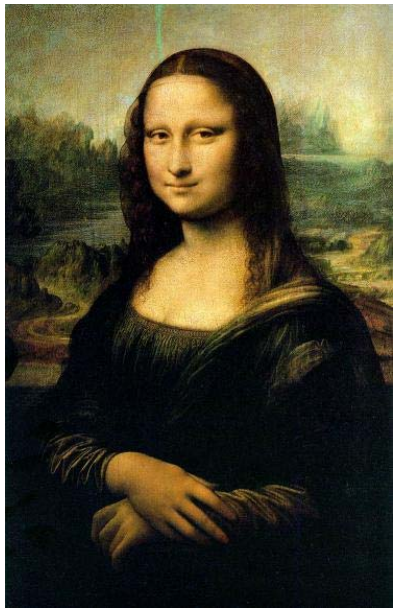
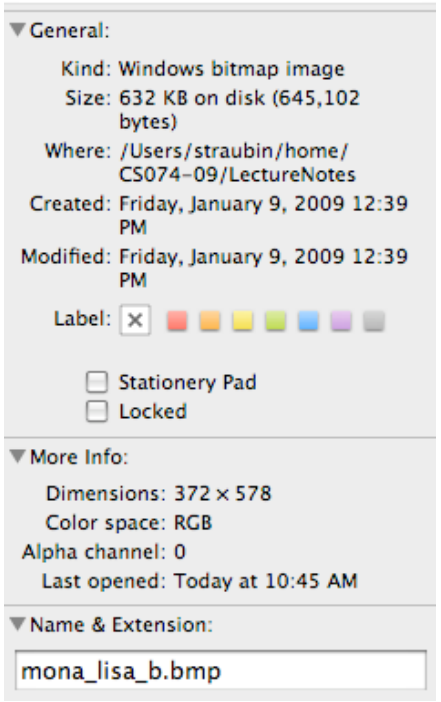
Compressing Information

The Digital World

An Experiment

- BMP file 372x578 pixels

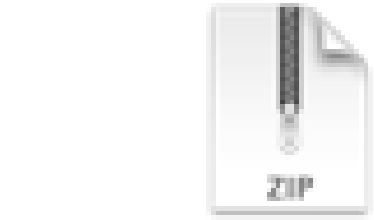




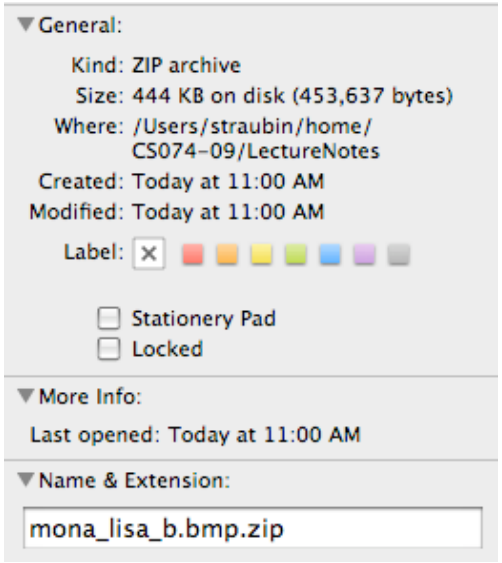
- The file size is about 3x(number of pixels)

An Experiment

- We can compress this using zip.



