Feedback — Digital Communication Exercises

You submitted this homework on **Wed 3 Apr 2013 2:14 PM CDT -0500**. You got a score of **0.00** out of **7.00**. You can attempt again, if you'd like.

Question 1

Letters drawn from a four-letter alphabet have the indicated probabilities.

Letter	Probability
a	$\frac{1}{3}$
b	$\frac{1}{3}$
c	$\frac{1}{4}$
d	$\frac{1}{12}$

How many bits on the average be are going to be needed to represent this alphabet?

You entered:

Your Answer		Score	Explanation
	✘	0.00	
Total		0.00 / 1.00	

Question Explanation

We calculate the entropy:
$\bar{B} \geq -\sum p_i \log_2 p_i = -\left(\frac{1}{3}\log_2 \frac{1}{3} + \frac{1}{3}\log_2 \frac{1}{3} + \frac{1}{4}\log_2 \frac{1}{4} + \frac{1}{12}\log_2 \frac{1}{12}\right) = -\left(-\frac{1}{3}\cdot 1.585 - \frac{1}{3}\cdot 1.585 - \frac{1}{4}\cdot 2 - \frac{1}{12}\cdot 3.585\right) = 1.855 \text{ bits}$
. So at least 1.855 bits are needed *on average* to represent this alphabet.

Question 2

What is the average number of bits required to represent this alphabet with a simple code?

You entered:

Your Answer		Score	Explanation
	✘	0.00	
Total		0.00 / 1.00	

Question Explanation

Since we have a four letter alphabet, a simple code with two bits works. One example:

Letter	Simple Code
a	00

b	01
c	10
d	11

Question 3

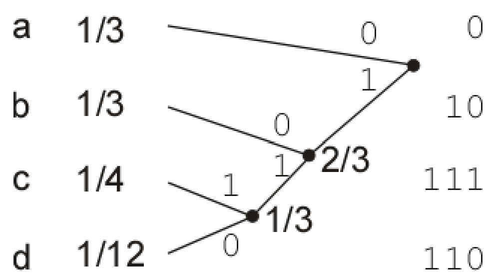
What is the average number of bits required by a Huffman code for this alphabet?

You entered:

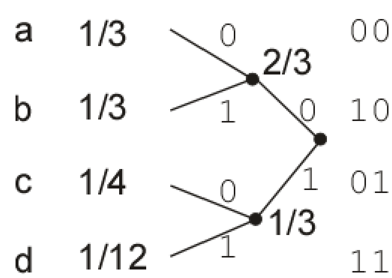
Your Answer		Score	Explanation
	✘	0.00	
Total		0.00 / 1.00	

Question Explanation

One Huffman tree is shown. Its average number of bits is $\frac{1}{3}\cdot 1+\frac{1}{3}\cdot 2+\frac{1}{4}\cdot 3+\frac{1}{12}\cdot 3=2$, a value equal to that of a simple code.



Whenever three or more probabilities or collapsed probabilities comprise the set of smallest probabilities, there are several ways of forming the coding tree. Another way of calculating the Huffman coding tree reveals that the simple code is indeed optimal.



Question 4

What is the shortest single-bit error correcting code that can be used when two data bits form the data block?

You entered:

Your Answer		Score	Explanation
	✘	0.00	

Total	0.00 / 1.00	

Question Explanation

We must have $2^{N-K} - 1 \geq N$ to have a single-bit error correcting code. Here, $K = 2$. Trying several values, the smallest N equals 5. So a $(5, 2)$ code is needed, which has an efficiency of $\frac{2}{5} = 0.4$.

Question 5

What is the shortest single-bit error correcting code that can be used when *four* data bits form the data block?

You entered:

Your Answer	Score	Explanation
✘	0.00	
Total	0.00 / 1.00	

Question Explanation

Now, a $(7, 4)$ Hamming code can be used. THis code is much more efficient; its efficiency is $\frac{4}{7} = 0.57$.

Question 6

Error Correction?
An engineer who has not taken FEE not only wants an error-correcting code, but also wants to hide the data from FEE engineers. He decides to represent 3-bit data with 6-bit codewords in which none of the data bits appear explicitly.

$$c_1 = b_1 \oplus b_2 \quad c_4 = b_1 \oplus b_2 \oplus b_3$$

$$c_2 = b_2 \oplus b_3 \quad c_5 = b_1 \oplus b_2$$

$$c_3 = b_1 \oplus b_3 \quad c_6 = b_2 \oplus b_3$$

What is the generator matrix for this code?

Type your answer in rows with a space between each element. For example, if

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

you would type `1 0 0 0 1 0 0 1 1`.

You entered:

Your Answer	Score	Explanation
✘	0.00	
Total	0.00 / 1.00	

Question Explanation

The generator matrix just mimics the coding formula.

$$G = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Question 7

What is the error-correcting capability of this code?

Your Answer	Score	Explanation
Cannot correct all single-bit errors.		
Corrects all single bit errors.		
Corrects all single-bit errors and some double-bit errors.		
Total	0.00 / 1.00	

Question Explanation

To correct single-bit errors, the Hamming distance between all pairs of codewords must be at least 3. These distances equal the number of ones in all codewords, which are equal to the columns and linear combinations of the columns of the generator matrix G. Note that the codeword for the data-bit sequence `1 1 1` is `0 0 0 1 0 0`. Therefore, the distance between at least one pair of codewords is one, which means this code *cannot* correct all single-bit errors.