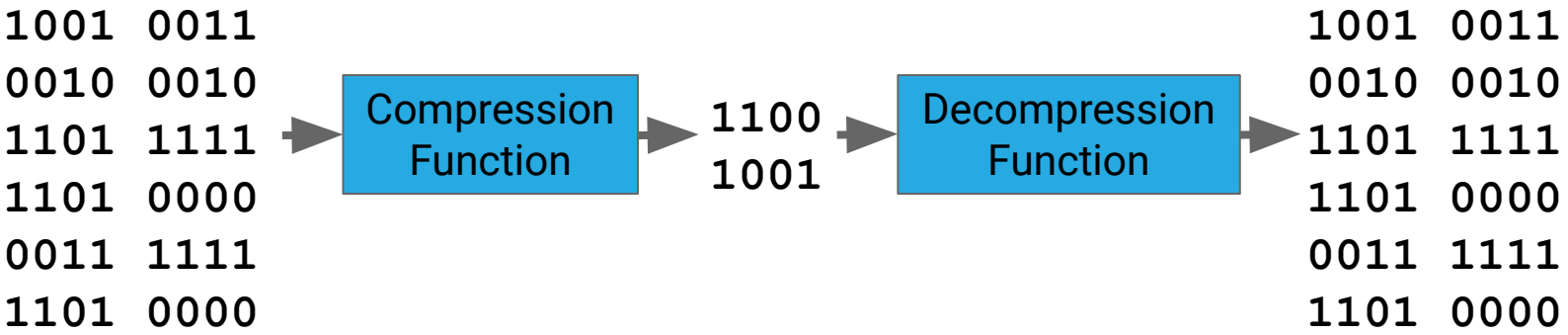
Data Compression is the process of encoding information using **fewer** bits than were used in the original representation.

We can use algorithms to **compress** information and use less bits for storage, then later we can **decompress** the information for full, proper viewing!



Data Compression

Lossless Compression happens when we **compress** information and then **decompress** it to get the same exact value. There are a variety of different algorithms for doing all kinds of different compression!



Data Compression

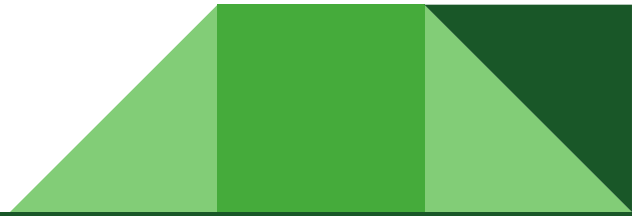
Just about every single piece of digital media is **compressed** before being stored or sent, which saves space and makes working with it faster!

Movies, music, images, and text are all compressed when they are stored digitally.

Why Bother?

Here are a few reasons why compression is a useful thing to do:

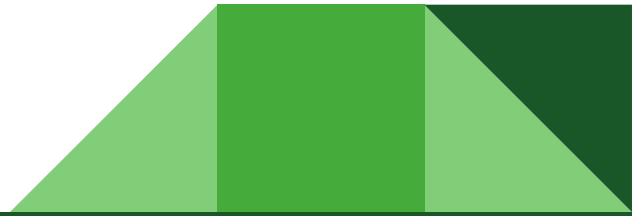
- Save Storage Space



Why Bother?

Here are a few reasons why compression is a useful thing to do:

- Save Storage Space
- Save time/bandwidth when transmitting data
  - If there are less bits to send, the process can be finished more quickly!



Why Bother?

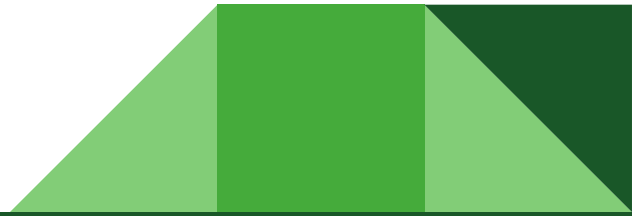
Here are a few reasons why compression is a useful thing to do:

- Save Storage Space
- Save time/bandwidth when transmitting data
  - If there are less bits to send, the process can be finished more quickly!
- Trade off storage, which is (in computer terms) expensive and slow, for computing power, which is cheap and fast
  - Our phones don't have the memory capacity to store the raw footage for a movie - those are usually hundreds of GB!
  - What our phones ***do*** have is enough **computational power** to decompress a 1GB compressed movie when we want to watch it

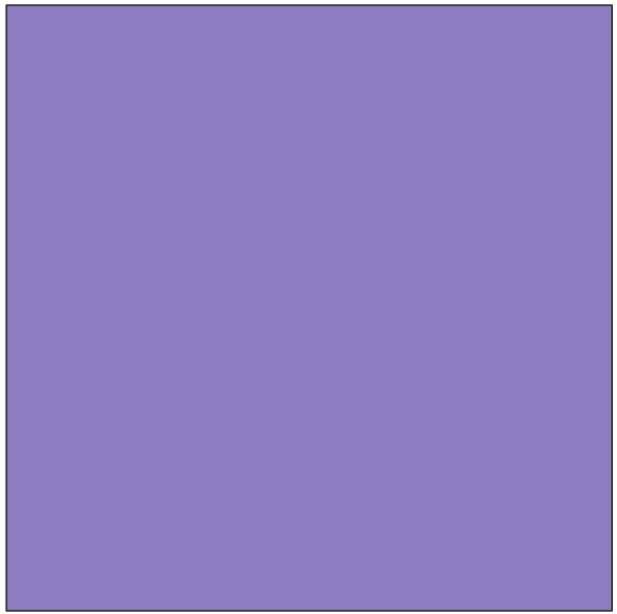
How Compression Works

Compression works by finding repeated patterns in the data, then replacing those patterns with a placeholder.

