15.1 Compression

Compression basics

Compression reduces the size of a file. Ex: An image/photo file may be 4 megabytes before compression, but only 1 megabyte after.

Compression enables a user to store more files on a drive. Compression also speeds up transfers of files via the Internet.

Compression can be applied to any file, but is commonly applied to audio, image, and video files because such files are naturally large. Most apps for such files automatically compress and decompress those files.

PARTICIPATION ACTIVITY 15.1.1: Compression reduces file size, enabling storage of more files, and faster file transfers.

Animation captions:

1. Compression reduces the size of a file.
2. Compression enables more files to be stored on a drive.
3. Compression also speeds up file transfers, like web downloads.

PARTICIPATION ACTIVITY 15.1.2: Compression basics.

1) By reducing a file's size, compression enables more files to be stored on a drive.
   - ○ True
   - ○ False

2) Compression has the drawback of increasing the time to download a file via the web.
   - ○ True
   - ○ False

Dictionary-based compression

A common approach to compressing data in a file is to create a custom dictionary for that data. In compression, a ***dictionary*** is a table of shorthand versions of longer data. Ex: Given a dictionary of 1: Department, 2: of, and 3: Redundancy (1, 2, and 3 are the shorthand items), then "1 2 3 1" is short for "Department of Redundancy Department". The compressed file will contain the dictionary itself plus the data in shorthand form, which combined is hopefully smaller than the original file. A typical text file might be compressed by 50% or more.

An ***LZ compression*** algorithm (named for creators Lempel and Ziv) examines data for long repeating patterns such as phrases, and creates a dictionary entry for such patterns. Alternatively, a ***Huffman encoding*** algorithm measures the frequency of each data item like each letter, and gives the most frequent items a shorter bit encoding (like the letters "a" and "e"), with least frequent items getting a longer encoding (like letters "q" and "z"). Many compression techniques combine LZ and Huffman algorithms.

PARTICIPATION ACTIVITY 15.1.3: Compression using an LZ algorithm dictionary approach.

Animation captions:

1. LZ compression algorithms search for repeating patterns, creating a dictionary entry for each.
2. The compressed file contains the dictionary, and the data in shorthand.
3. The compressed file, having both the dictionary and the shorthand data, may be smaller. Compressions of 50%-90% are common.

PARTICIPATION ACTIVITY 15.1.4: Dictionary-based compression.

1) Given the dictionary below for an LZ algorithm, what uncompressed phrase do these numbers represent: 1 3 2

   1: Oh
   2: gosh
   3: my

   - ○ Oh gosh my
   - ○ Oh my gosh
   - ○ Cannot be determined

2) Given the following uncompressed text, which is a more reasonable LZ dictionary?

   He sells sea shells.

- 1: ells
- 1: e
  2: s

3) How much smaller is the compressed text than the original? Use this equation: (uncompressed - compressed) / uncompressed. Count each character, including spaces.

   Original: To infinity and beyond!
   Compressed: To 1 and 2!

   - 52%
   - 48%
   - 0%

4) Given the below Huffman coding dictionary, what uncompressed letters do these bits represent: 01 01 01 001 00011111 01

   01: a
   001: b
   00011111: z

   - a b z a
   - a a a b z a

Zip files

A ***zip file*** is a common file format for combining multiple files into one file and that usually involves compression. Ex: Using a zip app on a Mac, a 20 kbyte Microsoft Word file and a 85 kbyte PDF file are compressed into a single 98 kbyte zip file..

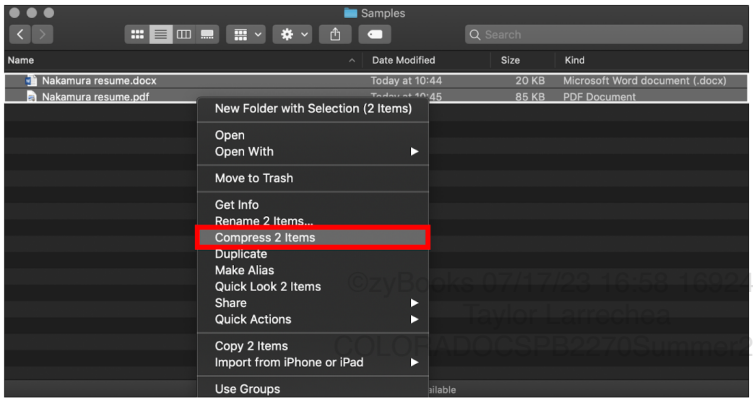
Figure 15.1.1: File compression/uncompression on a Mac computer.

