Huffman Coding Exercises

NAME: _______________

WORKED WITH: _______________

1. Consider the following string of characters:

   UUDDLRLRBASUUUUDDLA

   - If you were to invent a fixed-length encoding for this string, how many bits would be required per character? (Notice that there are seven distinct characters in the string.)

   - How many bits would be required to encode the entire string using your encoding?

   - Compute the frequency of each of the characters in the string:

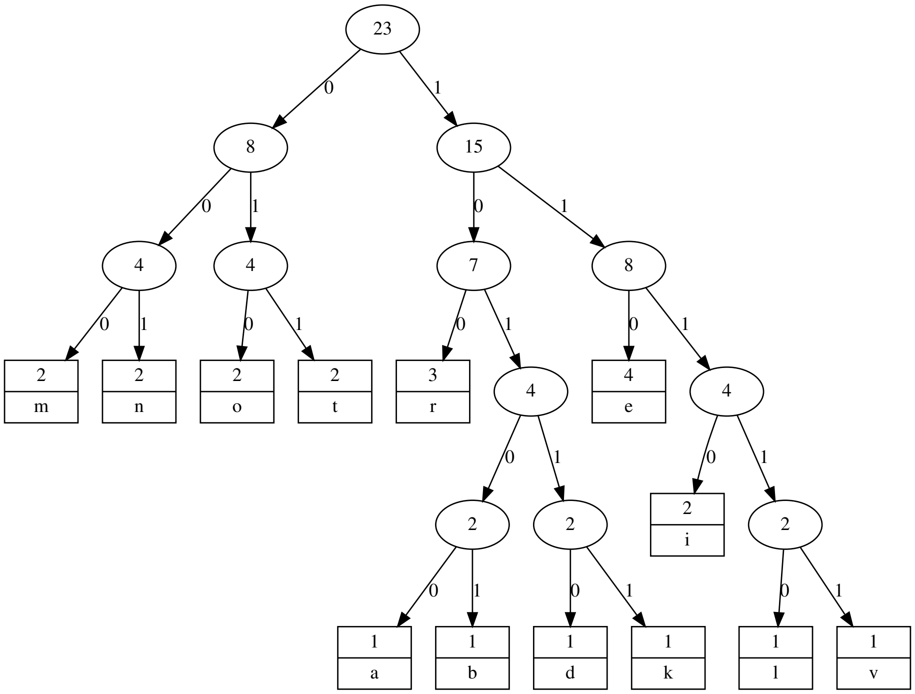
U	**D**	**L**	**R**	**B**	**A**	**S**

   - Now, build the Huffman tree corresponding to the sequence of characters above. When creating a new node, place the smaller frequency child on the left. Break ties alphabetically. In other words, if two subtrees have the same frequency, select the one containing the letter that is earliest in the alphabet. Once you have constructed your tree, use it to encode the character sequence.

   - How many bits were required to encode the sequence using your Huffman tree?

   - What was the compression ratio? (uncompressed bit count / compressed bit count)?

2. Use the following Huffman tree to decode the binary sequences below.



- ○ 10011000011000010101110100
- ○ 011010
- ○ 101101001110001101111
- ○ 0101111110100111100111110001110
- ○ There are 23 characters in the strings above, with 13 distinct characters. If we had used a fixed-length encoding, how many bits would be required to encode each string? What is the compression ratio for each string?