

“Stand-alone topics” outline:

- ▶ Foldl, and how to model mathematical functions (Monday)
- ▶ The Huffman encoding (**Today**)
- ▶ Review for the final exam (Friday)
- ▶ **Final exam (Wed, May 1, 1:30pm)**

Today:

- ▶ The idea of an encoding
- ▶ Full binary trees and encodings
- ▶ Building an optimal encoding
- ▶ Implementation

Final version of modeling project due Fri, Apr 26

# Encoding

ASCII/Unicode (last four bits):

|   |      |   |      |   |      |
|---|------|---|------|---|------|
| A | 0001 | F | 0110 | K | 1011 |
| B | 0010 | G | 0111 | L | 1100 |
| C | 0011 | H | 1000 | M | 1101 |
| D | 0100 | I | 1001 | N | 1110 |
| E | 0101 | J | 1010 | O | 1111 |

Sample encoding:

|      |      |      |      |      |      |
|------|------|------|------|------|------|
| 0001 | 1110 | 1110 | 1001 | 1011 | 0001 |
| A    | N    | N    | I    | K    | A    |

Message size:  $4 \times 6 = 24$  bits.

# Encoding

Variable-length codes (frequent letters are shorter):

|   |     |   |     |   |      |
|---|-----|---|-----|---|------|
| A | 0   | F | 100 | K | 01   |
| B | 10  | G | 101 | L | 0000 |
| C | 001 | H | 110 | M | 11   |
| D | 010 | I | 00  | N | 1    |
| E | 011 | J | 111 | O | 0001 |

Sample encoding:

|   |   |   |    |    |   |
|---|---|---|----|----|---|
| 0 | 1 | 1 | 00 | 01 | 0 |
| A | N | N | I  | K  | A |

Message size:  $1 + 1 + 1 + 2 + 2 + 1 = 8$  bits.

# Encoding

|   |     |   |     |   |      |
|---|-----|---|-----|---|------|
| A | 0   | F | 100 | K | 01   |
| B | 10  | G | 101 | L | 0000 |
| C | 001 | H | 110 | M | 11   |
| D | 010 | I | 00  | N | 1    |
| E | 011 | J | 111 | O | 0001 |

|   |   |   |    |    |   |
|---|---|---|----|----|---|
| 0 | 1 | 1 | 00 | 01 | 0 |
| A | N | N | I  | K  | A |

Or did you mean

|     |      |   |
|-----|------|---|
| 011 | 0001 | 0 |
| E   | O    | A |

# Encoding

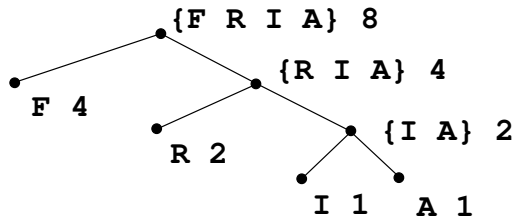
Prefix code

|   |     |   |     |   |     |
|---|-----|---|-----|---|-----|
| A | 0   | F | ... | K | 111 |
| B | ... | G | ... | L | ... |
| C | ... | H | ... | M | ... |
| D | ... | I | 110 | N | 10  |
| E | ... | J | ... | O | ... |

|   |    |    |     |     |   |
|---|----|----|-----|-----|---|
| 0 | 10 | 10 | 110 | 111 | 0 |
| A | N  | N  | I   | K   | A |

Message size:  $1 + 2 + 2 + 3 + 3 + 1 = 12$  bits.

# Trees



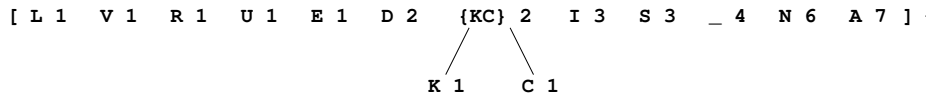
RIFFRAFF

|   |     |
|---|-----|
| A | 111 |
| F | 0   |
| I | 110 |
| R | 10  |

## Building the tree

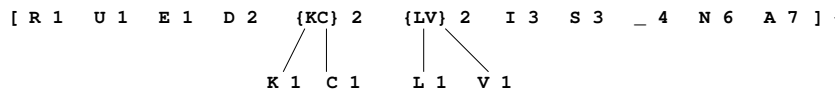
[ K 1   C 1   L 1   V 1   R 1   U 1   E 1   D 2   I 3   S 3   \_ 4   N 6   A 7 ] .