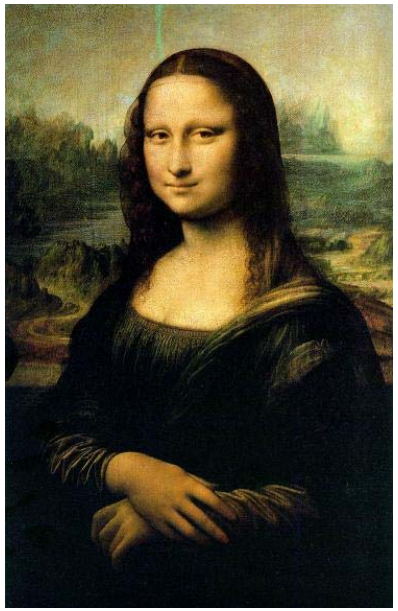
mona_lisa_b.bmp.zip

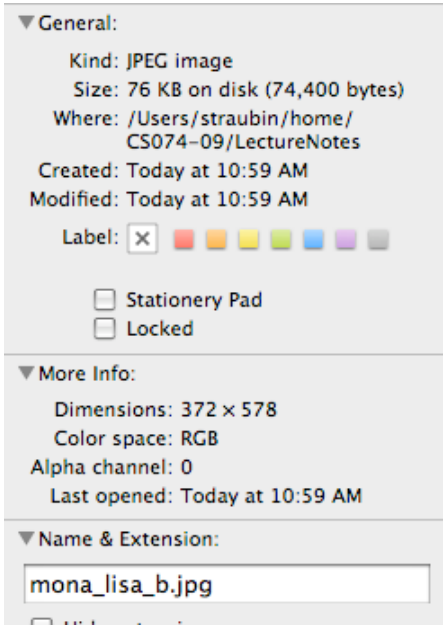


- The file size is 453/645=70% of the original.

An Experiment

- Alternatively, we can compress using JPEG

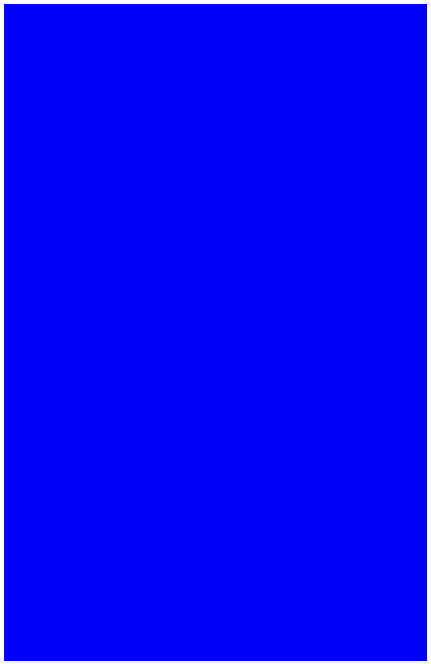


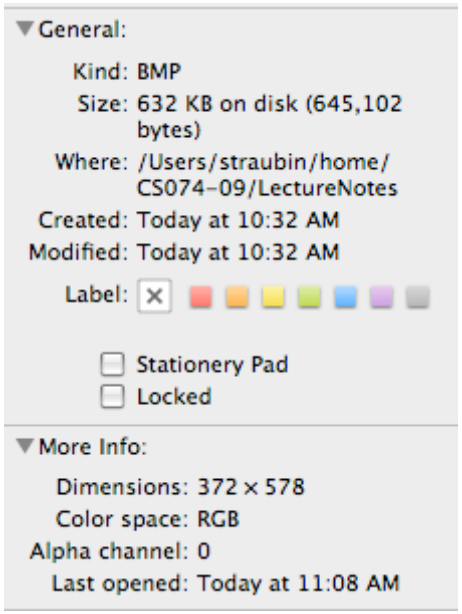


- The compression ratio is now 74/645=11.5%

An Experiment

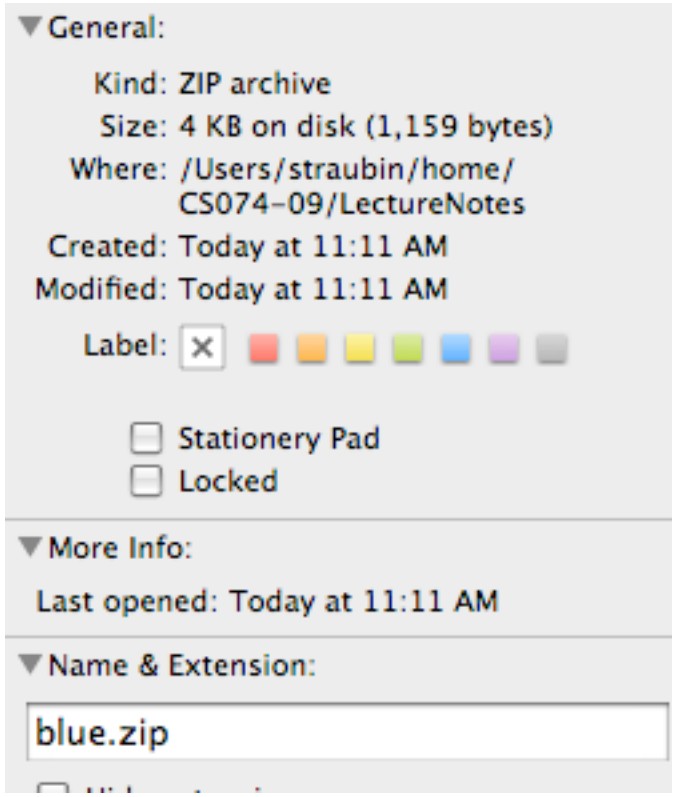
- What if we tried the zip compression with this image?





