COMP2005 - IIP Sample Question + Commentary

This document contains some examples of the type of question that might be asked on the G52IIP exam paper. The question text is in bold font. Key points of the answer are in italics below, to give you an idea of what I would be looking for when marking the question.

When we set questions we are required to label them according to Bloom's Taxonomy, as examining *knowledge*, *understanding* or *application* of knowledge. Some questions touch on more than one classification. This doesn't appear on the exam paper – it's done so we can assess the coverage of the questions – but you might find it useful to think for a second what type of question you're faced with.

Some examples……

(a) Explain with the aid of an example how run-length encoding can be used to reduce the amount of memory needed to store a binary image. What type of redundancy does run-length coding exploit?

(6 marks)

This is a straightforward knowledge question, do you know what run-length encoding is?

Run length encoding, in it pure form, would store a set pairs of the form (0,N) or (1,M), signifying N 0s or M 1s. A common extension for binary images is to reduce storage space by taking advantage of the fact that 0 must be followed by 1, and vice versa, so (0, N), (1,M) becomes 0, N, M. For full marks the student should mention both approaches and show a correct example. [5 marks]

Run length encoding exploits spatial redundancy [1 mark]

(6 marks)
[knowledge]

(a) The Sobel filters are used to highlight edges by computing approximations to image derivatives.

i) Give the 3×3 Sobel filters for computing the horizontal and vertical derivatives of an image.

(4 marks)

1	*0*	*-1*
2	*0*	*-2*
1	*0*	*-1*

-1	*-2*	*-1*
0	*0*	*0*
1	*2*	*1*

Again, a straightforward knowledge question - do you know what these filters are.

ii) Show how the filters you have given in part (i) would be applied to compute gradient magnitude from the image fragment below

9	**8**	**6**
9	**6**	**3**
8	**5**	**1**

(4 marks)

Convolution: overlay each mask in turn over the image fragment, multiply corresponding values and sum the result. This gives

9 + 18 + 8 – 6 – 6 –1 = 22 for the vertical mask and

8 +10 + 1 – 9 – 16 – 6 = -12 for the vertical one.

*Gradient magnitude is computed using Pythagoras' theorem: sqrt(22*22 + (-12) * (-12))*

This moves into understanding, do you understand how this process works well enough to work through it, which is a little different to writing a description of how convolution works.

(b) To detect edges a thresholding operation is required. Many have been developed, each with its own strengths and weaknesses. Figure 1 shows gradient data extracted from the Lena image.

Figure 1

i) What thresholding method would you apply to this image, and why?
(4 marks)

There are several acceptable answers to this. You might choose thresholding with hysteresis because it is designed specifically for edge data and its dual thresholds are easier to set than single thresholds, or you might choose Rosin's algorithm because the histogram of a gradient image is unimodal. If you chose Rosin, for example, I would like a description of what it means to be Unimodal and an explanation of why gradient images are. This starts out as an application-style question. In this example you have seen the image before, in a real exam you may not have seen the specific image, but what you need to do remains the same: look at it, think about its properties, and match them to the techniques you know about.

The 'and why?' part moves into understanding. Here I'd be looking for you to say why the solution you chose matches the task, the more technical detail you can give, to show that you understand the method, the better.

ii) Explain how the method you have chosen works
(5 marks)

The answer here clearly depends on the method you have chosen, but for this number of marks I would be looking for something like 1/3 to ½ a page outline that method. This is a knowledge question and primarily tests recall of the lecture material, though I will give marks for additional relevant information about your chosen solution.

(c) What is the goal of histogram equalisation? In what way does achieving that goal (usually) enhance an image? Explain why that is. (9 marks)

The goal of histogram equalisation is an image whose histogram bins are all approximately equal valued, i.e. the probability distribution of those grey levels which appear in the image is uniform. Histogram equalisation enhances the contrast of the image.

[3 marks]
[Knowledge]

Imagine an image is being created by selecting pixel values, without replacement, from the probability density function represented by a given image's histogram. If the pixel values are clustered around particular grey levels, i.e. there are distinct peaks in the histogram, the likelihood is those colours will be selected to be adjacent to each other with high probability. The image will have low contrast where this happens. If the distribution is uniform, all values are equally likely to be selected, and all contrasts are equally possible. An image with a uniform histogram is likely to have higher contrast than the less-uniform histogram it was created from.