Compress

1. Right-click on file(s)
2. Select “Compress”
3. Compressed file appears in folder

Uncompress

1. Double-click on zip file
2. Uncompressed file appears in folder



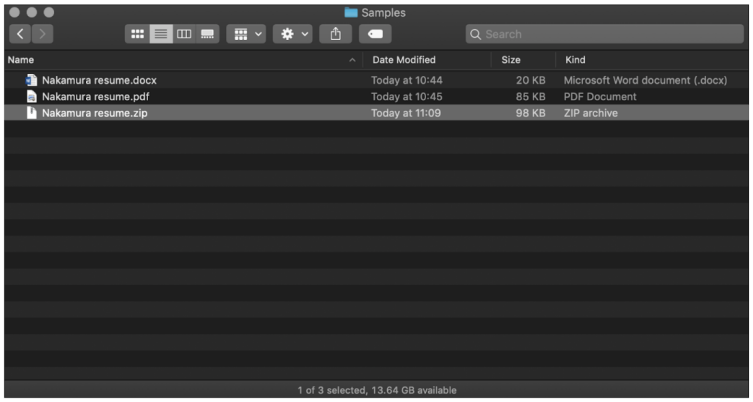
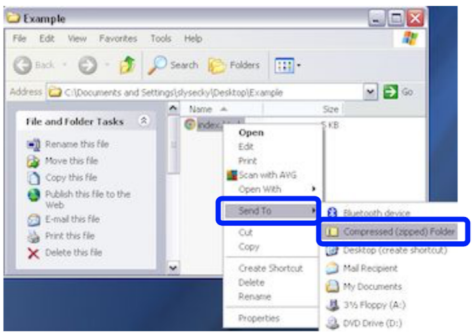


Figure 15.1.2: File compression on a Windows computer.

