

Slide 1

### Administrivia

- Homework 8 to be updated/finalized soon. Official due date still Friday, but will be accepted without penalty through sometime Monday.
- Final is May 7. I'll put a review sheet on the Web soon. A review session is very possible. We'll decide next class or Friday. We can also do some reviewing Wednesday. Think about what topics you'd like me to cover, and send me mail.
- Extra-credit problems possible if several people are interested. Would be due after final.

Slide 2

### Minute Essay From Last Lecture

- Question: Write a recursive function to count the number of nodes in a tree.
- Answer?

### Encoding — Fixed-Length Versus Prefix

Slide 3

- ASCII represents one way to represent text in binary form, 8 bits per character. Is there anything more compact? If we have fewer characters, could choose a different encoding, possibly fewer bits per character needed.
- Or — if some characters occur much more often than others, can we take advantage of that?  
Yes — in addition to “fixed-length codes”, there are “prefix codes”, in which no codeword (encoding of a character) is a prefix of another codeword.
- How is this related to trees? consider encoding/decoding. Encoding is easy for either scheme. Decoding for fixed-length code is also not too tough. Decoding for prefix code? Easy if we represent as tree. How to build such a tree?

### Huffman Encoding

Slide 4

- A way to generate a prefix code, starting with a table of characters and frequencies:  
Start with a “forest” of one-node trees.  
While there’s more than one tree in the forest, remove the two with the smallest frequencies, combine them, and put the result back in the forest.  
(Example.)
- Intuitively it would appear that this would do a good job. Can in fact prove that it’s optimal.

## Minute Essay

- Any topics you know you'd like me to review?

Slide 5