# Building the tree

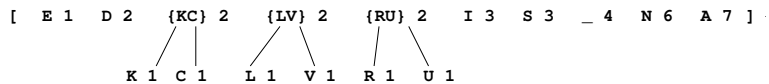




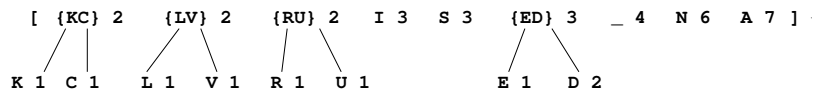
# Building the tree



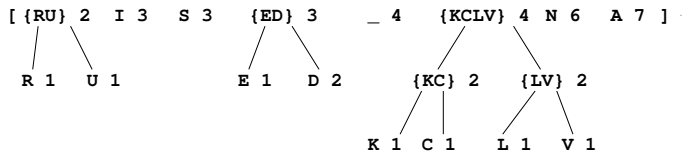
# Building the tree



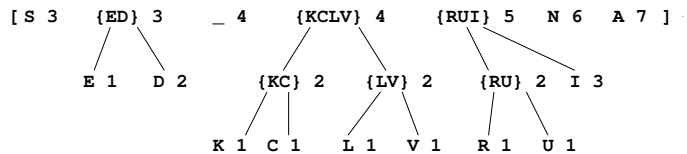
# Building the tree



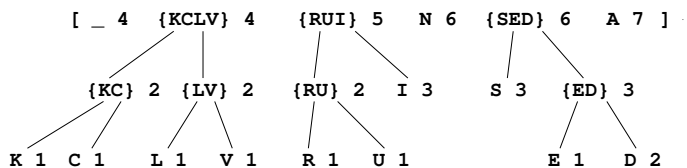
# Building the tree



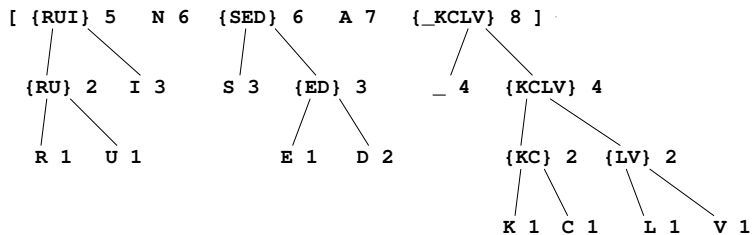
# Building the tree



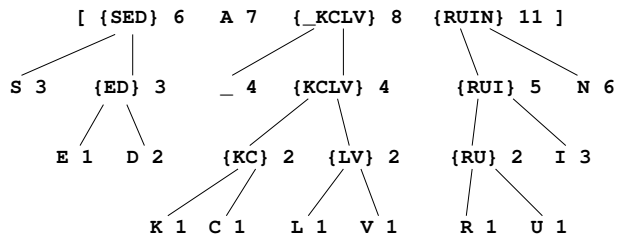
## Building the tree



## Building the tree

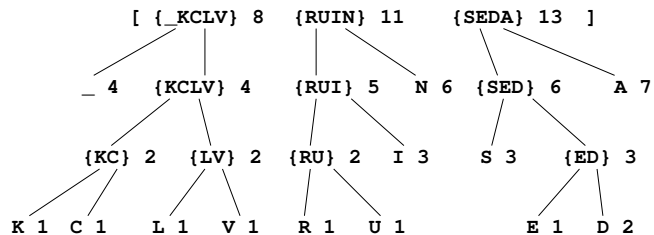


## Building the tree

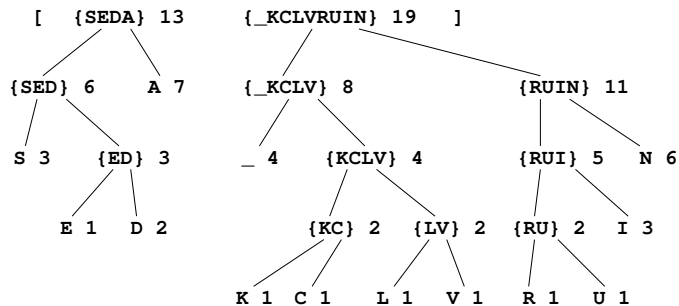