Compress

1. Right-click on file
2. Select "Send To"
3. Select "Compressed (zipped) Folder"
4. Compressed file appears in folder



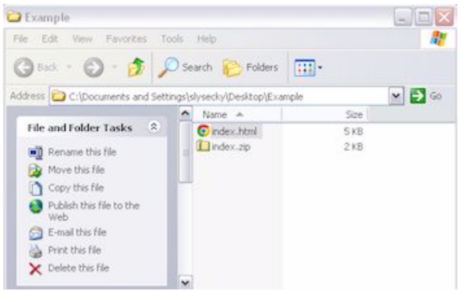
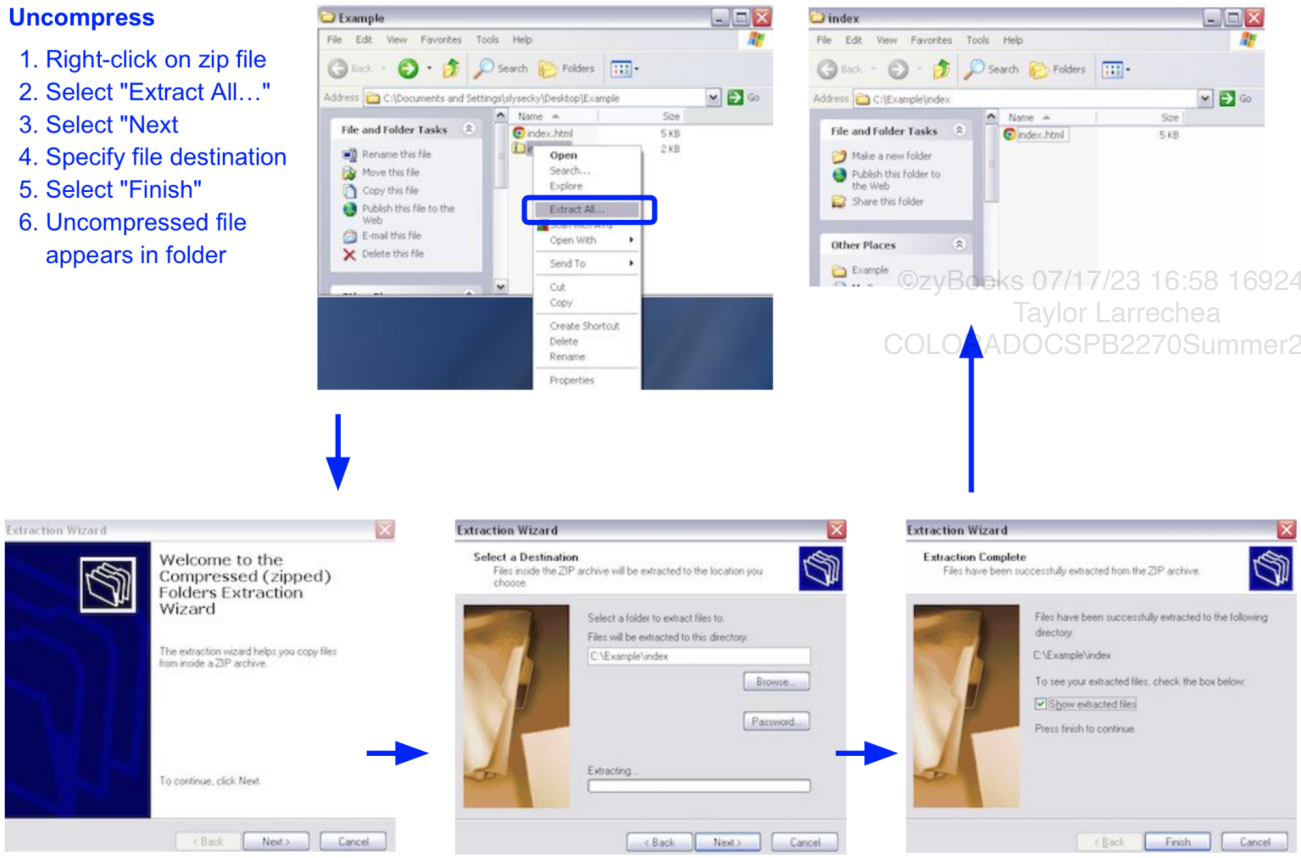


Figure 15.1.3: Uncompressing a file on a Windows computer.

Uncompress

1. Right-click on zip file
2. Select "Extract All…"
3. Select "Next
4. Specify file destination
5. Select "Finish"
6. Uncompressed file appears in folder



PARTICIPATION ACTIVITY 15.1.5: Zip files.

1) A zip file is commonly created to collect multiple files into a single file.
   - True
   - False

2) The process of creating a zip file may compress the file's contents.
   - True
   - False

Lossless versus lossy compression

Lossless compression loses no information, so that decompression yields an identical file to the original. LZ and Huffman approaches are lossless.

Lossy compression loses some information, so the decompressed file is close but not identical to the original. An example lossy compression approach is rounding. Ex: Given an original file containing 255, 64, 231, the one's place can be dropped (a form of rounding), yielding 25, 6, 23. A decompressor,

knowing of the rounding but not knowing what numbers were dropped, may append 5's, yielding 255, 65, and 235, which is close but not identical to the original. Lossy compression may be OK for data like images or audio where humans barely notice a quality difference, but is clearly not OK for precise data like text or bank account numbers since the meaning would change.

PARTICIPATION ACTIVITY 15.1.6: Lossless versus lossy compression.

1) Lossy compression means that some files that were to be compressed were lost.
   - True
   - False

2) Lossy compression is acceptable for a file containing music for casual listening.
   - True
   - False

3) Lossy compression is acceptable for a file containing phone numbers.
   - True
   - False

4) Compression is not possible unless loss of information is acceptable.
   - True
   - False

JPEG compression

JPEG is a compression approach specifically for images. An image may consist of millions of ***pixels*** (short for "picture elements"), each pixel being a colored dot. A pixel may be 3 numbers (each a byte) indicating the amount of red, green, and blue. If an image has 4 million pixels, and each pixel requires 3 bytes, then a single uncompressed image is basically just a series of 12 million numbers (so 12 Mbytes). JPEG compresses the image using several techniques.