[6 marks]
[Understanding]

[9 marks]

(d) A grey level image and its histogram are shown below (Figures 2 and 3). Would you expect histogram equalisation to be successful in producing an enhanced image from this input? Explain your answer. (10 marks)

Figure 2

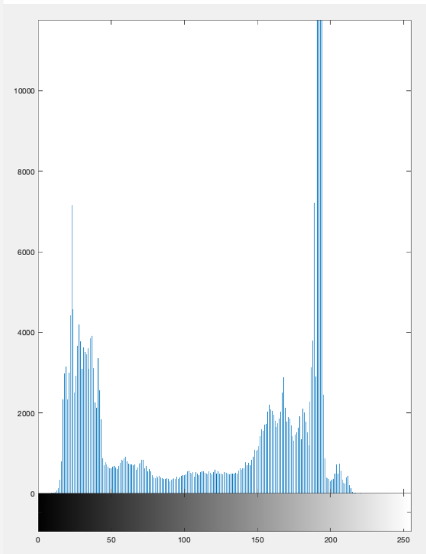


Figure 3

No. The result of applying histogram equalisation to the Figure 1 is shown below (YOU DON'T NEED TO PRESENT THIS IN EXAM – THIS IS JUST A GUIDELIE TO ESTIMATE OUTPUT IMAGE). All the pixels assigned to in a given bin in the input histogram must appear in a common bin in the output histogram. The method equalises the histogram by applying an intensity transform which effectively combines low valued bins to approximate the 'height' of higher valued bins, removing some grey levels from the output image to do so. The dominant grey levels to the left and *right of figure make it impossible for the histogram to be equalised without re-assigning the majority of the mid-range grey levels in the centre of the histogram, greatly reducing the number of grey levels in the output image and introducing the artefacts seen below. For full marks the student should recognise that this happens both at the light and dark ends of the intensity range here.*

[Understanding]
[10 marks]

d) An image has the following normalized histogram. Derive a Huffman code for each pixel value, showing how you obtained your code and calculate the compression ratio.

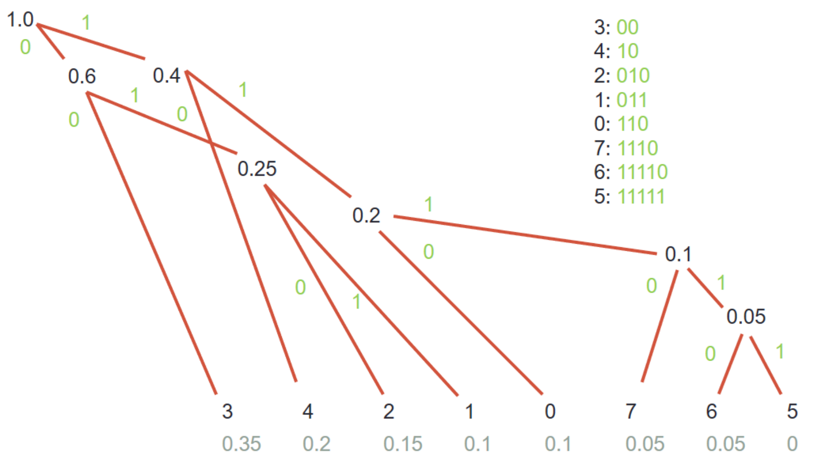
Pixel value	Normalised Frequency
0	0.3
1	0.25
2	0.15
3	0.1
4	0.1
5	0.05
6	0.05
7	0

(12 marks)

Huffman coding builds a binary tree in which symbols to be coded are nodes.
The algorithm:
Create a list of nodes, one per for symbol, sorted in order of symbol frequency (or probability)
REPEAT (until only one node left)
{
Pick the two nodes with the lowest frequencies/probabilities and
create a parent of them
RANDOMLY assign the codes 0,1 to the two new branches of the tree and delete the children from the list
Assign the sum of the parents' probabilities to their parent and insert it in the list
}
The path from root to node gives the code for corresponding symbol

Because there is a random component to the algorithm, many different codings are possible. For full marks the student should apply this method correctly to the data provided An example Table and Huffman coding should be created using the algorithm explained above :

Pixel Value	Normalised Frequency							
3	0.35	0.35	0.35	0.35	0.35	0.4	0.6	1.0
4	0.2	0.2	0.2	0.2	0.25	0.35	0.4	
2	0.15	0.15	0.15	0.2	0.2	0.25		
1	0.1	0.1	0.1	0.15	0.2			
0	0.1	0.1	0.1	0.1				
7	0.05	0.05	0.1					
6	0.05	0.05						
5	0							



Tree view – the coded values can be made using random assignment of 0s and 1s to each branch`

		Code length	Normalised Freq,
3: 00	0: 110	3	0.1
4: 10	1: 011	3	0.1
2: 010	2: 010	3	0.15
1: 011	3: 00	2	0.35
0: 110	4: 10	2	0.2
7: 1110	5: 11111	5	0
6: 11110	6: 11110	5	0.05
5: 11111	7: 1110	4	0.05

Mean bits/pixel L_{avg} = 0.3 +0.3 +0.45 +0.70 + 0.4 + 0 + 0.25 + 0.2 = 2.6

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k)p(r_k)$$

Without compression L_{avg} = 3

Compression ratio = 2.6/3 = 0.86

[12 marks] Application