Problem 1

The Authorized Version of the Bible contains 4,017,010 characters (including spaces and all punctuation). Assume that this is represented in Unicode, where each character takes 24 bits. How many copies of the Bible could you fit on a 1 Gigabyte flash drive?

Problem 2

It is said that a “picture is worth a thousand words.” Let’s test this precisely. Assume the average English word, with the space that follows it, takes 6 characters, and each character is stored as a Huffman code, as described in lecture (where we discussed Huffman codes for French and English, and the average number of bits per character for each)---assume this number, even though in this problem, there are spaces and punctuation). Take a 1 mega-pixel picture (common on cell phones, and where “mega” here = 1,000,000) of 24-bit color depth. How many words is it worth if we compare the total number of bits? That is, how many words will take up as much space as that one picture? Is it true that a “picture is worth a 1000 words” or did they misstate the relationship?

Problem 3

Using the following Huffman Code from lecture on Wednesday 10/29:

<table>
<thead>
<tr>
<th>Character</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
</tr>
<tr>
<td>C</td>
<td>11</td>
</tr>
</tbody>
</table>

give the binary encoding of the following message:

ABCCCAAB

and then decode the following message:

101011010001011011.
Problem 4

Suppose our alphabet is A, B, C, D, with probabilities as follows: A = .5, B = .3, C = .1, and D = .1. Along the same lines as the presentation in lecture on Friday, construct the Huffman Tree and the resulting Huffman Code, and then encode the message “BAD”. Show all work. Finally, calculate the average number of bits that this code would use for messages that have these probabilities, and give the corresponding compression ratio.