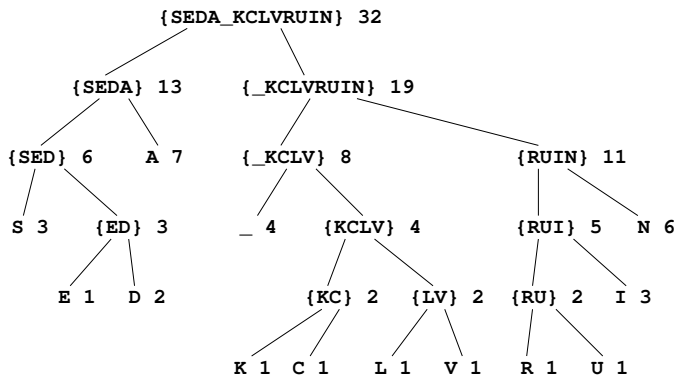
## Building the tree



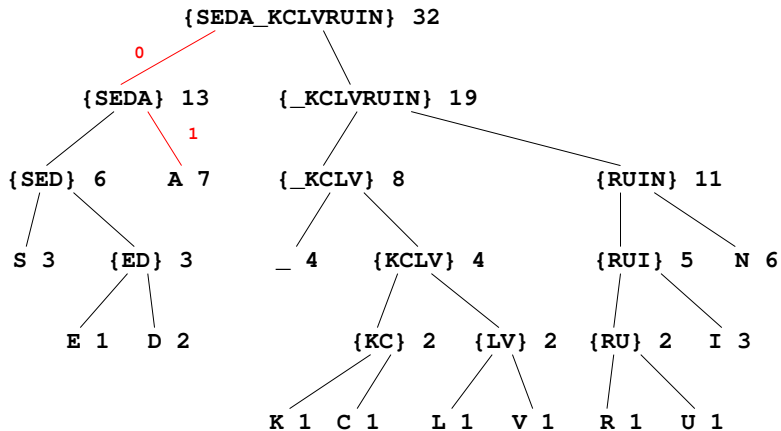
## Building the tree



## Building the tree

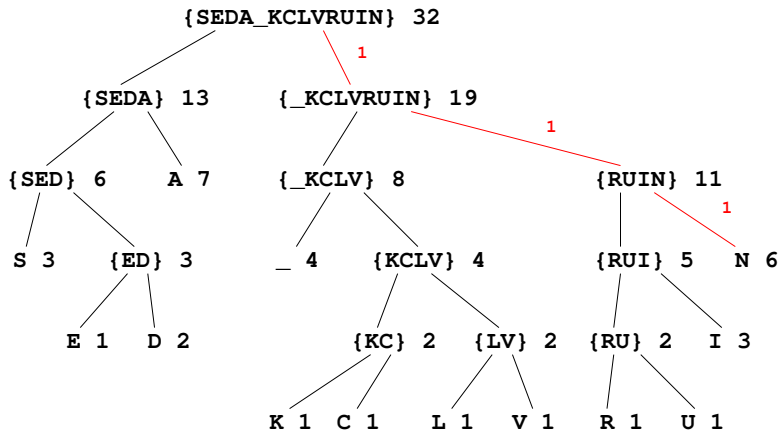


# Encoding the message



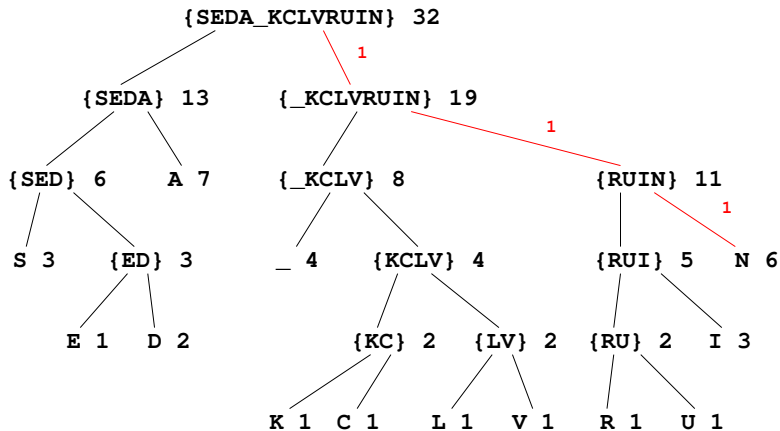
|    |
|----|
| 01 |
| A  |

# Encoding the message



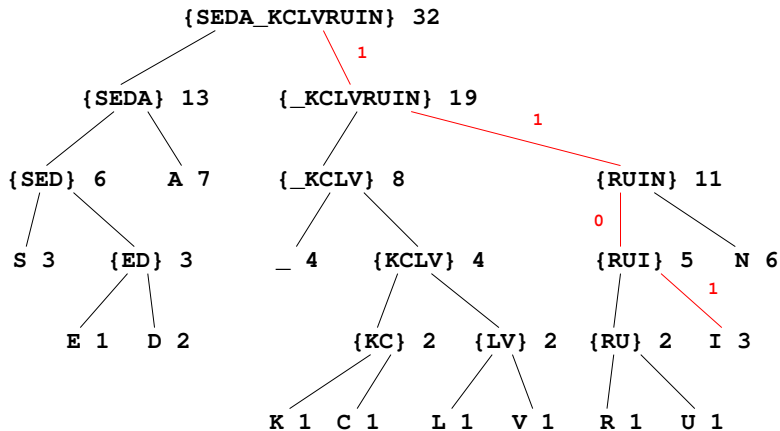
|    |     |
|----|-----|
| 01 | 111 |
| A  | N   |

# Encoding the message



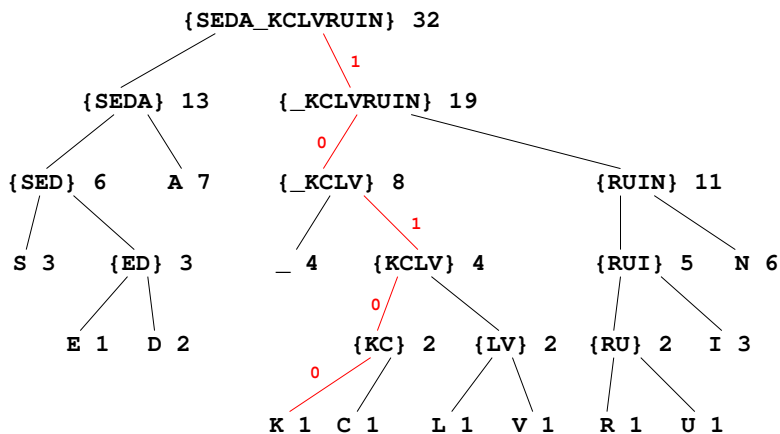
|    |     |     |
|----|-----|-----|
| 01 | 111 | 111 |
| A  | N   | N   |

# Encoding the message



|    |     |     |      |
