computer science
illuminated

Compression

Nell Dale & John Lewis
(adaptation by Michael Goldwasser)

Compression

- **Data compression**–reducing the amount of space needed to store a piece of data.
- **Compression ratio**–is the size of the compressed data divided by the size of the original data.
- A data compression technique can be **lossless**, which means the data can be retrieved without losing any of the original information. Or it can be **lossy**, in which case some information is lost in the process of compaction.

Text Compression

- It is important that we find ways to store text efficiently and transmit text efficiently:
  - keyword encoding
  - run-length encoding
  - Huffman encoding

Keyword Encoding

- Frequently used words are replaced with a single character. For example:

Word	Symbol
as	^
the	~
and	+
that	$
must	&
well	%
those	#

Keyword Encoding *(Cont'd)*

- The following paragraph:
  - The human body is composed of many independent systems, such as the circulatory system, the respiratory system, and the reproductive system. Not only must all systems work independently, they must interact and cooperate as well. Overall health is a function of the well-being of separate systems, as well as how these separate systems work in concert.

Keyword Encoding *(Cont'd)*

- The encoded paragraph is:
  - The human body is composed of many independent systems, such ^ ~ circulatory system, ~ respiratory system, + ~ reproductive system. Not only & each system work independently, they & interact + cooperate ^ %. Overall health is a function of ~ %- being of separate systems, ^ % ^ how # separate systems work in concert.

Keyword Encoding *(Cont'd)*

- Thee are a total of 349 characters in the original paragraph including spaces and punctuation. The encoded paragraph contains 314 characters, resulting in a savings of 35 characters. The compression ratio for this example is 314/349 or approximately 0.9.
- **The characters we use to encode cannot be part of the original text.**

Run-Length Encoding

- A single character may be repeated over and over again in a long sequence. This type of repetition doesn't generally take place in English text, but often occurs in large data streams.
- In run-length encoding, a sequence of repeated characters is replaced by a *flag character*, followed by the repeated character, followed by a single digit that indicates how many times the character is repeated.

Run-Length Encoding *(Cont'd)*

- AAAAAAA would be encoded as: *A7
- *n5*x9ccc*h6 some other text *k8eee would be decoded into the following original text:

  nnnnnxxxxxxxxxccchhhhhh some other text kkkkkkkkeee
- The original text contains 51 characters, and the encoded string contains 35 characters, giving us a compression ratio in this example of 35/51 or approximately 0.68.
- Not really that good of a method for text (though often quite useful for images!)

Huffman Encoding

- Why should the character “X”, which is seldom used in text, take up the same number of bits as the blank, which is used very frequently?
- Huffman codes using **variable-length bit strings** to represent each character.
- A few characters may be represented by five bits, and another few by six bits, and yet another few by seven bits, and so forth. Others characters may use two bits.

Morse Code

The morse code consists of groups of dots and dashes, each group represents a letter or a number. You can communicate by Morse Code by flashing a light, by sound or by using a flag. The dots should be made as short as possible, and the dashes should be three times as long as the dots.

The Morse Alphabet

A for Alfa	·—	N for November	—·
B for Bravo	—···	O for Oscar	———
C for Charlie	—·—·	P for Papa	·——·
D for Delta	—··	Q for Quebec	——·—
E for Echo	·	R for Romeo	·—·
F for Foxtrot	··—·	S for Sierra	···
G for Golf	——·	T for Tango	—
H for Hotel	····	U for Uniform	··—
I for India	··	V for Victor	···—
J for Juliet	·———	W for Whiskey	·——
K for Kilo	—·—	X for X-ray	—··—
L for Lima	·—··	Y for Yankee	—·——
M for Mike	——	Z for Zulu	——··

Huffman Encoding *(Cont'd)*

- For example

Huffman Code	Character
00	A
01	E
100	L
110	O
111	R
1010	B
1011	D

Huffman Encoding *(Cont'd)*

- DOORBELL would be encode in binary as: 10111101101111010011100100.

- If we used a fixed-size bit string to represent each character (say, 8 bits), then the binary from of the original string would be 64 bits. The Huffman encoding for that string is 25 bits long, giving a compression ratio of 25/64, or approximately 0.39.

Prefix-Free Encoding

- An important characteristic of any Huffman encoding is that no bit string used to represent a character is the prefix of any other bit string used to represent a character.

Why?

Hypothetical Code:

A: 01
B: 011
C: 110
D: 10

How do you decode

0 1 1 1 0

0 1 1 1 0

0 1 1 1 0

Constructing a Huffman Encoding

Though not discussed in the text, there is a nice method for generating a Huffman encoding, given known character frequencies.

Interestingly, one can prove that this method produces the minimal length message for any such encoding with the given frequencies.

Constructing a Huffman Encoding

We will create an encoding “tree” as follows:

1) Initially create a “group” for each individual character, labeled with the frequency.

2) Find the two groups with the smallest lables. Merge them to form one larger group, labeled with the combined frequency.