- One is to convert to the "frequency domain", which is beyond this material's scope.
- A second is to use Huffman encoding. Ex: An image with a lot of bright pixels may have millions of "255" values for pixel colors. 255 in binary is usually 11111111, but Huffman encoding may

create a dictionary entry like 01: 11111111, so those millions of 11111111's can be replaced by 01's.

- A third compression approach is to round the numbers, known as ***quantization***. So a pixel of 255 red, 64 green, and 231 blue (which is a shade of purple) may become 25, 6, and 23. Decompression might append a 5, yielding 255, 65, and 235, which is an unnoticeably different purple than the original.

(Note: The conversion to frequency domain would modify the discussion of the latter two steps, but the intuition is the same).

JPEG is lossy compression, due in part to the rounding (as well as conversion to frequency domain). Image apps may allow a user to reduce file size, achieved by losing more information, such as by doing even more rounding. The loss in quality may not be noticeable unless the image is enlarged.

Figure 15.1.4: iPhone asks a user to choose the size of the photo transmitted.

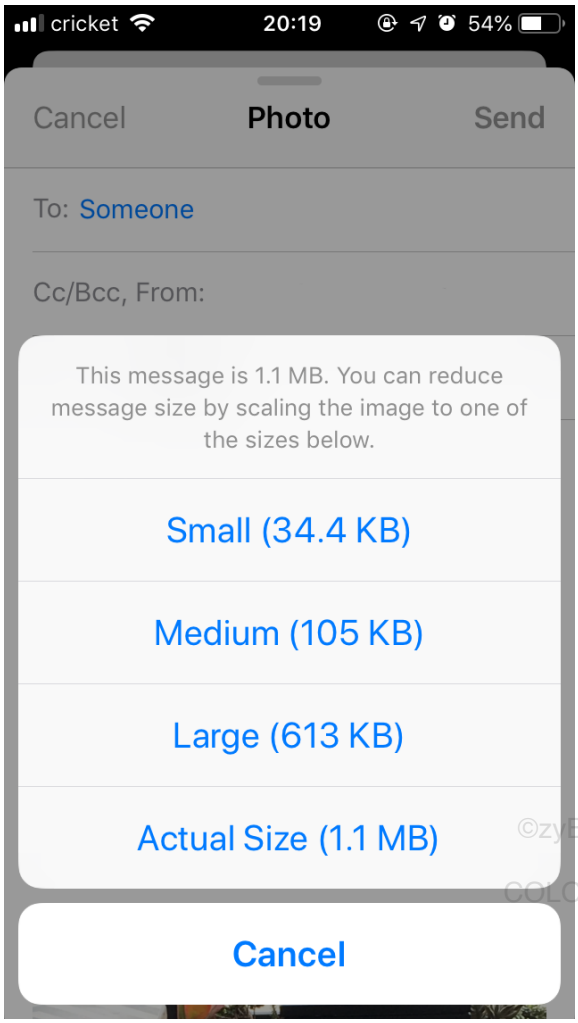


Figure 15.1.5: Mac Preview (and other computer apps) enable a user to resize an image.

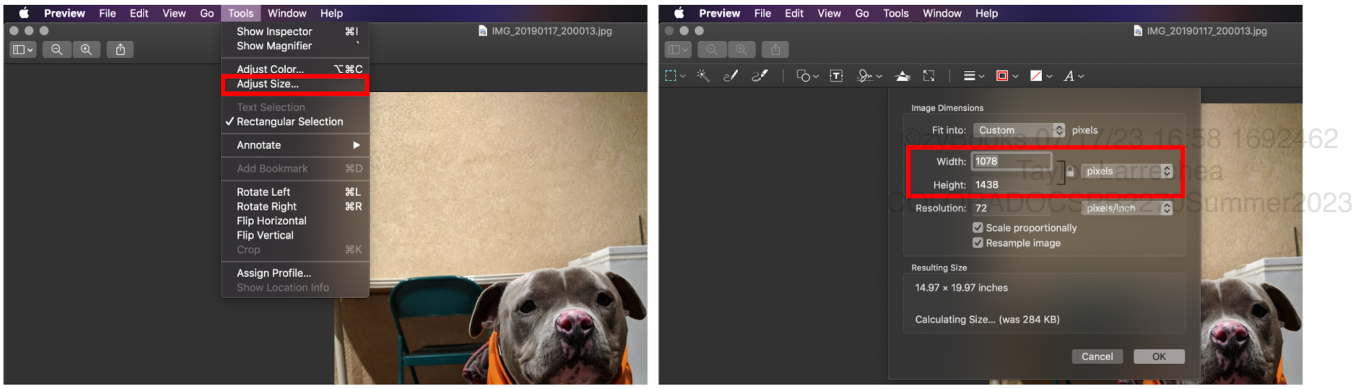


Figure 15.1.6: For a highly-compressed JPEG image, a small image displays well, but the image enlarged shows the loss in quality.

