THIS PAPER IS NOT TO BE REMOVED FROM THE EXAMINATION HALLS

UNIVERSITY OF LONDON **CO3325 ZA**

BSc Examination

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

Data Compression

Date and Time: Friday 20 May 2016: 10.00–12.15

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) Consider the alphabet of four symbols (A, B, C, D). Discuss the possibility of finding

i. a uniquely decodable binary code in which the codeword for A is of length 2, for B is of length 1, and for C or D is of length 3.
ii. a shorter variable length prefix code than the one described in part a.(i).

Give your reasons and one example to justify your argument. [7]

(b) Amenda claims that the binary code (1, 01, 001, 010) is a prefix code since it satisfies the Kraft inequality. Check if the code indeed satisfies the Kraft inequality and explain what is wrong with Amenda's claim. [7]

(c) 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]

`KKH␣␣␣␣␣␣␣␣␣T␣␣UU5555555␣␣BBBAA`

(d) Discuss the absolute limit of lossless compression by showing why more than 99% of files cannot be compressed even by one byte. [5]

Question 2

(a) Consider a source with a small alphabet (A, B) and an imbalanced probability distribution (0.3, 0.7) for A and B respectively. Assume that the static Huffman encoding algorithm is applied to compress the source. [10]

i. Suggest a preprocessing method that can improve the *compression efficiency* of the Huffman algorithm on the source.
ii. Demonstrate step by step how the source can be compressed using the method in a.(i) and the Huffman algorithm.
iii. Demonstrate that the compression efficiency with the preprocessing method is better than without.

(b) Outline the Arithmetic decoding algorithm.

A binary sequence of length 5 (symbols) was encoded on the binary alphabet (B,W) using Arithmetic encoding algorithm. Suppose that the probability $Pr(B) = 0.2$ and the encoded output is 0.12. Demonstrate how the Arithmetic decoding algorithm would derive the original sequence of symbols step by step. Copy the table below in your answer book and fill the updated values of $L, d, d*p1, d*p2, [L, L+d*p1), [L+d*p1, L+d)$, $Output$ for each step from step 1, where step 0 represents the initial state. [10]

step	L	d	d*p1	d*p2	[L, L+d*p1)	[L+d*p1, L+d)	Output
0	0	1					
1							
2							
3							
4							
5							

(c) The paragraph below contains a number of text gaps, but is meant to explain, with the aid of a small example, some concepts in video compression. Write in your answer book the missing words to form truth statements. [5]

In video compression, certain frames are selected to be ___(1)___ as ___(2)___, but others, i.e. those ___(3)___ are replaced by ___(4)___. P-pictures means ___(5)___. They are the frames to be compared with ___(6)___ frames. For example, ___(7)___.

Note: The last gap, i.e. gap (7) (example) can be filled by words or a diagram as your answer.

Question 3

(a) Describe the main idea of *predictive encoding*. Suppose the matrix below represents the pixel values (in decimal) of part of a grayscale image. Using the predictor $x = (Q + S)/2$ in JPEG $\begin{array}{|c|c|}\hline \text{T} & \text{S} \\ \hline \text{Q} & \text{x?} \\ \hline \end{array}$, illustrate step by step how the predictive encoding algorithm may be applied to the matrix: [10]

1	1	1	1
5	1	1	1
5	5	5	5
7	9	5	5

(b) Demonstrate step by step how the Basic LZW *encoding* and *decoding* algorithms maintain the same version of a dictionary without ever transmitting it in a separate file, using a small string `AGGAGAGAG` as an example. Assume that the dictionary consists of $A, \cdots, Z; a, \cdots, z; 0, \cdots, 9$ initially, with the index of `A` being 33, and the index of the first available location is 96 for the first new entry. In each step, trace the updated values of (1) symbols to be read, (2) x, (3) $word + x$, (4) output, (5) new entry to the dictionary, and (6) $word$. [15]

END OF PAPER