UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1996

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 / MC2.8

ALGORITHMS, COMPLEXITY AND COMPUTABILITY
Friday, May 10th 1996, 4.00 - 5.30

Answer THREE questions

For admin. only: paper contains
4 questions
2 pages (excluding cover page)

1a Design a Turing machine M, with input alphabet {a,b}, that accepts any input word of the form aaa...abbb...b that has the same number of a's and b's, and rejects any other input word.

For example, M should accept the words aaabbb, and ε. It should reject the words abab, abbb, aa, and bbaa.

Your Turing machine can have more than one tape, and you may assume that square 0 of each tape is implicitly marked. You may use pseudo-code or a flow-chart diagram; in the latter case you should explain your notation for instructions (that is, what the labels on arrows mean).

b i) Describe, giving illustrative diagrams, the <u>tape format</u> that the universal Turing machine U can use to simulate a given Turing machine M.

ii) Briefly (in about half a page) describe the cycle of instructions that U can use to simulate the execution of a single instruction of M. (U and M are as in part b(i).)

2 In this question, S denotes an <u>arbitrary</u> standard Turing machine, and w an <u>arbitrary</u> word of the standard typewriter alphabet C. S[w] is a Turing machine that overwrites its input with w and then runs S. You may freely use the Turing machine EDIT, which is defined by:

- $f_{EDIT}(code(S)*w) = code(S[w])$

a Consider the problem: 'does S *halt* (either halt & succeed or halt & fail) on input w?' Show that this problem is not solvable by a Turing machine.

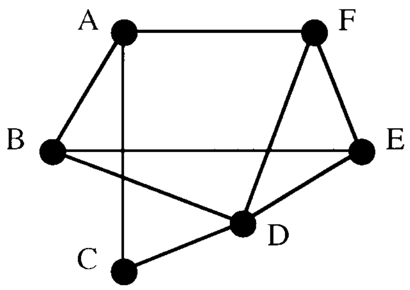
b i) Show that if S halts on input w, then S[w] runs in polynomial time.

ii) Show that if S does not halt on input w, then S[w] does not run in polynomial time.

iii) Deduce that there is no algorithm to decide whether an arbitrary standard Turing machine runs in polynomial time.

The two parts carry, respectively, 55%, 45% of the marks.

3a Starting at node A and using breadth-first <u>or</u> depth-first search, find a spanning tree of the following graph. State which search strategy you use, and give a diagram showing the edges of the tree.



b For $n \geq 1$, how many edges do the following have?

i) A Hamiltonian circuit of a graph with n nodes

ii) A minimal spanning tree of a connected weighted graph with n nodes

iii) A complete graph with n nodes

Briefly justify your answer in each case.

c Let n be a positive integer. Give a formula for the maximum number of edges that a graph with n nodes and two connected components can have. Show all working.

The three parts carry, respectively, 30%, 40%, 30% of the marks.

4 In this question, A and B denote arbitrary yes-no problems. The *complementary problem of A* (in symbols, co-A), is the yes-no problem whose yes-instances are the no-instances of A, and whose no-instances are the yes-instances of A.

a i) Define the classes P and NP of yes-no problems.

ii) Explain what is meant by the statement 'A reduces to B in polynomial time' (in symbols, '$A \leq B$').

b Show that if A is in P then so is co-A.

c Explain why the approach of part b <u>does not prove</u> that if A is in NP then so is co-A.

d Show that if $A \leq B$ then co-$A \leq$ co-B. (Here, '$\leq$' is as in part aii.)

The four parts carry, respectively, 35%, 25%, 25%, 15% of the marks.

End of paper