CMPSCI 754
Homework Assignment 1: Compression
Solution

1. Prove that no Huffman code can be a prefix of another Huffman code.

   Answer: Proof by contradiction.

   Assume that there exist two Huffman codes A and B corresponding to characters a and b respectively, such that A is a prefix of B.

   Recall that a Huffman tree from which codes are generated satisfies the following two properties:

   (a) Each character appears in a leaf node of the tree.
   (b) Every branch from a non-leaf node is labelled with a different symbol.

   So, according to our assumption, there are two possibilities:

   (a) A's path from the root node in the Huffman tree is a sub-path of B's path from the root node. In this case, a would have to lie on a non-leaf node, which contradicts property (a) above.
   (b) A's path is not a sub-path of B's path in the tree, in which case the node at which their paths diverge has two edges labelled with the same symbol. This contradicts property (b) above.

2. Explain the tradeoffs in

   (a) Choosing 8x8 as the size of a DCT block in JPEG.

   Answer: In a JPEG image, the smaller the block size, the better is the image quality achieved, as there is more correlation among pixels in a small block. But, this also leads to worse compression ratio, as there are more DC components per image which can't be compressed as efficiently.

   If the block size is too large, the image quality becomes worse because of approximate correlation among various pixels and quantization. But this leads to a better compression ratio as we have fewer non-zero DC components in the image. Another drawback could be the amount of padding required to make the image a multiple of block size.

   An 8x8 block size achieves a balance between these conflicting requirements.

   (b) Choosing 16x16 as the size of a macro-block in MPEG.

   Answer: In an MPEG stream, a macro-block is the unit of determining the motion vector distance between successive frames.

   The smaller the size of each macro-block, the more accurate would be the matching of motion vectors among frames, leading to higher compression ratios. But there would also be a larger number of vectors to match, which would lead to slower compression times.

On the other hand, the larger the macro-block, the less correlation we can find among motion vectors, and the compression ratio is worse. It leads to less matches and faster compression though.

a 16x16 macro-block achieves a balance between these conflicting requirements.

3. Design a scheme to run length encode strings that can contain any ASCII characters. (Hint: your scheme will need to distinguish the length of a run from bonafide numerals in the string).

   Answer: There are several schemes possible. A couple of such schemes are:

   (a) *Variable length encoding:* Use a special escape character, e.g. '/' followed by a number to denote the number of times a character is repeated in a run. The appearance of the character '/' itself in the string can be denoted by '//'. Thus, the string "AAABBBBBBBC/222//" can be represented as
   "A/3 B/7 C // 2/3 ///2".

   The drawback is that each run has to be less than 256, so that we might have to break up longer breaks into sub-runs.

   (b) *Fixed length encoding:* Use a two-byte sequence to denote each run, where the first byte is the ASCII character and the second byte is the length of the run. In that case, the string above can be encoded as
   "A3 B7 C1 /1 23 /2".

   Again, here, each run has to be at most 255 in length and even single occurrences have to be represented using two bytes.

4. What would you expect if you employed row major ordering rather than zig-zag ordering of AC coefficients in JPEG?

   Answer: JPEG employs DCT transforms on its blocks, which usually leads to large number of zero AC components clustered in the lower and right hand sides of the DCT matrix. Using zig-zag ordering of these coefficients leads to more opportunities for run-length encoding and hence, better compression. Using row-major ordering would reduce long runs of 0 components and lead to worse compression ratios.

5. Consider an MPEG video stream with an encoding pattern $IBBPBBPBBI$.... In what order would you store these frames if you wanted to minimize buffer space overhead at the decoder? What is the buffer space overhead of your scheme (in terms of number of frames)?

   Answer: Given an MPEG stream consisting of frames $I_1B_1B_2P_1B_3B_4P_2B_5B_6I_2\ldots$, the best way to order the frames is: $I_1P_1B_1B_2P_2B_3B_4I_2B_5B_6\ldots$.

   Thus, the buffers will be utilized and the video played out in the following order:

Buffer 1	Buffer 2	Buffer 3	Play
I_1			
I_1	P_1		I_1
I_1	P_1	B_1	B_1
I_1	P_1	B_2	B_2
	P_1	P_2	P_1
B_3	P_1	P_2	B_3
B_4	P_1	P_2	B_4
I_2		P_2	P_2
I_2	B_5	P_2	B_5
I_2	B_6	P_2	B_6
	...		

This ordering requires the use of **3** buffers.

6. Why does the MPEG standard not specify the exact motion compensation and the discrete cosine transform algorithms to be used to encode video streams?

   Answer: Different videos and applications have different characteristics, and using different motion compensation and DCT algorithms would be suitable for each of them. Hence, the MPEG standard leaves the exact algorithms to the implementations, so that appropriate choices can be made according to the context.

7. List some advantages and disadvantages of scalable compression techniques.

   Answer: Advantages:

   - This allows multiple QoS levels for clients with different capabilities.
   - It provides error and loss resilience to the enhancement layers of the stream.
   - It allows backward compatibility with MPEG-1 which can be used as the base stream.
   - It allows multiple clients to receive different qualities of the video without the server having to send redundant information.

   Disadvantages:

   - It leads to more complex compression and decompression.
   - It leads to decreased compression efficiency.

8. What is the additional storage space overhead of storing pre-computed multi-resolution images (in the form of a pyramid) as against storing only the highest resolution image? Assume that each image in your pyramid is $\frac{1}{4}^{th}$ the size of the previous image.

   Answer: Suppose there are n layers in the multi-resolution image, where the highest resolution layer is of size S. Then, the size of the n-layered image is:

$$\sum_{i=0}^{n-1} \left(\frac{1}{4}\right)^i \cdot S = \frac{1-(\frac{1}{4})^n}{1-\frac{1}{4}} \cdot S < \frac{1}{1-\frac{1}{4}} \cdot S = \frac{4}{3} \cdot S.$$

   Thus, the additional storage space overhead $< \frac{1}{3}$.

9. Why does MPEG-2 not allow frames in the base sub-stream to depend on frames in the enhancement sub-stream? What are the drawbacks of employing such a policy?

   Answer: The goal of providing multiple streams in MPEG-2 is to allow clients with limited capability to access low-quality video by receiving the base sub-stream. If the frames in the base sub-stream were to depend on those in the enhancement sub-stream, then such clients would have to fetch multiple streams anyway, and thus have to do more complex decoding as well as require more buffer space. This would in turn defeat the purpose of providing multiple QoS levels to multiple clients.