UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

Department of Computing

B. Sc. Examination 2018

IS53057A
Advanced Algorithms and Data Structures

Duration: 2 hours 15 minutes

Date and time:

This paper is in two parts: part A and part B. You should answer ALL questions from part A and TWO questions from part B. Part A carries 40 marks, and each question from part B carries 30 marks. The marks for each part of a question are indicated at the end of the part in [.] *brackets.*

There are 100 marks available on this paper.

Electronic calculators must not be programmed prior to the examination. Calculators which display graphics, text or algebraic equations are not allowed.

THIS PAPER MUST NOT BE REMOVED
FROM THE EXAMINATION ROOM

Part A

Question 1 Parts (a)–(g) require true or false answers with justifications. You should read each statement carefully and then determine whether the statement is TRUE or FALSE. If the statement is true, explain why it is true or give a supportive example. If it is false, explain what should be the correct answer or provide a counterexample.

(a) In dynamic programming, the output of (current) stage n becomes typically the input to (previous) stage $n-1$. True or False? Justify your answer. [4]

(b) The Bubble Sort Algorithm does not need to employ the dynamic programming approach. True or False? Justify your answer. [4]

(c) If a dynamic-programming problem satisfies the optimal-substructure property, then a locally optimal solution is globally optimal. True or False? Justify your answer. [4]

(d) In *4-queen problem*, the symmetry of the queens can be used for finding the alternative solution to a known solution. True or False? Justify your answer. [4]

(e) Huffman coding is not optimal unless all the probabilities of the alphabet is the same. True or False? Justify your answer. [4]

(f) In the context of Quantum Computing, the superposition s for a level 2 quantum system can be measured as a value between 0 and 1, i.e. $0 < s < 1$. True or False? Justify your answer. [4]

(g) The efficiency of a parallel algorithm is defined as

$$E(n,p) = \frac{T(n,1)}{pT(n,p)}$$

True or False? Justify your answer. [4]

(h) In the following diagrams, which one better represents behaviour of a randomised algorithm? Justify your answer. Assume X is deterministic and Y is random. [4]

(i) random input $\rightarrow$ $\boxed{\text{XXXXXXX}}$ $\rightarrow$ output distribution

(ii) deterministic input $\rightarrow$ $\boxed{\text{XXXYXXX}}$ $\rightarrow$ output.

(i) Show there is no solution to the "2-queen problem". [4]

(j) Explain, with the aid of an example, what is meant by an *optimisation problem* in the context of Algorithm Design. [4]

Part B

Question 2

(a) Explain what is meant by saying that a problem is NP-hard, and what kind of algorithmic solutions people would seek for a known NP-hard problem. [7]

(b) Explain why the following algorithm is not normally considered as a polynomial time algorithm in the context of efficient algorithms. What is it meant by a polynomial time algorithm? [6]

```
int algorithmX(n) {
      result=2
      for i=1 to n
         result=result*result
      return result
}
```

(c) What is *backtracking* in the context of algorithm design? Demonstrate and explain how the backtracking may be used for solving the *4-Queen problem* in diagrams. [17]

Question 3

Consider the coin-changing problem.

(a) Describe the so-called *coin-change* problem. [5]

(b) Write two different instances of the problem including one *special case. Hint: A special case can be a situation where there is no solution or that requires an unusual solution. Add assumption if necessary.* [6]

(c) Design and outline an algorithm for this problem (though you are not expected to provide a full solution for full credit.). Explain the data structure to be used and the input and output the algorithm expects. [14]

(d) Demonstrate how the time complexity can be analysed of your algorithm for the worst case. [5]

Question 4

(a) Design and propose an approximation algorithm for the bin-packing problem using a greedy approach. Show all your work and highlight (or mark clearly) the final version of the algorithm. [15]

(b) Given a linked list of n elements, all of which are stored in an array $A[1..n]$ of processors. Propose a *parallel algorithm* to compute for each element its rank in the list. What is the time complexity of your algorithm in $O()$ notation? Justify your answer. [15]