An Experiment

- It shrinks to almost nothing!



What accounts for the differences between all these compression methods?

Lossless versus Lossy Compression

- **Lossless:** Every bit of the original file can be recovered from the compressed version. Used for text, software. ZIP is a lossless compression method.
- **Lossy:** Some information is irretrievably lost, but the compressed version should look or sound (almost) as good as the original. Used for audio, graphics, video. JPEG and MP3 are lossy compression methods.

Approaches to Lossless Compression

- Morse Code (devised for 19th Century telegraph)

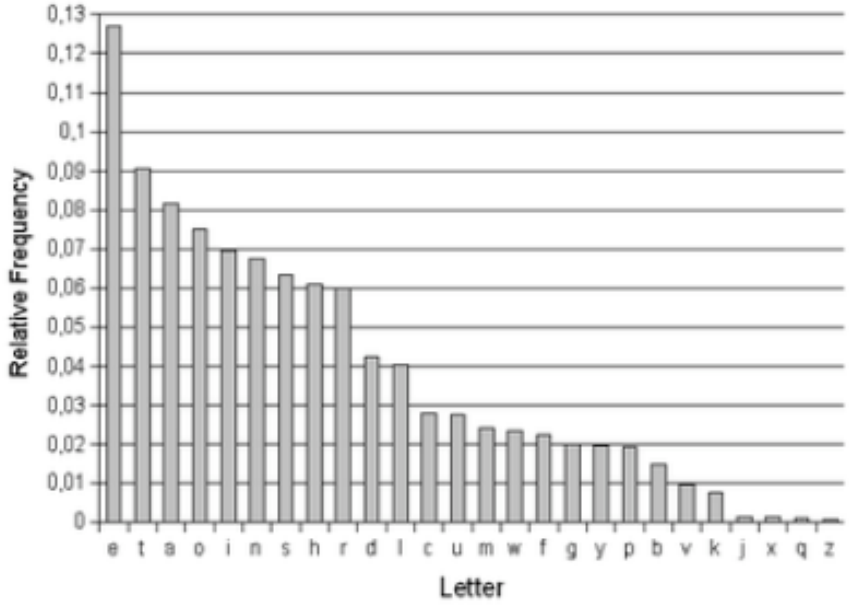
A . –	J . – – –	S . . .	2 . . – – –
B – . . .	K – . –	T –	3 . . . – –
C – . – .	L . – . .	U . . –	4 . . . . –
D – . .	M – –	V . . . –	5 . . . . .
E .	N – .	W . – –	6 – . . . .
F . . – .	O – – –	X – . . –	7 – – . . .
G – – .	P . – – .	Y – . – –	8 – – – . .
H	Q – – . –	Z – – . .	9 – – – – .
I . .	R . – .	1 . – – – –	0 – – – – –

Approaches to Lossless Compression

- Morse Code (devised for 19th Century telegraph)
- *Variable-length Encoding:* More frequently-occurring letters have shorter encodings.

Approaches to Lossless Compression

- Variable-length encoding works because of highly skewed distribution of letter frequencies in English.



Approaches to Lossless Compression

- *Dictionary Methods* compile a “dictionary” of frequently-occurring byte sequences in the file. Instead of recording the byte sequence itself, the compressed file contains an encoding of its position in the dictionary. (This is not a precise account of how such methods work, but it gives the idea.)

Approaches to Lossless Compression

- *Dictionary Methods* For example, the novel Bleak House by Charles Dickens contains 364,000 words and 1.87 million characters, counting the spaces, but only 15,000 distinct words.
- You could encode each word by a 14-bit string (<2bytes per word) and send the list of distinct words in ASCII. The resulting file would contain fewer than half the bytes of the original text in ASCII.
- (This analysis ignores distinction between upper and lower case, as well as punctuation.)

