Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.02 Fall 2011 **Solutions to Chapter 2** Updated on: February 8, 2012

Please send information about errors or omissions to hari; questions best asked on piazza.

1. There are a total of 8 possible 3-bit binary numbers, and we will assume they are all equally likely a priori. Alice knows it is even, so she has received $\log_2(8/4) = 1$ bit of information. Bob knows it is one of 5 possibilities, so he has received $\log_2(8/5)$ bits of information. Charlie knows it has exactly two 1's, so he knows it is either 011, 101, or 110. Therefore he has received $\log_2(8/3)$ bits of information. Deb is given all three clues, so she in fact knows that the number is 101. Hence, she has received $\log_2 8 = 3$ bits of information.

2. (a) $\log_2(1/0.001) = \log_2 1000$ bits.

   (b) $0.25 \log_2(1/0.25) + 0.75 \log_2(1/0.75) = 0.5 + 0.75 \log_2(4/3) = 2.5 - \log_2 3 = 0.92$ bits.

3. There are $2^4 = 16$ 4-bit numbers. Of these, the unknown number X differs from the known one, Y in 2 bits, which means that there are $\binom{4}{2} = 6$ possibilities for X. Hence, you have been given $\log_2(16/6) = 1.42$ bits of information.

4. We've narrowed down the choices for the dealer's face-down card from 52 (any card in the deck) to one of 49 cards (since we know it can't be one of three visible cards. Therefore, the amount of information you know now is a rather miniscule $\log_2(52/49) = 0.086$ bits.

5. (a) We get the least amount of information for Course VI (EECS): $\log_2(100/38)$ bits, because that has the highest probability of occurrence.

   (b) The tree is constructed greedily starting with the smallest two probabilities and working upwards. One solution is:

   code for course I: 0110 (length 4)
   code for course II: 010 (length 3)
   code for course III: 0111 (length 4)
   code for course VI: 1 (length 1)
   code for course X: 000 (length 3)
   code for course XVI: 001 (length 3)

   There are of course many equivalent codes derived by swapping the "0" and "1" labels for the children of any interior node of the decoding tree. So any code that meets the following constraints would be fine.

   (c) The average length would be equal to $100 \cdot (0.13 \cdot 3 + 0.12 \cdot 3 + 0.23 \cdot 3 + 0.07 \cdot 4 + 0.07 \cdot 4 + 0.38 \cdot 1) =$ 238 bits. For comparison, if we had a code that achieved entropy, the number of bits required for 100 students would be 230.5.

6. (a) $\log_2(100/25) = 2$ bits of information.

   (b) $3/100 \cdot \log_2(100/3) + 7/100 \cdot \log_2(100/7) + 25/100 \cdot \log_2(100/25) + 31/100 \cdot \log_2(100/31) +$ $34/100 \cdot \log_2(100/34) = 1.973$ bits of information.

(c) One possible encoding is:
Card A: 000, Card K: 001, Card Q: 01, Card J: 10, Card 10: 11

(d) $1000 \cdot (3 \cdot 0.03 + 3 \cdot 0.07 + 2 \cdot 0.25 + 2 \cdot 0.31 + 2 \cdot 0.34) = 2100$ bits.

(e) If the game was truly random, the best compression would not be able to compress better than the entropy of the distribution per bit sent. The entropy is given by Part (b) above, so one would not expect the compression of 1000 rounds to be done in less than 1973 bits. 43 bytes = 344 bits is substantially lower. The experimentally determined value of the actual entropy (344/1000 = 0.344 bits) indicates that one or more of the symbols must have a much higher probability of occuring than stated, which suggests a rigged game. The game is crooked; it isn't random.

7. (a) $\log_2 \frac{1}{(0.17+0.08)} = \log_2 4 = 2$ bits.

(b) O, A, and E have encodings of length 2, I/U have encodings of length 3. The code must be prefix-free, of course.

```
      /\
     /  \
    /    \
   /\    /\
  A  O  E  \
           /\
          I  U
```

A: 00, O:01, E:10, I:110, U:111

(c) $100[(.25)(3) + (.75)(2)] = 100(0.75 + 1.5) = 225$ bits.

(d) Ben is wrong, his implementation probably has a bug! The entropy is 2.19 bits... Ben's encoding uses fewer bits, which is impossible, so his code or his derviation of the expected length must be bogus in some way.