Smaller image:

Enlarging the image leads to pixelation:

PARTICIPATION ACTIVITY 15.1.7: JPEG compression.

1) JPEG is a _____ compression approach for images.
   - ○ lossless
   - ○ lossy

2) _____ is a compression approach where pixel values are rounded. Ex: 145 is rounded to 14 by dropping the one's place.
   - ○ Huffman encoding
   - ○ JPEG
   - ○ Quantization

3) Given the number of occurrences of a pixel color, which Huffman encoding yields the best compression?

Pixel value	# of occurrences
11111111	5,200,000
11110000	4,300,000
00000011	1,270,000

- 011: 11111111
  101: 11110000
  111: 00000011
- 01: 11111111
  001: 11110000
  0001: 00000011

Video compression

MPEG is a compression approach specifically for video (a recent well-known version of MPEG is MP4). ***Video*** is a series of images called ***frames***. If shown faster than about 15 frames per second (movies use 24, TVs use 30 or more), a human's vision system sees the images as a continuous video. The key idea of video compression is that successive frames differ only slightly, so a frame can be represented just as the difference from the previous frame. Ex: Frame 1 may be a full image, but Frame 2 may just be "Previous frame shifted left 2 pixels".

Clearly not all frames can be represented as the difference of another frame. Thus, such video compression sends an image frame and then perhaps 10 "predicted" frames, followed by another image frame. Note: Image frames are also compressed using image compression like JPEG.

H264 is a more recent video compression approach than MPEG, intended to reduce bits further for fast transmission of video over networks.

Video compression is lossy, in part due to the images being compressed using JPEG (which is lossy), and more so because predicted frames clearly aren't entirely accurate. Video apps may support different quality levels, with lower quality achieving smaller file sizes via use of more predicted frames and more-compression of images too. Video compression amounts of 50x-100x or more are common.

PARTICIPATION ACTIVITY 15.1.8: Video compression.

1) A video consists of a woman sitting and presenting the news. Is this video amenable to extensive compression?
   - Yes
   - No

2) A video consists of a car exploding. Is this video amenable to extensive compression?
   - Yes
   - No

3) To compress video, an app reduces frames per second to eight. Will the video quality be acceptable?
   - Yes
   - No

4) Was H264 created before MPEG?
   - Yes
   - No

Audio compression

MP3 and ***OGG*** are audio compression techniques. Audio is captured electronically as varying voltages on a wire. Those voltages are converted to numbers for digital storage. An uncompressed 3-minute song may require tens of megabytes. A ***WAV*** file stores audio uncompressed. MP3 and OGG compression use techniques (introduced above) like converting to frequency domain, quantization, and Huffman coding. A 3-minute song may be compressed to just a few megabytes.

MP3's name comes from MPEG, as MP3 was used for compressing the audio part of video files. OGG was developed as a free open-source audio compression technique.

PARTICIPATION ACTIVITY 15.1.9: Compression for video and audio files.

If unable to drag and drop, refresh the page.

MP3 WAV H264

	Video compression technique.
	Audio compression technique.
	Uncompressed audio file.

Reset

Exploring further:

- Data compression (Wikipedia)
- How file compression works (How stuff works)

15.2 Data compression

Basic compression idea

Given data represented as some quantity of bits, ***compression*** transforms the data to use fewer bits. Compressed data uses less storage, and can be communicated faster too.

The basic idea of compression is to encode frequently-occurring items using fewer bits. Ex: ASCII characters use 8 bits each, but instead more-frequently-occurring characters could use fewer bits and other characters use more bits.

PARTICIPATION ACTIVITY 15.2.1: The basic ideas of compression is to use fewer bits for frequent items (and more bits for less-frequent items).