Patterns are more common than you'd think in things like text, video, and images!

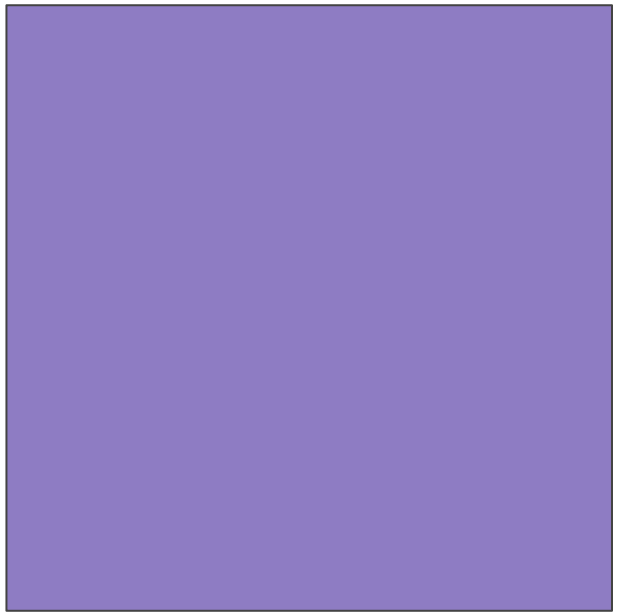


Compression Example



100 pixels
24 bits per pixel
2400 bits

Compression Example

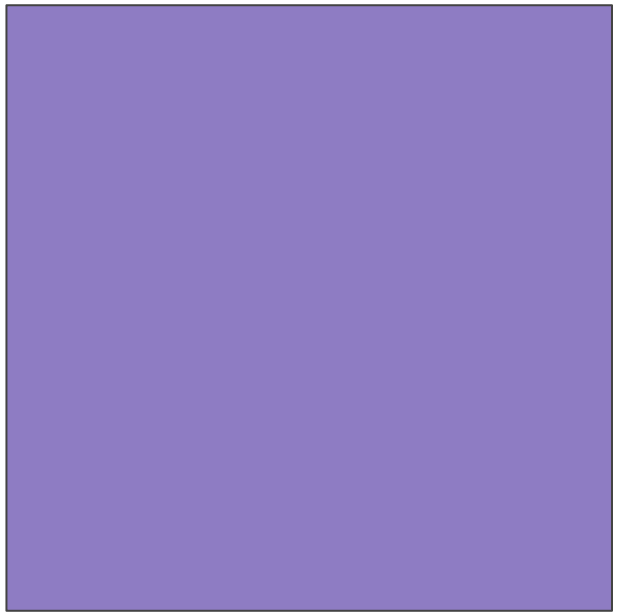


100 pixels
24 bits per pixel
2400 bits

x 100

1 pixel (24 bits)
# of repeats (32 bits)
56 bits

Compression Example



100 pixels
24 bits per pixel
2400 bits

x 100

1 pixel (24 bits)
# of repeats (32 bits)
56 bits

Text Compression - Run Length Encoding

An example of a Text Compression algorithm is Run Length Encoding. This algorithm works well when there is a string with a character repeated many times.

`HEEEEEEEEEYHIIIIIIIIIII`

Text Compression - Run Length Encoding

An example of a Text Compression algorithm is Run Length Encoding. This algorithm works well when there is a string with a character repeated many times.

```
HEEEEEEEEEYHIIIIIIIIIII

H1
```

Text Compression - Run Length Encoding

An example of a Text Compression algorithm is Run Length Encoding. This algorithm works well when there is a string with a character repeated many times.

```
HEEEEEEEEEYHIIIIIIIIIII

H1E9
```

Text Compression - Run Length Encoding

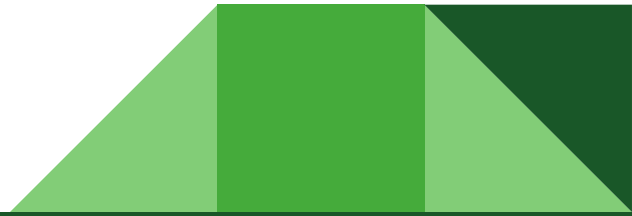
An example of a Text Compression algorithm is Run Length Encoding. This algorithm works well when there is a string with a character repeated many times.

```
HEEEEEEEEEYHIIIIIIIIIII

H1E9Y1
```

Text Compression - Run Length Encoding

An example of a Text Compression algorithm is Run Length Encoding. This algorithm works well when there is a string with a character repeated many times.

```
HEEEEEEEEEYHIIIIIIIIIII

H1E9Y1H1
```



Text Compression - Run Length Encoding

An example of a Text Compression algorithm is Run Length Encoding. This algorithm works well when there is a string with a character repeated many times.

```
HEEEEEEEEEYHIIIIIIIIIII

H1E9Y1H1I11
```

Text Compression - Run Length Encoding

An example of a Text Compression algorithm is Run Length Encoding. This algorithm works well when there is a string with a character repeated many times.