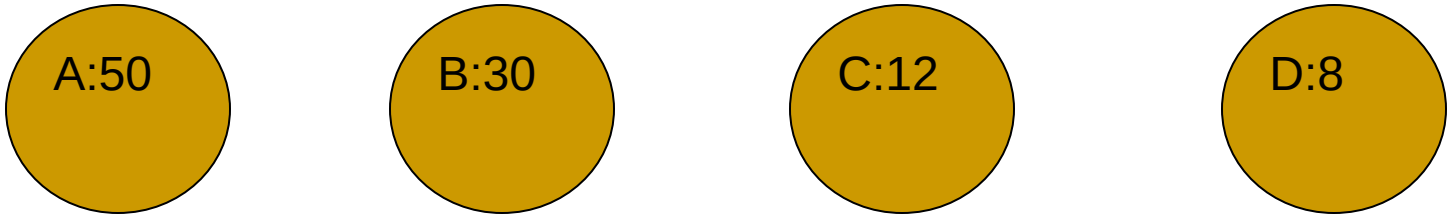
Approaches to Lossless Compression

- Methods in actual use can be applied to arbitrary files, not just text!

Huffman Coding-A Variable-Length Method

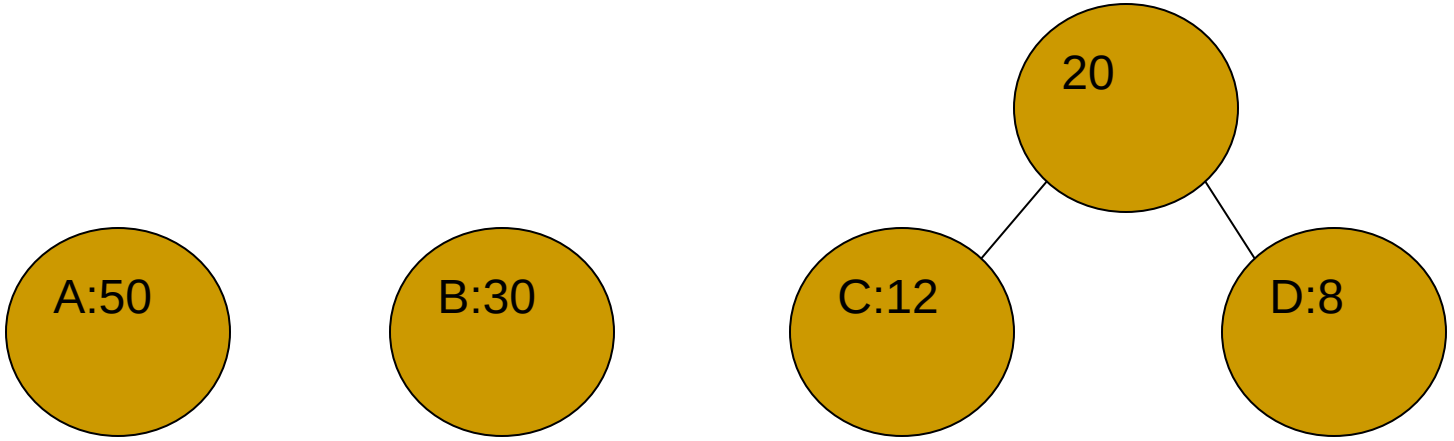
- Suppose we have a sequence of twenty-thousand a's, b's, c's and d's: **abbacaaddaaacbbbbacababababad…**
- If we encode each symbol by 2 bits the resulting file will have 40,000 bits = 5000 bytes.
- What if the distribution of symbols was not uniform? For instance, suppose 50% are a's, 30% b's, 12% c's and 8% d's.

