

Figure 1: The tree I produced

Assignment 1 – text compression

due at 23:59 on Tue Sep 25 (60 points)

Introduction

In this activity, we will investigate the Huffman algorithm for text compression. You've already seen one example of a Huffman encoding, represented by the strange-looking tree on the handout labeled "[variable-bit Huffman encoding](#)."

You will follow the Huffman algorithm and create a tree of your own, based on the character frequencies of [a message that I assigned to you](#) (see that link). The video below illustrates the algorithm on paper. I apologize that the resolution and audio quality aren't great, but it should be understandable. The final encoding and tree are also pictured below.

*Error at 21:54 – 39×5 should be **195 bits**.

You should also answer the following questions.

1. How many distinct characters did your message contain?
2. If we were using a fixed-width encoding, how many bits (per character) would you need to represent just those characters?
3. What is the most frequent character in your message, and how many times did it appear?

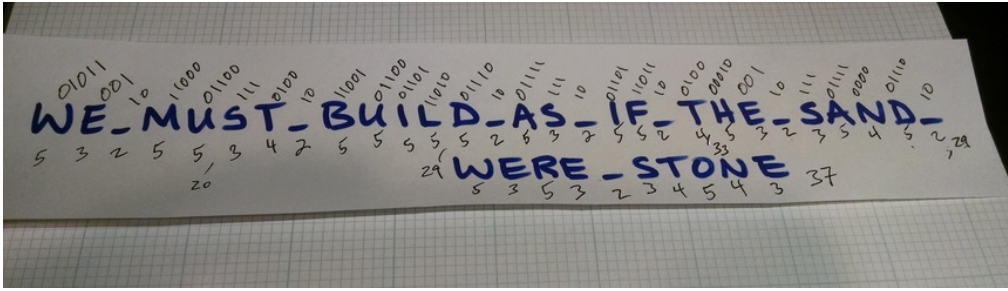


Figure 2: Encoding the phrase using that tree, result is 148 bits

4. How many bits are used to represent the most frequent character in your message?
5. What is the most number of bits used to encode any character in your message?
6. Use the tree you produced to encode the entire message you were given. How many bits are used, in total?

How to submit

Take a photo of the tree you drew that represents your variable-width Huffman encoding. Try to make it as legible as possible – redraw on a fresh sheet of paper if needed.

Write your answers to the six questions into a text document – the formats .doc, .docx, .txt, or odt are all fine.

Upload both files to [this dropbox for assignment 1](#).