Huffman codes

**Announcements**
- Program due *Saturday*
- Next program up - take a look over breaks
- Take out paper & pen
- Final is Monday the 13th
Transmitting information

prefix free

Is this a good way to transmit?

1000110110101 ← 15 bits

B ANANA EOM

ASCII: 8x6 = 48
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 bit-strings.

\(STJM\)

\(QZ\)

\(short\)

\(long\)
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 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.

Letters:

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

(ignoring punctuation, spaces... )
Frequency Analysis

\[
\begin{array}{cccccccccccccccc}
A & C & D & E & F & G & H & I & L & N & O & R & S & T & U & V & W & X & Y & Z \\
3 & 3 & 2 & 26 & 5 & 3 & 8 & 13 & 2 & 16 & 9 & 6 & 27 & 22 & 2 & 5 & 8 & 4 & 5 & 1
\end{array}
\]

Which ones do we want to use few bits for?

Which ones can use lots of bits?

\[B, D, A, C, G\]

C build bottom
Huffman's algorithm:
- Make two least frequent characters
- Merge them into character & recurse
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:

THISSSENTENCECONTAI...
Exercise: [Binary Code]

Message: HELLO

How many bits? 26
Why do all this again?

170 letters

How does ASCII do?

$170 \times 8 = 1360$

Using our tree:

642 bits

(to send tree: 20 chars, 8 bits each, 80 bits for tree)
Good to know:

Huffman trees are optimal. They use fewest possible bits for any message. (greedy)