THIS PAPER IS NOT TO BE REMOVED FROM THE EXAMINATION HALLS

UNIVERSITY OF LONDON **CO3325 ZB**

BSc Examination

COMPUTING AND INFORMATION SYSTEMS, CREATIVE COMPUTING, AND COMBINED DEGREE SCHEME

Data Compression

Date and Time: Wednesday 3 May 2017: 14.30–16.45

Duration: 2 hours 15 minutes

There are THREE questions in this paper. Candidates should answer all **THREE** questions. All questions carry equal marks and full marks can be obtained for complete answers to **THREE** questions. The marks for each part of a question are indicated at the end of the part in [.] brackets.

There are 75 marks available on this paper.

A hand held calculator may be used when answering questions on this paper but it must not be pre-programmed or able to display graphics text or algebraic equations. The make and type of machine must be stated clearly on the front cover of the answer book.

Question 1

(a) Explain why the Reflected Gray code is a good representation for coding the colours of greyscale images. Derive the *reflected Gray code* for the decimal number 12. [5]

(b) What will be the output if the HDC algorithm is applied to the sequence below? Explain the meaning of each *control symbol* that is used in your answer. [6]

YYB␣␣␣␣␣␣␣␣␣F␣␣KK8888888␣␣BBBHH

(c) Suppose that a binary sequence of length 5 (symbols) was encoded on the binary alphabet (B, W) using the Arithmetic encoding algorithm, the probability $Pr(B) = 0.3$, and the encoded output is 0.34.

Demonstrate, step by step, how the Arithmetic decoding algorithm would derive the original sequence of symbols.

In your answer, you should

i. Outline the decoding algorithm in a flowchart or pseudocode. [4]
ii. For each iteration, trace the values of the variables $L, d, d*p1, d*p2, [L, L+d*p1), [L+d*p1, L+d)$ and any output symbols, and present them in a table like the one below. [9]
iii. Give the decoded string output by the algorithm. [1]

Iteration	L	d	d∗p1	d∗p2	[L, L+d∗p1)	[L+d∗p1, L+d)	Output
0	0	1					
1							
2							
3							
4							
5							

Question 2

(a) Demonstrate how to derive step by step a canonical minimum-variance Huffman code for the message `ABAABCDAAA` using *two* lists. [5]

(b) Encode the string `AABACCABBAAACCC` following the LZW algorithm. Assume that the dictionary initially contains single characters A-F and occupies cells at 0–5 only. Demonstrate the content changes of the main variables x, $word$, $word + x$, and the dictionary. [8]

(c) Consider $S_1 = (A, B)$, the alphabet of a binary source file in which symbols A and B occur independently with the probability distribution $P_1 = (p_A, p_B)$, respectively. Let S_2 denote the four-element extended alphabet from S_1, P_2 denote the probability distribution of S_2, H_1 denote the entropy of S_1, and H_2 denote the entropy of S_2.

i. Write the extended alphabet S_2. [2]

ii. Suppose $P_1 = (0.2, 0.8)$. Compute the probability distribution P_2, entropy H_1 and entropy H_2. Show all your work. [4]

iii. Demonstrate, for any P_1, $H_1 = \frac{1}{2}H_2$, that is, the (first-order) entropy of S_1 is half the value of the entropy of S_2. [6]
[*Hints*]: $H_1 = \frac{1}{2}H_2$ means $2H_1 = H_2$; $p_{AB} = p_A \times p_B$; $log_2(p_A)^2 = 2log_2 p_A$; $log_2(p_A \times p_B) = log_2 p_A + log_2 p_B$; $p_A + p_B = 1$.

Question 3

(a) Explain briefly what is meant by a *cartoon-like image* in the context of Data Compression. Give an example of such an image by providing a 5×5 matrix of colour data (colour map) to aid your explanation. [5]

(b) Consider the binary code $B = (0, 11, 101, 011)$. Would it be possible to find a prefix code that is of the same length as B? Would it be possible to find a prefix code that is shorter than B? Justify your answers. [5]

(c) Outline in a flowchart or pseudocode, the adaptive Huffman algorithm for encoding. [5]

(d) Demonstrate how the adaptive Huffman algorithm works for encoding `BAAHA`. Assume that the fixed-length codes for characters $A, \cdots, Z$ are $01000001, \cdots, 01011010$, respectively, and that each new symbol is added to the front (left most position) of the alphabet before a stable sort. A stable sort would not change the order of two symbols that have the same frequency.

i. Trace the values of the `input`, `output`, `alphabet` and the `tree structure` on each iteration. [8]
ii. Write down the encoding result and compute the compression factor. [2]

END OF PAPER