

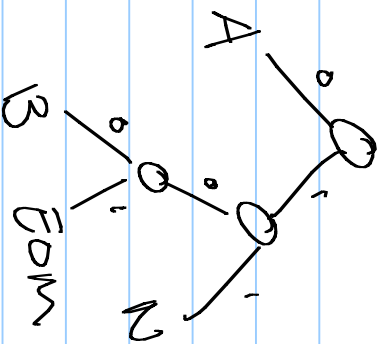
# Huffman Codes - Part 2

## Announcements

- Program due Monday (you are welcome to use Tuesday also)
- Tomorrow - Show up!
- Review - Monday in class

# Transmitting Information

(See current program!)



Send string:

100b116110101

↳ BANANA (EOM)

## Idea

We wish to transmit information using as few bits as possible.

Standard ASCII  $\rightarrow$  8 bits per character

How can we do better?

- Can figure out which characters we actually need.
- More common letters get shorter bitstrings.

First - what do we need?

This sentence contains three a's, three c's, two d's, twenty-six e's, five f's, three g's, eight h's, thirteen i's, two j's, sixteen n's, nine o's, six r's, twenty-seven s's, twenty-two t's, two u's, five v's, eight w's, four x's, five y's, and only one z.

Letters:

A, C, D, E, F, G, H, I, J, L, N, O, R, S, T, U,  
V, W, X, Y, Z

(Missing something... Spaces, Period, comma,  
apostrophe...)

# Frequency Analysis

A	C	D	E	F	G	H	I	L	N	O	R	S	T	U	V	W	X	Y	Z
3	3	2	26	5	3	8	13	2	16	9	6	27	22	2	5	8	4	5	1

Which ones do we want to use few bits for?

OR: which ones for?? Can we use many bits for??

Huffman's algorithm:

Make two least frequent characters

merge them into character & recurse

↓

A	C	D	E	F	G	H	I	L	N	O	R	S	T	U	V	W	X	Y	Z
3	3	2	26	5	3	8	13	2	16	9	6	27	22	2	5	8	4	5	1

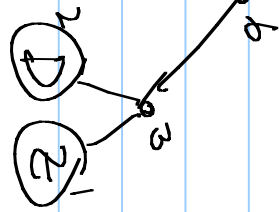
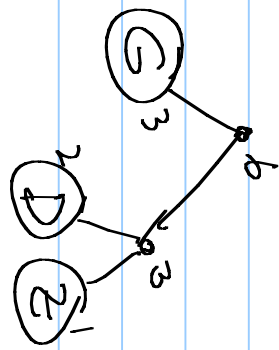
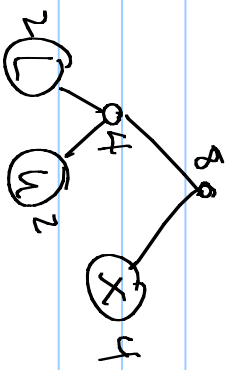
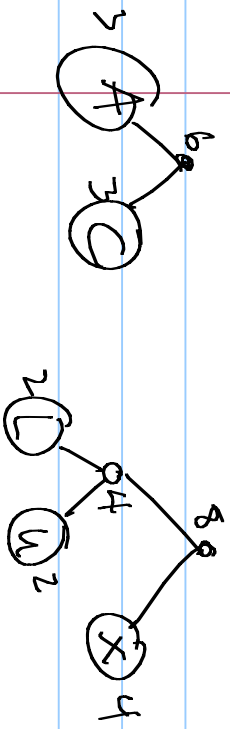
→

A	C	E	F	G	H	I	L	N	O	R	S	T	U	V	W	X	Y	Z
3	3	26	5	3	8	13	2	16	9	6	27	22	2	5	8	4	5	3

take two least frequent

frequency

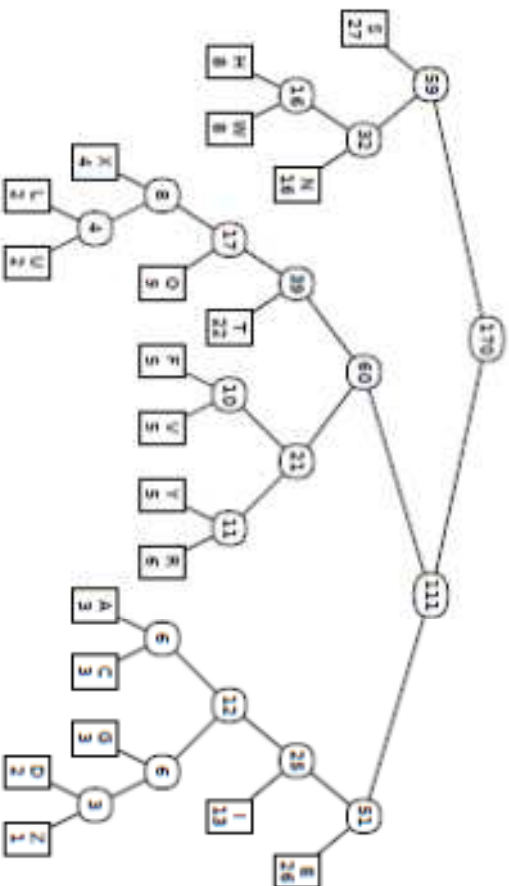
new "joined" character



In the end, this:

A	C	D	E	F	G	H	I	L	N	O	R	S	T	U	V	W	X	Y	Z
3	3	2	26	5	3	8	13	2	16	9	6	27	22	2	5	8	4	5	1

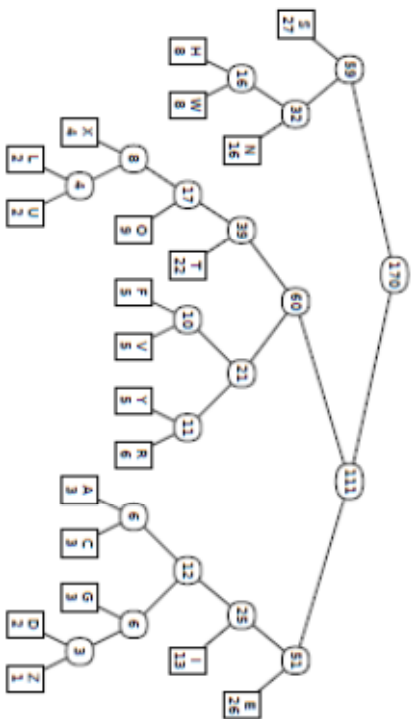
turns into a decoder tree (like in program)



to send an S,  
transmit 00,  
to send G: 110010

Original message:

1001 0100 1101 00 00 11 1 011 1001 111 011 110001 111 110001 10001 011 1001 110000 1101  
T H I S S E N T E M C E C O N T A I







Why do all this again?

170 letters

This sentence contains three a's, three c's, two d's, twenty-six e's, five f's, three g's, eight h's, thirteen i's, two l's, sixteen n's, nine o's, six r's, twenty-seven s's, twenty-two t's, two u's, five v's, eight w's, four x's, five y's, and only one z.

How does ASCII do?

$$\begin{array}{r} 170 \\ \times 8 \\ \hline 1360 \end{array}$$

Using our tree:

642 bits

(to send tree: 20 char, 8 bits each, 80 bits for tree)