Friday 13 December 2019
9.30 am – 11.00 am
(1 hour 30 minutes)

DEGREES of MSci, MEng, BEng, BSc, MA and MA (Social Sciences)

ALGORITHMICS I (H)

Answer all 4 questions

This examination paper is worth a total of 60 marks.

The use of calculators is not permitted in this examination.

INSTRUCTIONS TO INVIGILATORS: Please collect all exam question papers and exam answer scripts and retain for school to collect. Candidates must not remove exam question papers.

1. Suppose that the Boyer-Moore algorithm is to be used to search for the first occurrence, if any, of a string s in a text t.

(a) Define the array p (indexed by the characters appearing in the text) of values that would be set up by the algorithm in its pre-processing phase, and give a simple algorithm to compute this array of values for a starting s of length `m`. [4]

(b) Describe the main searching phase of the algorithm, with particular reference to how the array p is used to facilitate the search and the jump step cases, i.e. what to do when there is a mismatch between the text and string. [6]

(c) Indicate precisely which character comparisons would be made if the Boyer-Moore algorithm were used to locate the first occurrence of the string $s =$ `tgta` in the text $t =$ `agagtactgta`. [5]

2. **(a)** Describe Dijkstra's algorithm for finding the shortest path between two specified vertices u and v in a weighted undirected graph, where the length of a path is defined as the sum of the weights of the edges in that path.

[6]

(b) What restriction on the edge weights is assumed by Dijkstra's algorithm? Give an example to show that the algorithm may not work correctly if this restriction is not satisfied.

[4]

(c) Suppose that you wanted to know whether the path returned by the algorithm was the unique shortest path between u and v. How could the algorithm be extended to return, in addition, a boolean value indicating the answer to this question?

[5]

3. **(a)** What is meant by each of the following:

(i) the class NP; [1]

(ii) a polynomial-time reduction; [3]

(iii) the statement that a given decision problem Π is NP-complete. [2]

(b) Consider the following two decision problems:

Name: Graph 3-Colouring Problem (**3-GCP**)
Instance: undirected (unweighted) graph $G=(V,E)$;
Question: can one of 3 colours be attached to each vertex of G such that adjacent vertices always have different colours?

Name: Clique Cover (**CC**)
Instance: undirected (unweighted) graph $G=(V,E)$ and target integer K;
Question: can we partition the vertices into K disjoint sets such that each set forms a clique in G?

Suppose that you have a proof that **3-GCP** is NP-complete, present a formal proof showing that **CC** is NP-complete.

[9]

Hint: Consider the complement graph $G'=(V,E')$ of $G=(V,E)$ where

$$E' = \{(u,v) \in V \times V \mid (u,v) \notin E\}$$

and a specific value of K.

4. **(a)** Describe, using diagrams or otherwise, deterministic finite state automata to recognize the following languages:

- all strings over the alphabet $\{a,b\}$ that start and end with the same character;
- all strings over the alphabet $\{a,b\}$ that start and end with different characters.

You can assume the empty string is not part of either language and a single character is part of the first language.

[6]

(b) Provide regular expressions for the above two languages.

[2]

(c) Give an outline and describe, using a suitable form of pseudocode, or otherwise, a Turing Machine that recognizes the language $L = \{a^n bc^{2n} \mid n \geq 0\}$ over the alphabet $\Sigma = \{a,b,c\}$.

[7]