What You’ll Learn Today

– Review: how ASCII works and the great unfairness of bits
– What are the main strategies for compressing data?
– Examples of compression algorithms
Data Compression
(saving space)

Data compression
Reduction in the amount of space needed to store a piece of data.

Data compression techniques can be:
- lossless, which means the data can be retrieved without any loss of the original information
- lossy, which means some information may be lost in the process of compaction

Example: Text Compression

Problem: Unicode assigns 16 bits to each character in a document; uses a heck of a lot of space.

We need ways to store and transmit text efficiently. Why?

Common lossless compression techniques:
- keyword encoding
- run-length encoding
- Huffman encoding
Keyword Encoding

Replace frequently used words with a single character

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>as</td>
<td>^</td>
</tr>
<tr>
<td>the</td>
<td>~</td>
</tr>
<tr>
<td>and</td>
<td>+</td>
</tr>
<tr>
<td>that</td>
<td>$</td>
</tr>
<tr>
<td>must</td>
<td>&amp;</td>
</tr>
<tr>
<td>well</td>
<td>%</td>
</tr>
<tr>
<td>these</td>
<td>#</td>
</tr>
</tbody>
</table>

Keyword Encoding Example

Given the following paragraph,
We hold these truths to be self-evident, that all men are created equal, that they are endowed by their Creator with certain unalienable Rights, that among these are Life, Liberty and the pursuit of Happiness. That to secure these rights, Governments are instituted among Men, deriving their just powers from the consent of the governed, That whenever any Form of Government becomes destructive of these ends, it is the Right of the People to alter or to abolish it, and to institute new Government, laying its foundation on such principles and organizing its powers in such form, as to them shall seem most likely to effect their Safety and Happiness.
Keyword Encoding Example

The encoded paragraph is
We hold # truths to be self-evident, $ all men are created equal, $ ~y are endowed by ~ir Creator with certain unalienable Rights, $ among # are Life, Liberty + ~ pursuit of Happiness. — $ to secure # rights, Governments are instituted among Men, deriving ~ir just powers from ~ consent of ~ governed, — $ whenever any Form of Government becomes destructive of # ends, it is ~ Right of ~ People to alter or to abolish it, + to institute new Government, laying its foundation on such principles + organizing its powers in such form, ^ to ~m shall seem most likely to effect ~ir Safety + Happiness.

Keyword Encoding

Compression ratio
The size of the compressed data divided by the size of the original data (0 < c.r. <= 1)

What did we save?
Original paragraph: 656 characters
Encoded paragraph: 596 characters
Characters saved: 60 characters
Compression ratio: 596/656 = 0.9085

Could we use this substitution chart for all text?
Run-Length Encoding

Consider a single character which is repeated over and over again in a long sequence. Replace a repeated sequence with
- a flag character
- repeated character
- number of repetitions
Example: *n8
- * is the flag character
- n is the repeated character
- 8 is the number of times n is repeated

Original text
```
bbbbbbbbjjjkllqqqqqq
+++++
```
Encoded text
```
*b8jjkl*q6++5 (Why isn't l encoded? J?)
```
The compression ratio is 15/25 or .6

Original text
```
xxxxppplkkkkkk
```
Encoded text
```
*x4*p4l*k7
```

This type of repetition isn't very helpful for English text; can you think of a situation where it might be helpful?
Huffman Encoding

Why should the character “X" and "z" take up the same number of bits as "e" or " "?  

Huffman codes use variable-length bit strings to represent each character.

More frequently used letters have shorter strings to represent them.

Huffman Encoding Example

ballboard would be
101001001001010110001111011

Encoded is 28 bits vs 144 bits with Unicode; The compression ratio is 28/144 or 0.39

Try to encode roadbed
Huffman Encoding

Prefix Property
No character's bit string is the prefix of any other character's bit string.

To decode
look for match left to right, bit by bit
record letter when a match is found
begin next character where you left off

Huffman Encoding Example

Try it!

<table>
<thead>
<tr>
<th>Huffman Code</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>A</td>
</tr>
<tr>
<td>01</td>
<td>E</td>
</tr>
<tr>
<td>100</td>
<td>L</td>
</tr>
<tr>
<td>110</td>
<td>O</td>
</tr>
<tr>
<td>111</td>
<td>R</td>
</tr>
<tr>
<td>1010</td>
<td>B</td>
</tr>
<tr>
<td>1011</td>
<td>D</td>
</tr>
</tbody>
</table>

Decode
1011111001010
Huffman Encoding

The technique for creating codes guarantees the prefix property of the codes.

There is no single “Huffman code” -- each depends on the application. Two types of Huffman codes:

- general, based on use of letters in English, Spanish, …
- specialized, based on text itself or specific types of text

Lossless Compression Strategies

Recall:

- Keyword encoding
- Run-length encoding
- Huffman encoding

Next week:

How can we apply these to images?
What You Learned Today

– Compression strategies
  • Lossless compression
  • Lossy compression
– Compression algorithms:
  • Keyword encoding
  • Run-length encoding
  • Huffman encoding

Student To Dos

– HW05 due Wednesday 2/23
– Readings:
  • Wong ch 1 pp 13-19, ch 2, pp 26-44 (next week)

– Monday is a Holiday - NO CLASS
  • Tuesday will be a MONDAY Schedule