Huffman Coding Creates a Tree

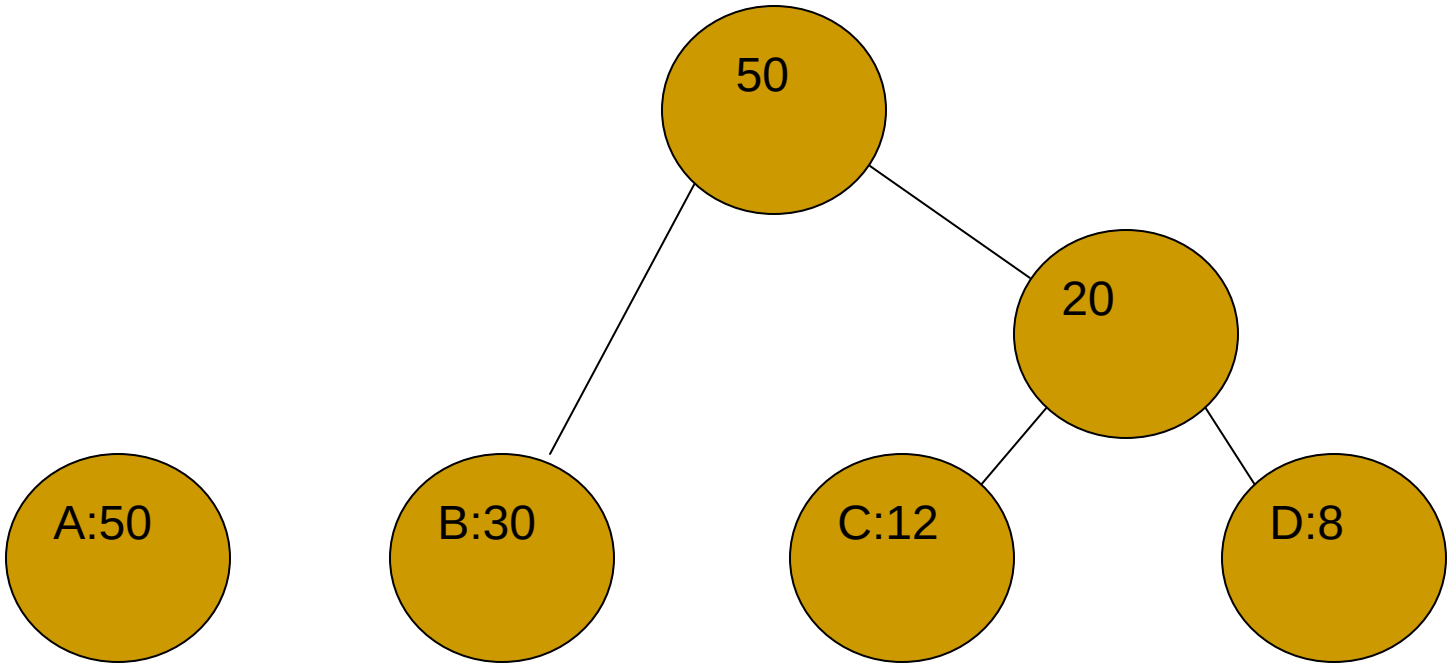


Huffman Coding Creates a Tree

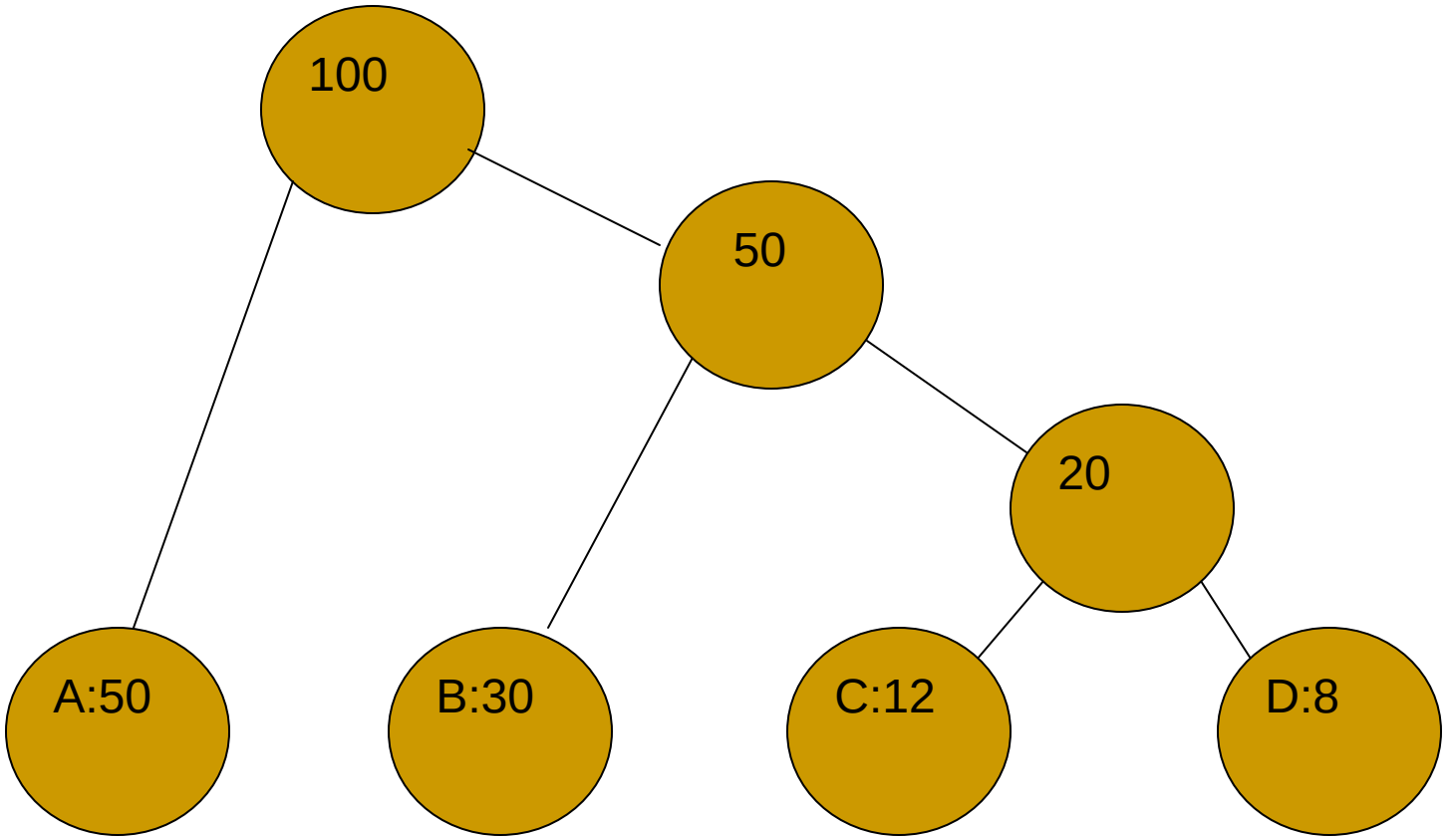
Repeatedly coalesce two nodes with smallest value into a single node.



