Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Advanced Algorithms I

Date: Wednesday 18th January 2012
Time: 14:00 - 16:00

Please answer THREE Questions: ONE Question from EACH Section
Use a SEPARATE answerbook for EACH Question

For full marks your answers should be concise as well as accurate
Marks will be awarded for reasoning and method as well as being correct.

This is a CLOSED book examination
The use of electronic calculators is NOT permitted

[PTO]
Section A

Answer exactly one question from this section

1. a) Describe how to calculate the fail array for the Knuth-Morris-Pratt (KMP) Algorithm.

   (3 marks)

   b) Compute the fail array for the string of length 5: “Hello”.

   (3 marks)

   c) Give the KMP algorithm, and describe how it works.

   (5 marks)

   d) Analyse the time complexity of the KMP algorithm for a pattern of length \( m \) and an input of length \( n \).

   (5 marks)

   e) Is this the best algorithm? If not, what is a better one and why?

   (4 marks)

2. a) Briefly explain how to represent graphs as adjacency lists and adjacency matrices.

   (2 marks)

   b) On what grounds do we use one or the other of these representations?

   (2 marks)

   c) In a depth-first traversal of a directed graph, what are tree-edges, back-edges, forward-edges, and cross-edges?

   (4 marks)
d) A strongly connected component analysis divides a graph up into cyclic components, coupled with an acyclic dependency between these components. Using an example, describe this process.

(4 marks)

e) Give Tarjan’s Strongly connected component algorithm, and explain how it works.

(6 marks)

f) What is the time complexity of Tarjan’s Algorithm?

(2 marks)
Section B

Answer exactly two questions from this section

3. a) The problem REACHABILITY is defined as follows

Given: a directed graph \( G = (V,E) \) and nodes \( u_0, v_0 \) in \( V \)
Return: Yes, if there is a path in \( G \) from \( u_0 \) to \( v_0 \); No, otherwise.

Write a deterministic algorithm for REACHABILITY and argue that it runs in time \( O(n^2) \), where \( n = |V| \). (You should present your algorithm in pseudo-code.)

(4 marks)

b) The following diagram represents a flow network with integral (whole-number) capacities on each edge, together with a flow for that network, indicated by the numbers in parentheses.

![Flow Network Diagram]

Draw the ‘auxiliary graph’ for this network, as used in the lectures to compute optimum flows.

(3 marks)

c) Hence or otherwise, compute an improved flow for the network depicted in part 3b. Clearly indicate in a fresh drawing the new flow.

(3 marks)

d) Explain how, given a flow network of \( n \) nodes with integral capacities bounded by some number \( C > 0 \), an optimal flow can be computed in time \( O(Cn^3) \). You need not prove the correctness of the algorithm.

(4 marks)
e) Assuming the correctness of the algorithm given in your answer to 3d, explain why, if a flow network has integral capacities, there must be a maximal flow with integral values. Give an example of a flow network with integral capacities together with an optimal flow involving some non-integral values. (Use a small network; otherwise it will be hard to know that your flow is optimal.)

(3 marks)

f) The problem MATCHING is defined as follows

Given: a bipartite graph $G = (V, W, E)$
Output: Yes if $E$ includes a 1–1 onto function from $V$ to $W$; No otherwise.

Explain how the problem MATCHING can be solved in time $O(n^3)$.

(3 marks)

4. a) The problem REACHABILITY is defined as follows

Given: a directed graph $G = (V, E)$ and nodes $u_0, v_0$ in $V$
Return: Yes, if there is a path in $G$ from $u_0$ to $v_0$; No, otherwise.

Write a non-deterministic algorithm for REACHABILITY and argue that it runs in space $O(\log n)$. (You should present your algorithm in pseudo-code.)

(4 marks)

b) Let $C$ be a class of problems. What is meant by the following terms?

   i) log-space many-one reduction
   ii) $C$-hard (with respect to log-space many-one reductions)
   iii) $C$-complete (with respect to log-space many-one reductions)

(3 marks)

c) Carefully stating any standard (non-trivial) properties of log-space many-one reducibility that you require, explain why, if a problem $P$ is log-space many-one reducible to a problem $Q$, and $P$ is hard for a complexity class $C$, then $Q$ is hard for $C$.

