University of Bath

DEPARTMENT OF COMPUTER SCIENCE EXAMINATION

CM30073: ADVANCED ALGORITHMS AND COMPLEXITY

Candidates may use university-supplied calculators.

Full marks will be given for correct answers to THREE questions. If you opt to answer more than the specified number of questions, you should clearly identify which of your answers you wish to have marked. In cases where you have failed to identify the correct number of answers the marker is only obliged to consider the answers in the order they appear up to the number of answers required.

PLEASE FILL IN THE DETAILS ON THE FRONT OF YOUR ANSWER BOOK/COVER AND SIGN IN THE SECTION ON THE RIGHT OF YOUR ANSWER BOOK/COVER, PEEL AWAY ADHESIVE STRIP AND SEAL.

TAKE CARE TO ENTER THE CORRECT CANDIDATE NUMBER AS DETAILED ON YOUR DESK LABEL.

DO NOT TURN OVER YOUR QUESTION PAPER UNTIL INSTRUCTED TO BY THE CHIEF INVIGILATOR.

1. (a) Explain the purpose of introducing the concepts of *Non-deterministic Turing Machine (NTM)* and the complexity class *NP*. Illustrate your explanation by two examples of concrete **Yes-No**-problems which lead to these concepts. [6]

   (b) When is an input word said to be *accepted* by an NTM? When is a **Yes-No**-problem said to be *solved* by an NTM? Give a definition of the *complexity* of an NTM. [7]

   (c) Define the complexity class *co-NP*, and illustrate by an example of a concrete **Yes-No**-problem why one might be interested in this concept. Compare $P \neq NP$ and $NP \neq$ *co-NP* hypotheses. What can you say about the set $NP \setminus (P \cup NPC)$ (where NPC is the class of all NP-complete problems) under the assumption that $P \neq NP$? [7]

2. (a) Formulate the q-*Clique* problem for an undirected graph and the *CNF Satisfiability* problem. Prove that q-*Clique* problem is NP-complete using a polynomial transformation to this problem of the *CNF Satisfiability* problem (assumed NP-complete). [7]

   (b) Formulate the (**Yes-No** version of) the *Travelling Salesman* problem and the *Hamiltonian Cycle* problem. Prove that the *Travelling Salesman* problem is NP-complete using a polynomial transformation to this problem of the *Hamiltonian Cycle* problem (assumed NP-complete). [7]

   (c) Prove that the optimization version of the *Travelling Salesman* problem is NP-hard. [6]

3. (a) Define the terms "approximation algorithm with polynomial complexity" and "ratio bound". [4]

   (b) Describe an approximation algorithm with polynomial complexity for the optimization version of the *Travelling Salesman with triangle inequality* problem, using if necessary, some known polynomial complexity algorithms as subroutines. [8]

   (c) Does there exist an approximation algorithm with polynomial complexity and a constant ratio bound for the standard optimization version of *Travelling Salesman*? Justify your answer. [8]

4. (a) Give a definition of a semialgebraic set in $\mathbf{R}^n$. Characterize (without giving a proof) semi-decidable sets of Real Numbers machines in terms of semialgebraic sets. [3]

   (b) Give a definition of the class NP for Real Numbers machines. Describe an example of an NP-complete problem in this class (you don't need to *prove* NP-completeness of this problem). [7]

   (c) Prove the lower complexity bound $\Omega(n \log n)$ for sorting-by-comparisons. [7]

   (d) Formulate the *Distinctness* problem. What is the best lower complexity bound for this problem when using Algebraic Computation Trees? (You don't need to prove this bound.) [3]