`HEEEEEEEEEYHIIIIIIIIIII`

`H1E9Y1H1I11`

Original: 22 characters * 8 bits/character = **176** bits

Compressed: 11 characters & 8 bits/character = **88** bits

Text Compression - Run Length Encoding

An example of a Text Compression algorithm is Run Length Encoding. This algorithm works well when there is a string with a character repeated many times.

```
HEEEEEEEEEYHIIIIIIIIIII
```

```
H1E9Y1H1I11
```

H

Text Compression - Run Length Encoding

An example of a Text Compression algorithm is Run Length Encoding. This algorithm works well when there is a string with a character repeated many times.

`HEEEEEEEEEYHIIIIIIIIIII`

`H1E9Y1H1I11`

HEEEEEEEEE

Text Compression - Run Length Encoding

An example of a Text Compression algorithm is Run Length Encoding. This algorithm works well when there is a string with a character repeated many times.

`HEEEEEEEEEYHIIIIIIIIIII`

`H1E9Y1H1I11`

HEEEEEEEEEY

Text Compression - Run Length Encoding

An example of a Text Compression algorithm is Run Length Encoding. This algorithm works well when there is a string with a character repeated many times.

`HEEEEEEEEEYHIIIIIIIIIII`

`H1E9Y1H1I11`

HEEEEEEEEEYH

Text Compression - Run Length Encoding

An example of a Text Compression algorithm is Run Length Encoding. This algorithm works well when there is a string with a character repeated many times.

`HEEEEEEEEEYHIIIIIIIIIII`

`H1E9Y1H1I11`

HEEEEEEEEEYHIIIIIIIIIII

Measuring Compression

The space savings is a measure of how much smaller the compressed data is compared to the original!

$$\text{Space savings} = \left(1 - \frac{\text{size of compressed}}{\text{size of original}}\right) \times 100$$

Text Compression - Run Length Encoding

An example of a Text Compression algorithm is Run Length Encoding. This algorithm works well when there is a string with a character repeated many times.

`HEEEEEEEEEYHIIIIIIIIIII`

`H1E9Y1H1I11`

Original: 22 characters * 8 bits/character = **176** bits
Compressed: 11 characters & 8 bits/character = **88** bits

Text Compression - Run Length Encoding

An example of a Text Compression algorithm is Run Length Encoding. This algorithm works well when there is a string with a character repeated many times.

`HEEEEEEEEEYHIIIIIIIIIII`

`H1E9Y1H1I11`

Original: 22 characters * 8 bits/character = **176** bits
Compressed: 11 characters & 8 bits/character = **88** bits

Space Savings = (1 - 88/176) x 100

Text Compression - Run Length Encoding

An example of a Text Compression algorithm is Run Length Encoding. This algorithm works well when there is a string with a character repeated many times.

`HEEEEEEEEEYHIIIIIIIIIII`

`H1E9Y1H1I11`

Original: 22 characters * 8 bits/character = **176** bits
Compressed: 11 characters & 8 bits/character = **88** bits

Space Savings = **50%**

Text Compression - Run Length Encoding

An example of a Text Compression algorithm is Run Length Encoding. This algorithm works well when there is a string with a character repeated many times.

`HEEEEEEEEEYHIIIIIIIIIII`

`H1E9Y1H1I11`

Original: 22 characters * 8 bits/character = **176** bits
Compressed: 11 characters & 8 bits/character = **88** bits

Space Savings = **50%**

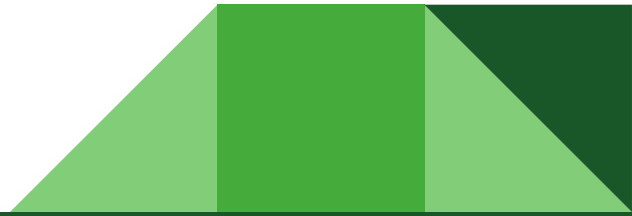
The compressed data is *50%* smaller!

Compression Algorithms

Run Length Encoding is only 1 example of a compression algorithm - all kinds of algorithms exist for compressing all kinds of different data!

JPEG is a common image compression format

zip is good at compressing large amounts of text



Compression Algorithms

In order for a Compression Algorithm to work super well, it needs to have a matching Decompression Algorithm to reverse the process.

Our computers use a variety of different **file formats** to keep track of which kind of compression is being used to store the data in that file.

.zip, **.jpeg**, **.tar**, **.txt**, etc!