(3 marks)
d) Suppose $G = (V, E)$ is a directed graph, and let $u_0, v_0$ be vertices of $G$. Regarding the vertices $V$ as proposition letters, define the set of 1- and 2-literal clauses $\Gamma_G$ by

$$\Gamma_G = \{u_0, \neg v_0\} \cup \{\neg u \lor v \mid (u, v) \text{ is an edge of } G\}.$$ 

i) Show that, if $v_0$ is not reachable from $u_0$ in $G$, then $\Gamma_G$ is satisfiable. (You must define a truth-assignment to the proposition letters in $V$ and show that it satisfies $\Gamma_G$.)

ii) Show that, if $\Gamma_G$ is satisfiable, then $v_0$ is not reachable from $u_0$ in $G$. (Hint: suppose there is a path from $u_0$ to $v_0$ in $G$; show that no truth-assignment can make all the clauses in $\Gamma_G$ true.)

iii) Using a standard result on the complexity of REACHABILITY (which you may simply state), deduce that 2-SAT is \text{NLOGSPACE}-hard.

(10 marks)

5. a) Define the decision problems:

i) linear programming feasibility

ii) integer programming feasibility

(You may be used to calling these problems, simply, \text{linear programming} and \text{integer programming}, respectively.)

(4 marks)

b) Write down a positive instance of linear programming feasibility which is at the same time a negative instance of integer programming feasibility. (Your answer may consist of a single equation!)

(2 marks)

c) Write down a linear equation (not an inequality) in two variables, $x$ and $x'$, such that: (i) there is a solution over $\mathbb{N}$ in which $x$ takes the value 0, and also a solution over $\mathbb{N}$ in which $x$ takes the value 1; and (ii) in any solution over $\mathbb{N}$, $x$ takes either the value 0 or the value 1.

(2 marks)
d) Write down a linear equation (not an inequality) in four variables, \( z_1, z_2, z_3, y \) such that: (i) if \( z_1, z_2, z_3 \) are natural numbers at least one of which is greater than zero, then there exists a \( y \in \mathbb{N} \) such that the values \( z_1, z_2, z_3, y \) are a solution; and (ii) in any solution over \( \mathbb{N} \), at least one of \( z_1, z_2, z_3 \) is greater than zero.

(2 marks)

e) Let \( \Gamma = \{ \gamma_1, \ldots, \gamma_m \} \) be a set of 3-literal clauses over proposition letters \( p_1, \ldots, p_n \). Using your answers to parts 5c and 5d, write down a system of linear equations \( E \) (not inequalities) in variables \( x_1, \ldots, x_n, x'_1, \ldots, x'_n \) and \( y_1, \ldots, y_m \), such that \( \Gamma \) is satisfiable if and only if \( E \) has a solution over \( \mathbb{N} \). Show that your system of equations has the required properties. Notice that it follows from this result that integer programming is NP-hard.

(10 marks)

6. a) Let \( M \) be a (non-deterministic) Turing machine and \( x \) a string over its alphabet. What does it mean to say that \( M \) accepts \( x \)?

(2 marks)

b) Define the terms

i) PTIME
ii) NPTIME
iii) PSPACE

(6 marks)

c) Given an informal explanation of why NPTIME \( \subseteq \) PSPACE.

(4 marks)
d) Recall that a *quantified Boolean formula in prenex form* is an expression of the form $Q_1 p_1 \ldots Q_n p_n \psi$, where $\psi$ is a formula of propositional logic involving only the proposition letters $p_1, \ldots, p_n$, and each $Q_i$ is $\forall$ or $\exists$. For each of the following quantified Boolean formula in prenex form, say whether they are true or false:

i) $\exists p_1 \forall p_2 (p_1 \lor p_2)$

ii) $\forall p_1 \exists p_2 \neg (p_1 \lor p_2)$

iii) $\forall p_1 \exists p_2 (p_1 \leftrightarrow p_2)$

iv) $\exists p_1 \forall p_2 (p_1 \leftrightarrow p_2)$

(4 marks)

e) Explain informally why the satisfiability problem for prenex form quantified Boolean formulas is in PSPACE.

(4 marks)