|----|-----|-----|------|
| 01 | 111 | 111 | 1101 |
| A  | N   | N   | I    |

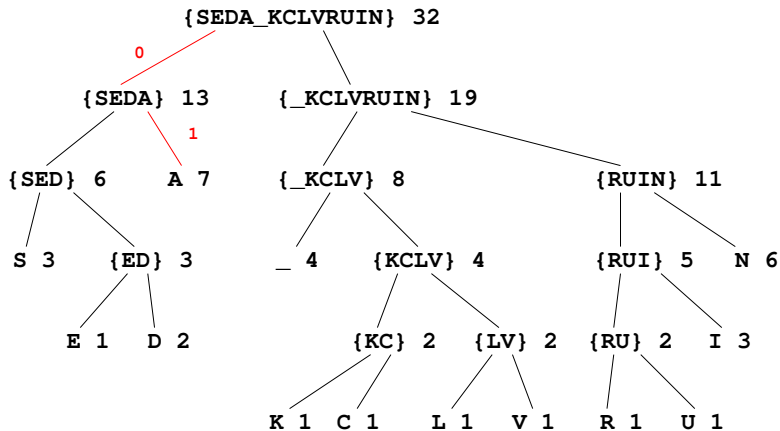
# Encoding the message



|    |     |     |      |       |
|----|-----|-----|------|-------|
| 01 | 111 | 111 | 1101 | 10100 |
| A  | N   | N   | I    | K     |



# Encoding the message



|    |     |     |      |       |    |
|----|-----|-----|------|-------|----|
| 01 | 111 | 111 | 1101 | 10100 | 01 |
| A  | N   | N   | I    | K     | A  |

**For next time:**

*“Project” 6.(A–D).*

*See new versions of these problems on Canvas There are a few changes from the book*