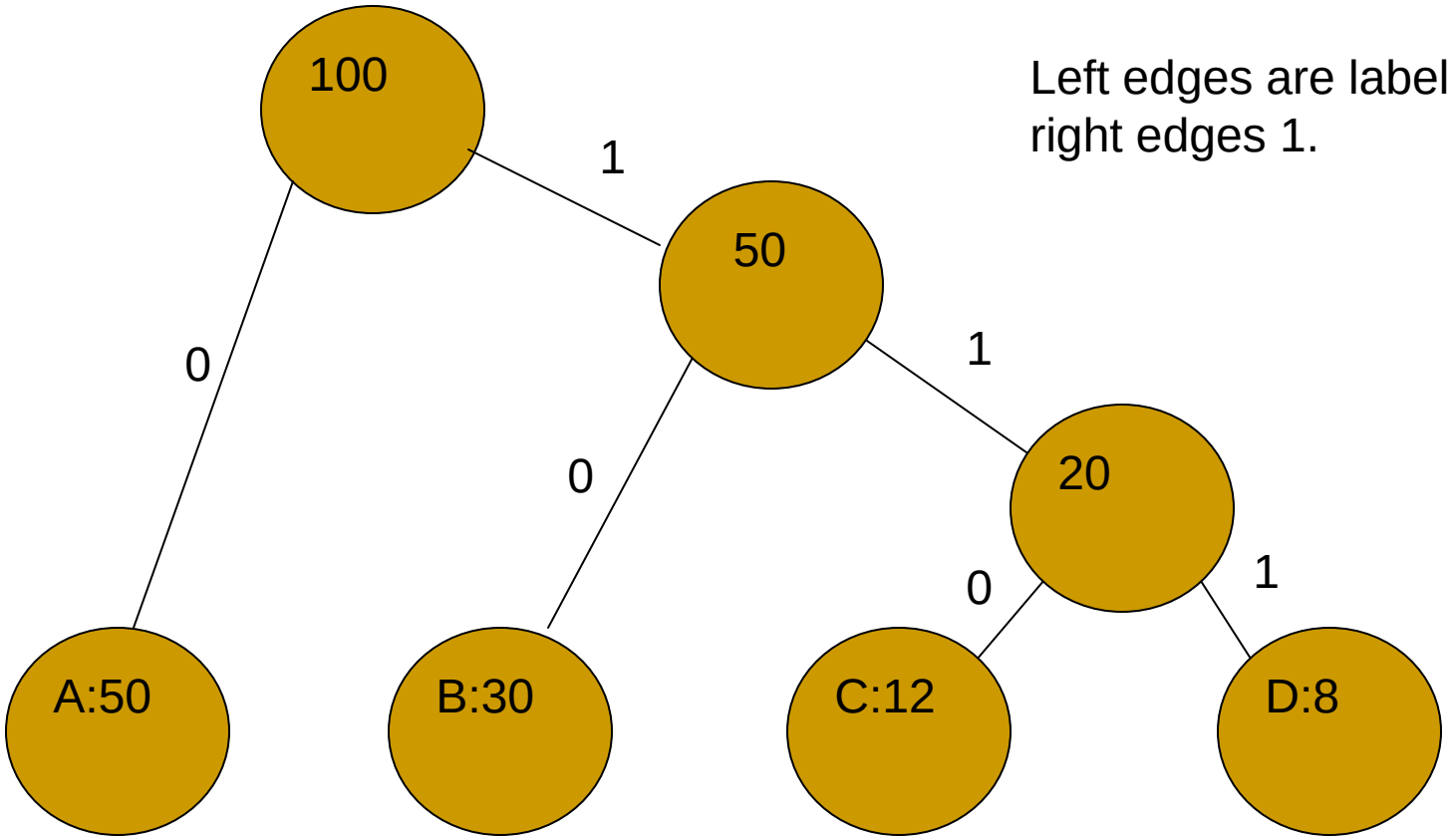
Huffman Coding Creates a Tree



Huffman Coding Creates a Tree



Huffman Coding Creates a Tree



Left edges are labeled 0, right edges 1.

Huffman Coding Creates a Tree

Resulting code is A:0,B:10,C:110,D:111

For instance, we would encode

abbacaaddaaacbbbbacababababad…

by

0101001100011111100011010101001100100…

Huffman Coding

- There is only one way to decipher a string of a's, b's, c's and d's that has been encoded this way.