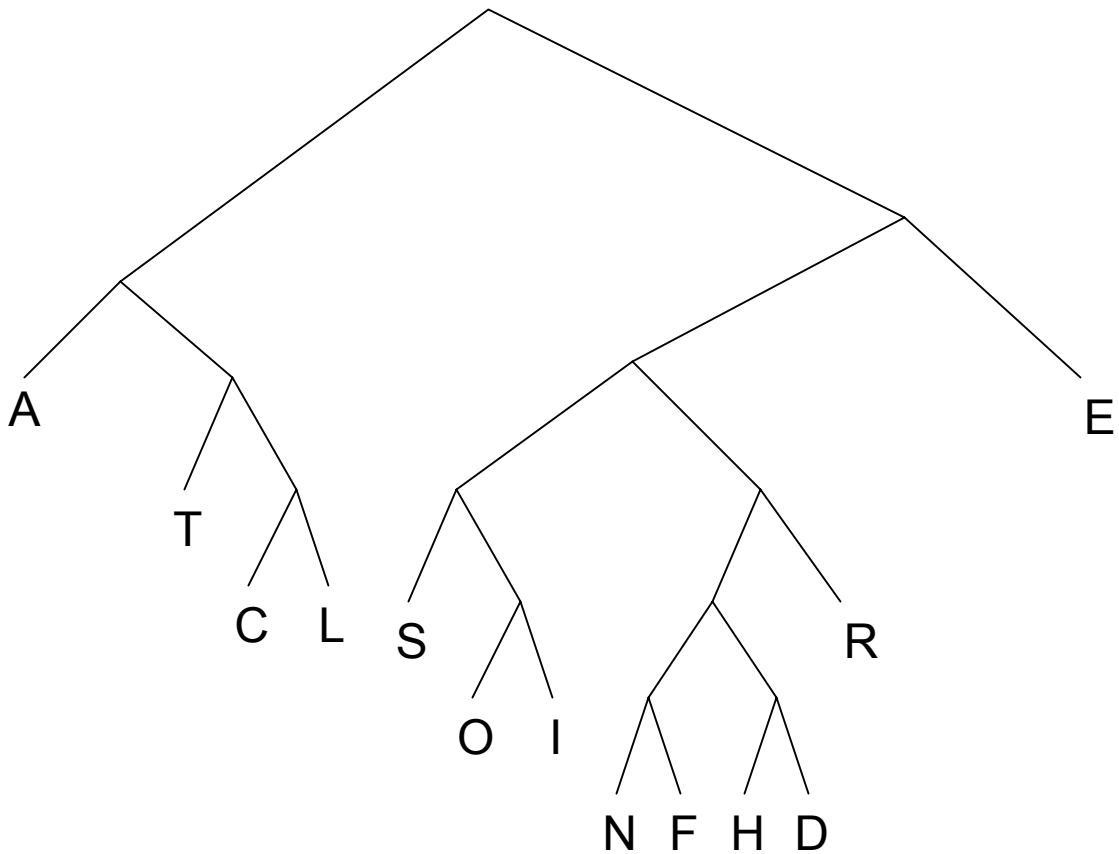
3) Repeat step 2 until only one group exists

An Example

(done in class)

char	freq
A	30
E	28
T	16
C	9
L	8
S	7
R	6
O	4
I	4
N	2
F	2
H	2
D	1

Reading the code from the tree



char	code
A	00
E	11
T	010
C	0110
L	0111
S	1000
R	1011
O	10010
I	10011
N	101000
F	101001
H	101010
D	101011

Compression Ratio

- Fixed-length code would require 4 bits per character.
  (we need 13 distinct patterns)
  Overall: 4x109 = 436 bits

- Huffman encoding would use
  30x2 + 28x2 + 16x3 + 9x4 + …
  = 366 bits.

- Compression ratio is:
  366/436 = 0.839

char	freq	code
A	30	00
E	28	11
T	16	010
C	9	0110
L	8	0111
S	7	1000
R	6	1011
O	4	10010
I	4	10011
N	2	101000
F	2	101001
H	2	101010
D	1	101011

Audio Formats

- Several popular formats are: WAV, AU, AIFF, VQF, and MP3. Currently, the dominant format for compressing audio data is MP3.
- MP3 is short for MPEG-2, audio layer 3 file.
- MP3 employs both lossy and lossless compression. First it analyzes the frequency spread and compares it to mathematical models of human psychoacoustics (the study of the interrelation between the ear and the brain), then it discards information that can't be heard by humans. Then the bit stream is compressed using a form of Huffman encoding to achieve additional compression.

Representing Video

- A video codec COmpressor/DECompressor refers to the methods used to shrink the size of a movie to allow it to be played on a computer or over a network. Almost all video codecs use lossy compression to minimize the huge amounts of data associated with video.

Representing Video *(Cont'd)*

- Two types of compression: temporal and spatial.
- **Temporal compression** looks for differences between consecutive frames. If most of an image in two frames hasn't changed, why should we waste space to duplicate all of the similar information?
- **Spatial compression** removes redundant information within a frame. This problem is essentially the same as that faced when compressing still images.