CS074-Midterm Exam
March 18, 2009

1. **Binary Representation.** Consider the sequence of three bytes whose hex encodings are:

   C4 2B 2B

   We can interpret these three bytes in several different ways.

   (a) Write each of the three bytes using binary, rather than hex notiation.

   Solution. *Just translate each hex digit into a 4-bit sequence:*

   11000100 00101011 00101011

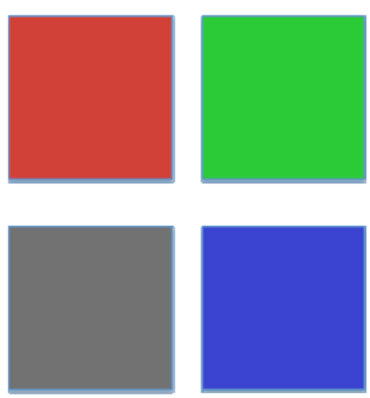
   (b) Each of the three bytes can be viewed as the hex representation of an integer. What integers are represented by C4 and 2B?

   Solution. *Use the binary representations given above:*

   $(11000100)_2= 128+64+4=193.$

   $(00101011)_2 = 32+8+2+1=43.$

   (c) The three-byte sequence can be viewed as the red, green, and blue components, respectively, of a color. One of the rectangles below is painted this color. Which one is it?



   Solution. *Since the color is 193 red, 43 green, 43 blue, the red will predominate. So the solution is the rectangle at the upper left.*

2. **Digital Image Processing**. Suppose we transform a file in one of the following four ways:

(a) Set every third byte to 0;

(b) Replace each byte by the its average with the two preceding bytes and the two following bytes. (This is not applied to the first two bytes or the last two bytes of the file, which remain unchanged). For example, if the first six bytes of the file in decimal are 48 100 115 76 83 47, then the first few bytes of the transformed file are 48, 100, 82 (=average of 48,100,115,76 and 83), 84 (=average of 100,115,76,83, and 47),...
(c) Reverse the hex digits of each byte; e.g., if a byte has value A3, replace it by 3A.
(d) Retain only the two high-order bits of each byte, and set the rest to 0: for example, if the byte has value C3 (=11000011 in binary), then it is replaced by C0 (=11000000).

These transformations were applied to a grayscale bitmap image, in which **each pixel is represented by a single byte** (00=black, FF=white) The successive bytes of the file encode the pixels of the image row by row. The results are shown on the attached page. The first picture is the original. Match each of the four transformations to one of the five following images. One of them is a fake! (Image II is blurred--it's not your eyes!)

Solution

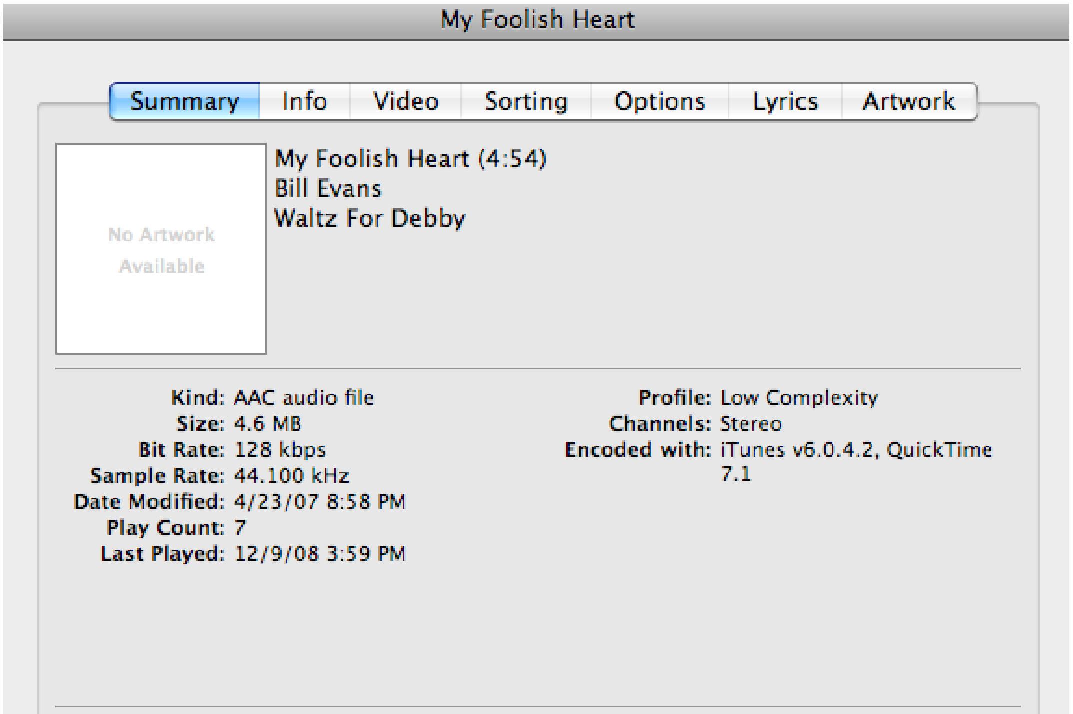
(a) turns every third pixel black. Depending on whether or not the width of the image in pixels is divisible by 3, this produces a series of vertical black lines or diagonal black lines superimposed over the image. Thus this corresponds to ***Image V*** *(and we can also conclude that the width of the image is divisible by 3).*

(b) is more subtle. Averaging makes the difference between horizontally adjacent bytes smaller, so there will be only gradual changes in brightness instead of sharp lines. The result is the blurred ***Image II****.*

(c) Swapping the low and high-order bytes completely changes the magnitude of each pixel, so we might not expect ANY resemblance at all with the original image. The most distorted image is the apparently drug-induced hallucination ***Image III****---it is remarkable how much of the original image is visible here.*

(d) keeps the pixels at approximately the same magnitudes, but only allows four different levels of brightness (black-0, dark gray 64, light gray 128, and off-white 192). This is ***Image I.***

3. **Bits and Bytes. Compression.** Below is the information from iTunes about one of the songs in my iPod.



Take particular note of the duration of the song (4 minutes, 54 seconds) the file size, bit rate and sample rate.

The sample rate gives the number of times per second (44,100) the sound was sampled in order to make the original CD from which this recording was made. (In fact, the original recording was a vinyl LP---I have lots of old stuff!---but it was eventually converted into a CD.) Four bytes are stored on the CD for each sample (two different amplitude values, because the recording is in stereo, with two bytes used to represent each amplitude).

(a) What was the size, in MB, of the song on the original CD?

Solution.

(4 bytes/sample) x (44100 samples/second) x (294 seconds) = 51861000 bytes

1 MB = 220 bytes = 1048576 bytes

File size = 51861000/1048576 = 49.5 MB.

An AAC Audio file is obtained from the original by a lossy compression method, much like MP3 compression.

(b) What is the compression ratio achieved for this song?

***Solution**. 4.6/49.5 = 9.3 %*

The bit rate gives roughly the number of bits (in this case, 128,000) of the compressed file required to represent each second of music. The figure of 128,000 is really only approximate, as different sections of the song will require more bits per second than others.

(c) Determine the actual average number of bits per second of music. Assume that the file is exactly 4.6 MB in length.

***Solution**. (4.6 MB) x (1048576 bytes/MB) x (8 bits/byte)/(294 sec) = 131250 bits/second,*

which is close to 128000, but not exact.