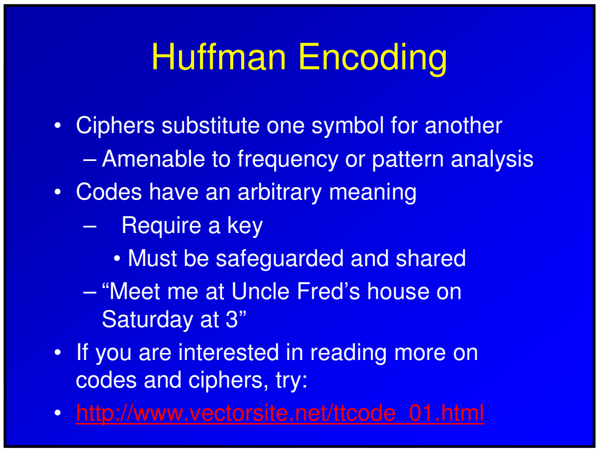
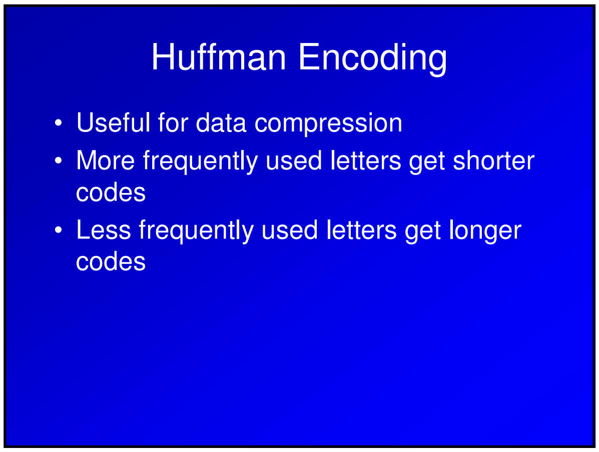


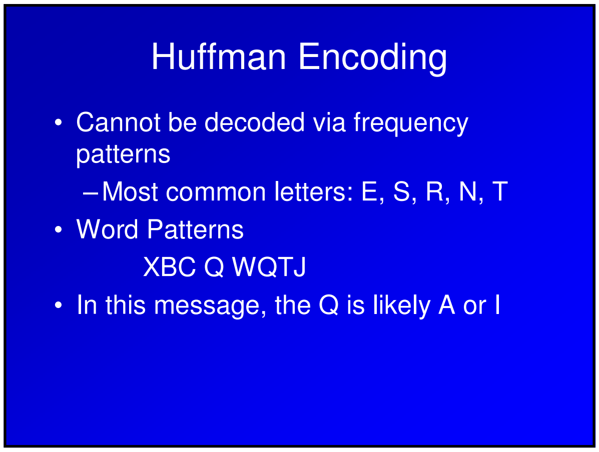
Huffman Encoding

- Ciphers substitute one symbol for another
  - Amenable to frequency or pattern analysis
- Codes have an arbitrary meaning
  - Require a key
    - Must be safeguarded and shared
  - "Meet me at Uncle Fred's house on Saturday at 3"
- If you are interested in reading more on codes and ciphers, try:
- http://www.vectorsite.net/ttcode_01.html



Huffman Encoding

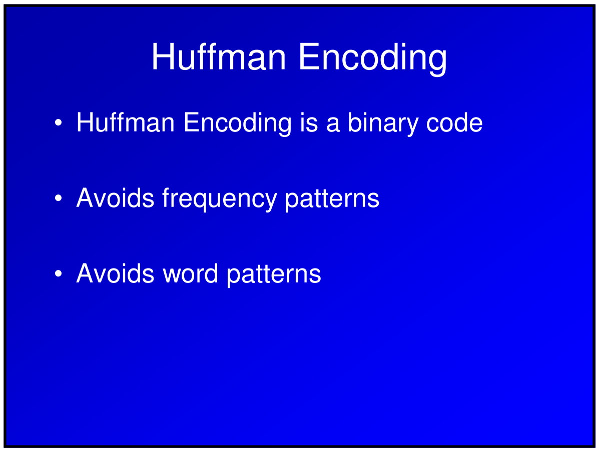
- Useful for data compression
- More frequently used letters get shorter codes
- Less frequently used letters get longer codes