Animation captions:

1. The text "AAA Go" as ASCII would use 6 * 8 = 48 bits. Such data is uncompressed.
2. Compression uses a dictionary of codes specifically for the data. Frequent items get shorter codes. Here, A (which is most frequent) is 0, space 10, G 110, and o 111.
3. Thus, "AAA Go" is compressed as 0 0 0 10 110 111. The compressed data uses only eleven bits, much fewer than the 48 bits uncompressed.

The example above has only four distinct characters (A, space, G, and o) so could be encoded using 2 bits (fixed-length code). However, the example is trivially simple, for learning. Actual text may use all

ASCII characters so a fixed-length code would require 8 bits per character, but with varying-length codes as above where the frequent characters in the data might use fewer bits (like 4).

PARTICIPATION ACTIVITY 15.2.2: Basic compression.

Given the following dictionary:
00000000: 00
11111111: 01
00000010: 10
00000011: 110
00000100: 111

1) Compress the following: 00000000 00000000 11111111 00000100

Check Show answer

2) Compress the following: 00000011 00000010

Check Show answer

3) Decompress the following: 00 01 00

Check Show answer

4) Does any code in the dictionary contain another code starting from the left of each code? Type yes or no. Ex: Consider a different dictionary having codes 1110 and 111; 1110 contains 111.

Check Show answer

5) Decompress the following, in which the spaces that were inserted above for reading convenience are absent: 0011000.

Check Show answer

Huffman coding

Huffman coding is a common compression technique that assigns fewer bits to frequent items, using a binary tree.

PARTICIPATION ACTIVITY 15.2.3: A binary tree can be used to determine the Huffman coding.

Animation captions:

1. Huffman coding first determines the frequencies of each item. Here, a occurs 4 times, b 3, c 2, and d 1. (Total is 10).
2. Each item is a "leaf node" in a tree. The pair of nodes yielding the lowest sum is found, and merged into a new node formed with that sum. Here, c and d yield 2 + 1 = 3.
3. The merging continues. The lowest sum is b's 3 plus the new node's 3, yielding 6. (Note that c and d are no longer eligible nodes). The merging ends when only 1 node exists.
4. Each leaf node's encoding is obtained by traversing from the top node to the left. Each left branch appends a 0, and each right branch appends a 1, to the code.

When merging, if two (or more) different node pairs would yield the same sum, the choice among those pairs is arbitrary.

PARTICIPATION ACTIVITY 15.2.4: Huffman coding example: Frequency counts.

Given the text "seems he fleed". Indicate the frequency counts.

1) s

Check Show answer

2) e

Check Show answer

3) Each of m, h, f, l, and d

Check Show answer

4) (space)

Check Show answer

PARTICIPATION ACTIVITY 15.2.5: Huffman coding example: Merging nodes.

A 100-character text has these character frequencies:
A: 50
C: 40
B: 4
D: 3
E: 3

1) What is the first merge?

- D and E: 6
- B and D: 7
- B and D and E: 10

2) What is the second merge?

- B and D: 7
- DE and B: 10
- C and A: 90

3) What is the third merge?

- DEB and C: 40

- DEB and C: 50

4) What is the fourth merge?

- None
- DEBC and A: 100

5) What is the fifth merge?

- None
- DEBCA and F

6) What is the code for A?

- 0
- 1

7) What is the code for C?

- 1
- 01
- 10

8) What is the code for B?

- 001
- 110

9) What is the code for D?

- 1110
- 1111

10) What is the code for E?

- 1110
- 1111

11) 5 unique characters (A, B, C, D, E) can each be uniquely encoded in 3 bits (like 000, 001, 010, 011, and 100). With such a fixed-length code, how many bits are needed for the 100-character text?

- 100

- 300

12) For the Huffman code determined in the above questions, the number of bits per character is A: 1, C: 2, B: 3, D: 4, and E: 4. Recalling the frequencies in the instructions, how many bits are needed for the 100-character text?

