UNIVERSITY OF LONDON

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1997

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 2nd 1997, 2.00 - 3.30

Answer THREE questions

For admin. only: paper contains 4 questions

1a Explain the difference between a *2-track* and a *2-tape* Turing machine.

*Below, the notation 1^n denotes a string 111...1 of n 1's. The symbol * is used as a delimiter. You may assume that square 0 of each Turing machine tape is implicitly marked.*

b Design a 2-tape Turing machine M with input alphabet {1}, such that if the initial contents of tape 1 are 1^n (for some $n \geq 0$) and the initial contents of tape 2 are 1^m (for some $m>0$), then M halts and succeeds if and only if m divides n without remainder. You may use pseudo-code or a flow-chart diagram; in the latter case you should explain your notation for instructions.

c By modifying M or otherwise, *briefly* explain how you would design a (2-tape) Turing machine M* with input alphabet {1,*}, such that for any $n \geq 0$ and $m>0$, $f_{M^*}(1^n * 1^m) = 1^r$, where r is the remainder when dividing n by m.

The three parts carry, respectively, 20%, 45% and 35% of the marks.

2 In this question:

- C denotes the standard typewriter alphabet.
- If S is a standard Turing machine and w a word of C, S[w] is a standard Turing machine that overwrites its input with w and then runs S.

  So $f_{S[w]}(x)=f_S(w)$ for any word x of C.
- EDIT is a standard Turing machine such that for any standard Turing machine S and word w of C, f_{EDIT}(code(S) *w)=code(S[w]).
- REMA is a standard Turing machine that removes all instances of character 'a' from its input (so, for example, f_{REMA}(abc)=bc).
- U is a (standard) universal Turing machine.

a Define what it means to say that a partial function $g : C^* \to C^*$ is *(Turing-)computable.*

b Let $g : C^* \to C^*$ be a partial function that "tells us whether the output of a standard Turing machine on a given input is 'huxley' or not". That is, for any standard Turing machine S and word w of C,

$$g(\text{code}(S) * w) = \begin{cases} y & \text{if } f_S(w) = \text{huxley} \\ n & \text{otherwise} \end{cases}$$

Show that g is not computable.

c Evaluate:

i) f_U(code(REMA) *deal)

ii) f_U(f_{EDIT}(code(REMA) *stack) *share)

iii) f_U(f_{EDIT}(code(U) *code(REMA) *buy) *sell)

The three parts carry, respectively, 20%, 50% and 30% of the marks.

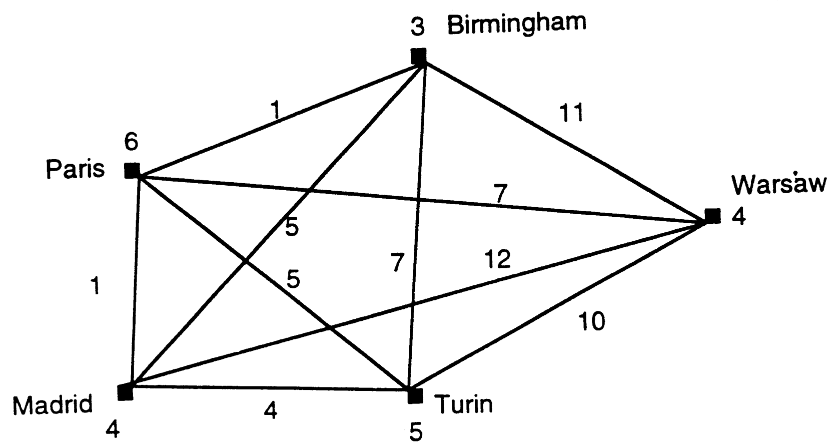
3a Explain the meaning of the terms

(i) *weighted graph*

(ii) *minimal spanning tree (MST) of a connected weighted graph.*

b Briefly describe an algorithm to find a minimal spanning tree of a weighted graph. (Explain its operation but do not prove its correctness).

c An airfreight company wishes to plan a network of routes linking the cities Paris, Birmingham, Milan, Madrid, Warsaw.



The annual cost of running a direct link between two cities as shown in the diagram above is

i) overheads and airport landing fees at each city (written next to the city).
ii) aircraft maintenance and fuel cost for the journey (written on the edge between the cities).

So, for example, running a direct freight link between Paris and Warsaw would cost 6+4+7 = £17 million per year; running a direct link from Paris to Turin would cost 6+5+5 = £16 million per year.

Describe an algorithm which would find the cheapest grid linking these five cities. Use it to find such a grid in the map above. Is this solution unique for the grid?

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

Turn over ...

4a Define a *non-deterministic Turing machine (NDTM)* and briefly describe how a NDTM operates.

b What is meant by saying that a "NDTM N *solves the yes/no problem A*"?

c Let A, B be yes/no problems.

What is meant by

i) $A \in P$

ii) $B \in NP$

iii) A reduces to B in p-time ($A \leq B$)

iv) $B \in NPC$

d (i) d if $A \in NPC$ and $B \in NPC$ does it follow that $B \leq A$? Justify your answer.

(ii) Suppose that it were true that $A \leq B$ for all $A, B \in NP$. Briefly explain why it would follow that $P = NP$. You may assume without proof that P is closed downwards under p-time reduction.

The four parts carry, respectively, 30%, 15%, 30% and 25% of the marks.

End of paper