Huffman Encoding

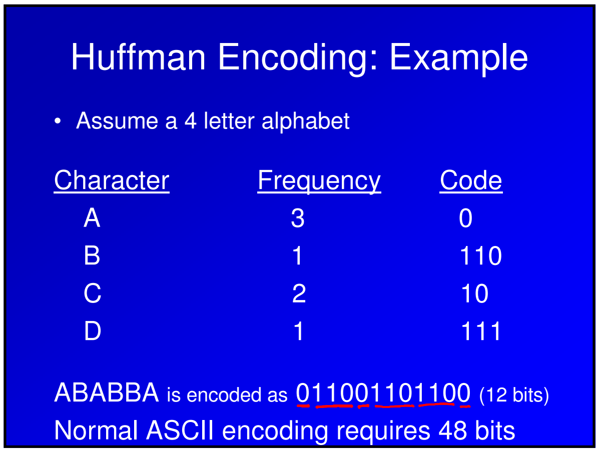
- Cannot be decoded via frequency patterns
  - Most common letters: E, S, R, N, T
- Word Patterns

  XBC Q WQTJ
- In this message, the Q is likely A or I



Huffman Encoding

- Huffman Encoding is a binary code
- Avoids frequency patterns
- Avoids word patterns



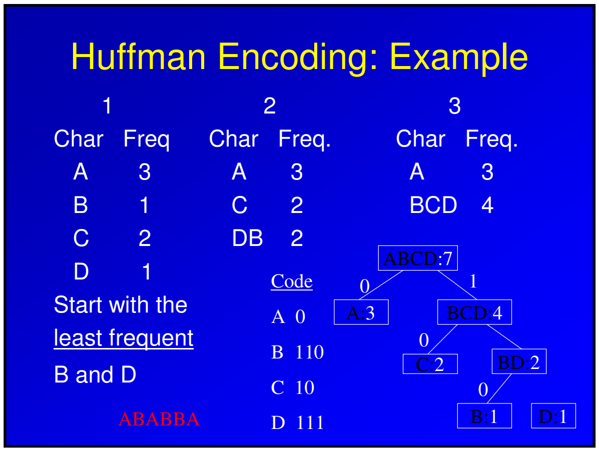
Huffman Encoding: Example

- Assume a 4 letter alphabet

Character	Frequency	Code
A	3	0
B	1	110
C	2	10
D	1	111

ABABBA is encoded as 011001101100 (12 bits)

Normal ASCII encoding requires 48 bits



Huffman Encoding: Example

1

Char	Freq
A	3
B	1
C	2
D	1

2

Char	Freq.
A	3
C	2
DB	2

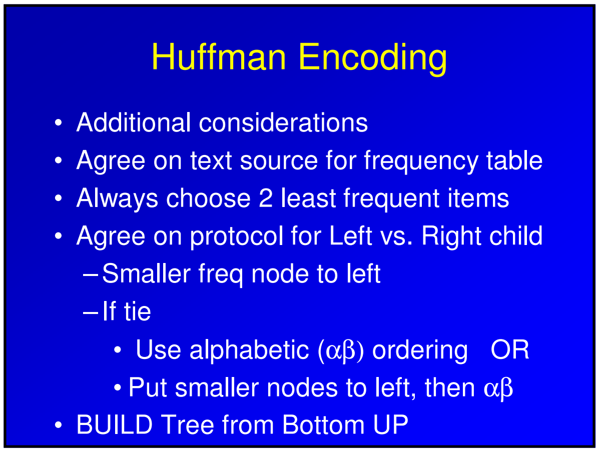
3

Char	Freq.
A	3
BCD	4

Start with the least frequent B and D

ABABBA

Code	
A	0
B	110
C	10
D	111



Huffman Encoding

- Additional considerations
- Agree on text source for frequency table
- Always choose 2 least frequent items
- Agree on protocol for Left vs. Right child
  - Smaller freq node to left
  - If tie
    - Use alphabetic (αβ) ordering OR
    - Put smaller nodes to left, then αβ
- BUILD Tree from Bottom UP