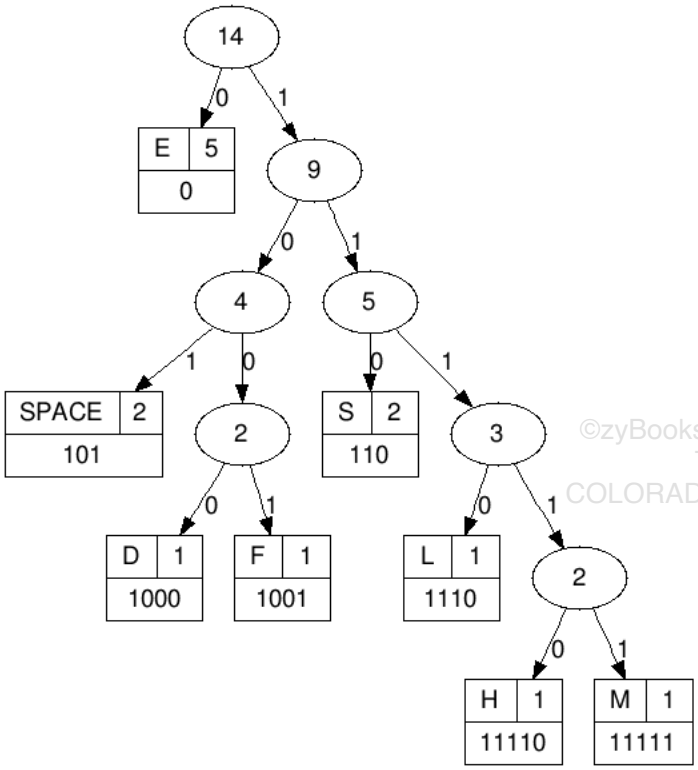
- 14
- 166
- 300

Note: For Huffman encoded data, the dictionary must be included along with the compressed data, to enable decompression. That dictionary adds to the total bits used. However, typically only large data files get compressed, so the dictionary is usually a tiny fraction of the total size.

Huffman tree web tools

This site has a tool that converts given text into a Huffman tree. For the earlier example of "seems he fleed", the site generated the following tree.

Figure 15.2.1: Huffman tree for the text: SEEMS HE FLEED.



Source: Huffman tree generator. No copyright held on generated images.

Table 15.2.1: Huffman and ASCII code table for the text: SEEMS HE FLEED.

Frequency	Chars	Huffman	ASCII
5	'E'	0	01000101
2	' '	101	00100000
2	'S'	110	01010011
1	'D'	1000	01000100
1	'F'	1001	01000110
1	'L'	1110	01001100
1	'H'	11110	01001000
1	'M'	11111	01001101

Note: CrypTool Project provides tools to generate a similar table comparing the sizes of Huffman and ASCII code.

For the text "SEEMS HE FLEED", Huffman code requires 39 bits while the ASCII code requires 112 bits.

- Huffman: 110 0 0 11111 110 101 11110 0 101 1001 1110 0 0 1000
- ASCII: 01010011 01000101 01000101 01001101 01010011 00100000 01001000 01000101 00100000 01000110 01001100 01000101 01000101 01000100

PARTICIPATION ACTIVITY 15.2.6: Huffman and ASCII code.

1) E's Huffman code is ____ .

- ○ 0
- ○ 01000101

2) _____ bits are needed to encode the 'SEEMS HE FLEED' using Huffman code.

- 39
- 112

Decompressing Huffman coded data

To decompress Huffman code data, one can use a Huffman tree and trace the branches for each bit, starting at the root. When the final node of the branch is reached, the result has been found. The process continues until the entire item is decompressed.

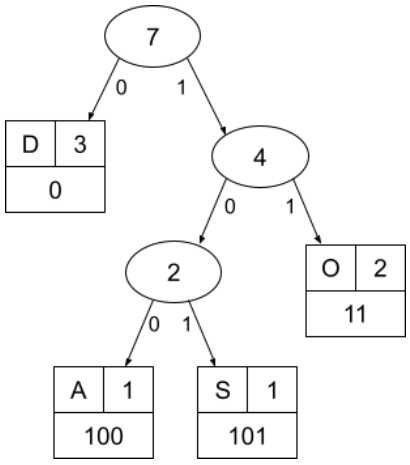
PARTICIPATION ACTIVITY 15.2.7: Decompressing Huffman code.

Animation captions:

1. The Huffman code is decompressed by first starting at the root. The branches are followed for each bit.
2. When the final node of the branch is reached, the result has been found.
3. Once the final node is reached decoding restarts at the root node.
4. The process continues until the entire item is decompressed.

