UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1999

BEng Honours Degree in Computing Part II
MEng Honours Degrees in Computing Part II
BSc Honours Degree in Mathematics and Computer Science Part II
MSci Honours Degree in Mathematics and Computer Science Part II
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the
Associateship of the Royal College of Science
Associateship of the City and Guilds of London Institute

PAPER 2.8 / MC 2.8

ALGORITHMS, COMPLEXITY AND COMPUTABILITY
Wednesday, May 12th 1999, 2.00 – 3.30

Answer THREE questions

For admin. only:
paper contains 4 questions

1 a Design a 1-tape Turing machine M which, given a word w, halts and succeeds if w is non-empty, leaving w*w on the tape, and halts and fails otherwise. The input alphabet is {1,0,*} and the full alphabet is {1,0,*,^}. You may assume that square 0 is implicitly marked.

Draw a diagram of your Turing machine and explain the notation you have used. Describe briefly how your Turing machine works, illustrating your answer with the case where w = 10*1.

b Derive an expression for the time function $time_M(n)$ of the Turing machine in your answer to part a, explaining each component clearly.

The two parts carry, respectively , 70%, 30% of the marks.

2a i) Describe briefly the operation of the Turing machine N in fig.1. (It has input alphabet $\{a,b\}$ and full alphabet $\{a,b,\wedge\}$.)

ii) Draw a simplified diagram for N using a parameter $x \in \{a,b,\wedge\}$.

iii) What is the input-output function of N?

b Write out *code* (N). Explain your notation.

c Assume S is any standard Turing machine and w is any word of C (the standard typewriter alphabet).

Given the standard Turing machines DUP, EDIT with input-output functions $f_{DUP}(w) = w{*}w$, $f_{EDIT}(code(S){*}w) = code(S[w])$ respectively, the standard Turing machine REV which reverses its input so that, for example, $f_{REV}(xyz) = zyx$ and the universal Turing machine U such that $f_U(code(S){*}w) = f_S(w)$:

evaluate:

i) $f_U(f_U(code(EDIT){*}code(REV){*}hello){*}goodbye)$.

ii) $f_U(f_U(f_U(f_U(code(DUP){*}code(EDIT)){*}code(REV){*}buy){*}sell){*}float)$.

The three parts carry, respectively, 40%, 40%, 20% of the marks.

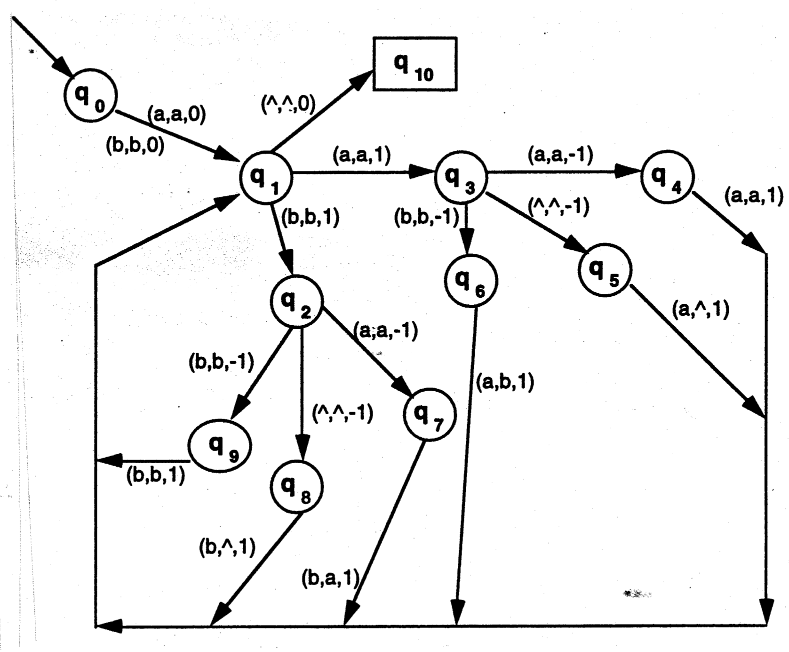

fig.1

[*Turn over ...*

3a Explain what is meant by a *minimal spanning tree* of a connected, weighted graph, G.

Define the *separation property* of a spanning tree for a weighted graph.

b Let T be any spanning tree of an arbitrary connected weighted graph G. Prove by contradiction or otherwise that if T has the separation property, then T is a minimal spanning tree of G.

c Derive a minimal spanning tree of the graph in fig.2, starting at A. Show your working.

The three parts carry, respectively, 25%, 40%, 35% of the marks.

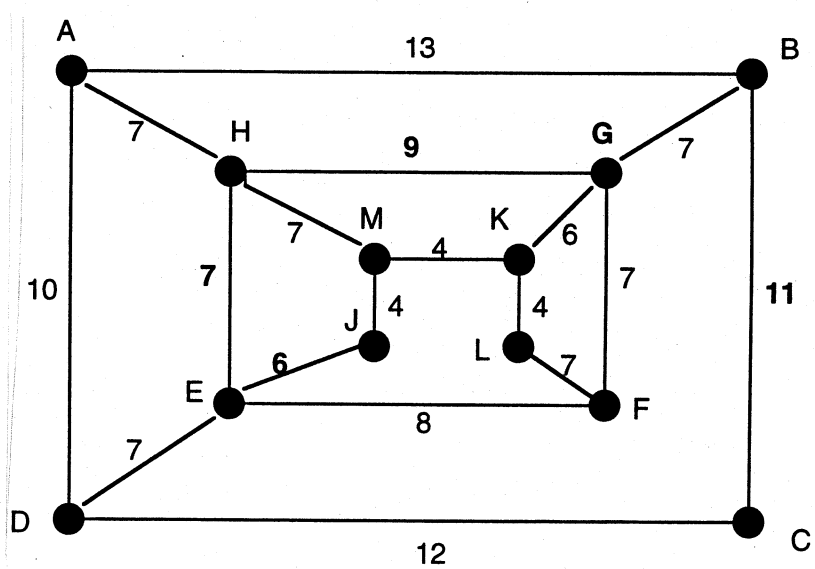

fig.2

4a Define the complexity classes P and NP of yes/no problems. Explain the relationship between P and NP.

b What do we mean by the expression *A reduces to B* where A and B are yes/no problems?

c Prove that the class NP is closed downwards under reduction.

d Define the class NPC of problems. Describe briefly two methods of determining whether a new problem, C, is in NPC.

The four parts carry, respectively, 30%, 20%, 20%, 30% of the marks.

[*End of paper*