- Our sample of 20,000 symbols contains 10,000 a's, 6,000 b's, and 4,000 c's and d's, so the string takes10000+2x6000+3x4000=34000 bits to encode, giving a compression ratio of 34/40=85%.

Results for real files

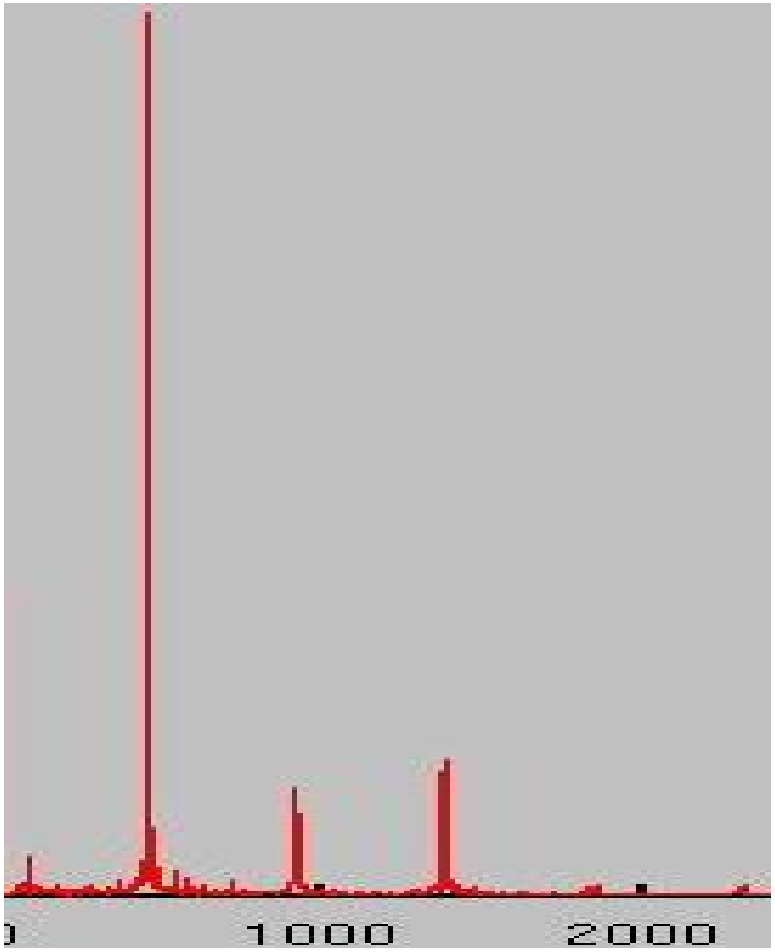
- A long ASCII text in English with 2.07 million bytes compressed to 1.193 million bytes, a compression ratio of 60%.
- ZIP, which is a dictionary-based method, compressed the same file to 768,000 bytes, a compression ratio of 38%.