PARTICIPATION ACTIVITY 15.2.8: Decompressing Huffman code.

Use the tree below to decompress 0111101000101.



1) What is the first decoded character?

Check Show answer

2) What is the second decoded character?

Check Show answer

3) 11 yields the third character O.
0 yields the fourth character D.

What is the next decoded character?

Check Show answer

4) What is the decoded text?

Check Show answer

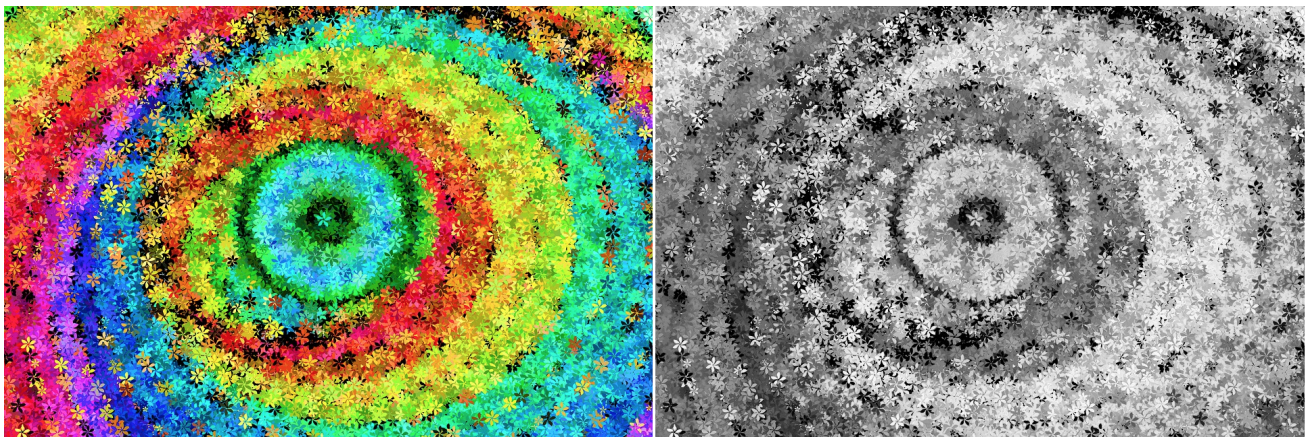
Text files, images, and videos

When compressing text files, an additional compression trick is used, wherein common sequences are treated as a pattern. Ex: "the box there is the right box" has the pattern "the" appear three times, and "box" twice. Thus, the dictionary may define a code for that pattern, so a code like 0010 might be listed not just as a letter like 't' but as a string like "the". Such a technique may be found in compression used to make ZIP files, for example.

Images may occupy much storage. Ex: An image with 1 million pixels and 3 bytes per pixel (for red, green, blue), may require 3 MB of storage. Each pixel is a number from 0 to 255. Some colors are much more common, like white (255, 255, 255) and black (0, 0, 0), so some numbers are much more common than others. Huffman coding is part of the common image compression technique known as JPEG. Image compression uses other techniques as well, which may lose some information (rounding, and discrete cosine transform, not discussed here) to achieve even greater compression.

Figure 15.2.2: Uncompressed image vs. a compressed image and color image vs. black and white image.

Source: zyBooks



Source: Pixabay

Video is a series of images known as frames. Thus, video compression techniques like MPEG include Huffman coding as well. Video compression also uses another compression technique: Because successive frames have only small differences, after an image, several successive frames may be represented just by the differences from the previous frame.

PARTICIPATION ACTIVITY 15.2.9: Text files, image, and video compression.

1) When compressing text, a dictionary may produce a code for a single character or a string.

   - True
   - False

2) A compressed image takes up the same amount of storage as an uncompressed image.

- ○ True
- ○ False

3) A common video compression technique involves only changing components that are different from the previous video frame.

- ○ True
- ○ False

Exploring further:

- Huffman coding (Wikipedia)
- Huffman tree generator (huffman.ooz.ie)
- Comparison of Huffman code and ASCII code (cyrptool-online.org)

©zyBooks 07/17/23 16:58 1692462
Taylor Larrechea
COLORADOCSPB2270Summer2023