Approaches to Lossy Compression

- Frequency analysis: The spectrum of a sound shows the strength of the sound signal at different frequencies.
- The original sound file can be recovered from the spectrum, if it is measured at a sufficient number of different frequencies.

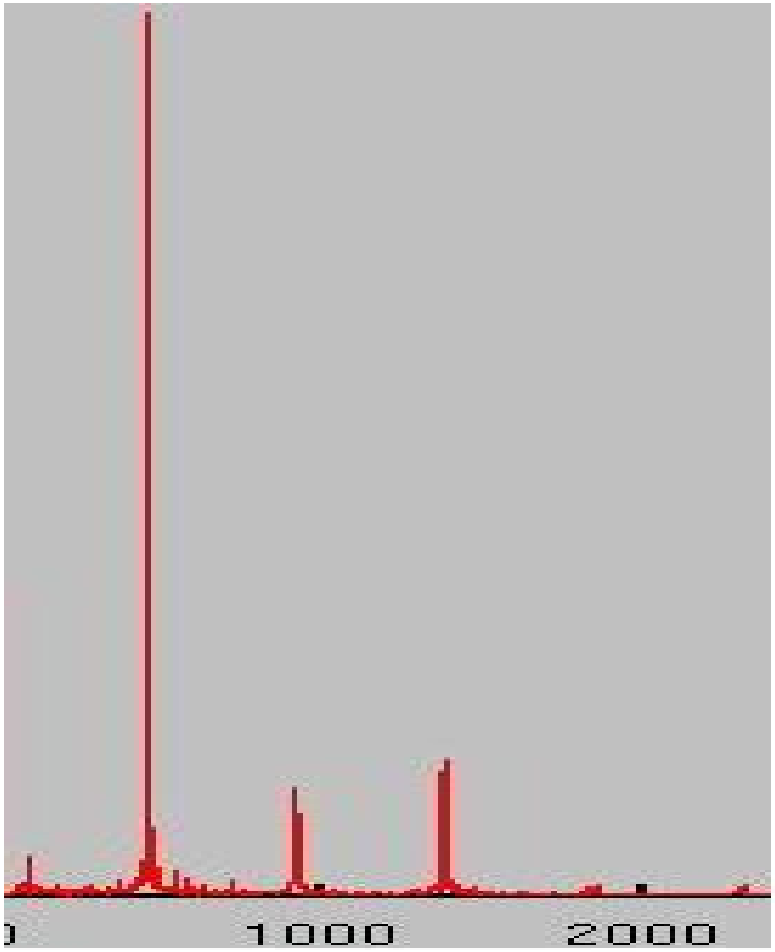
Frequency Spectrum of Flute Clip

- This is the spectrum fro a sequence of 8192 samples from the flute clip.
- There is a strong spike at about 500 hertz, lesser ones at about 1000 and 1500 hertz, and a large number of very small values.



Frequency Spectrum of Flute Clip

- The different frequency strengths can be encoded as integers.
- Very small values can be set to 0. Small integer values like 0,1,2 will predominate, so Huffman coding can then be used to compress these strengths.



Frequency Spectrum of Flute Clip

- Information is lost in approximating the signal strengths by integers.
- More sophisticated methods like MP3 use the fact that the ear is less sensitive at some frequencies, and use fewer bits to encode the signal strength at these frequencies.