IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2006

Corrected Copy

DISCRETE MATHEMATICS AND COMPUTATIONAL COMPLEXITY

Monday, 22 May 2:00 pm

Time allowed: 3:00 hours

There are FIVE questions on this paper.

Answer Question One (29%), Question TWO (29%) and TWO other questions.

Any special instructions for invigilators and information for candidates are on page 1.

Examiners responsible First Marker(s) : G.A. Constantinides
Second Marker(s) : T.J.W. Clarke

NOTATION

The following notation is used throughout this paper:

$\mathbb{R}$: The set of real numbers.

$\mathbb{Z}$: The set of integers.

$\mathbb{C}$: The set of complex numbers.

$\mathbb{N}$: The set of natural numbers.

The Questions

1. **[Compulsory]**

a) Give an example of a countable infinite set. [2]

b) i) Draw the digraph of the relation R on the set $\{a, b, c\}$, where xRy iff x immediately precedes y in the alphabet.

ii) State whether this R is symmetric, transitive, and/or reflexive.

iii) List the elements of R^* for this R.

[10]

c) Consider the function $f : \mathbb{C} \to \mathbb{R}$ given by $f(x) = |x|$.

i) Is this function injective? Justify your answer.

ii) Is this function surjective? Justify your answer.

[5]

d) Let p be the proposition 'I am in an exam', and let q be the proposition 'I am not allowed to talk'. Consider the proposition $q \rightarrow p$. Is it true now? Would it be true if you were revising in the 'silent section' of the library? Briefly justify your answers. [5]

e) Express the proposition 'there is an EE3 student who finds this exam easy' in symbolic logic, given the following predicates. $P(x)$ is the predicate 'x is an EE3 student', $Q(x)$ is the predicate 'x finds this exam easy'. You should take the set of students studying Discrete Mathematics and Computational Complexity as the universe of discourse. [4]

f) "This exam is either difficult or I didn't revise properly. But I'm sure I revised properly, so this exam must be difficult". What is the name given to the rule of inference being applied here? [3]

g) If $f(x)$ and $g(x)$ are both $O(x^2)$, use the results from the lectures to provide a big-O expression for (i) $f(x) + g(x)$ and (ii) $f(x)g(x)$. [4]

h) Briefly define the term 'polynomial-time reduction'. [7]

2. **[Compulsory]**

Let D be the set of all decision problems, and $A(d)$ be the set of all algorithms that solve a particular decision problem $d \in D$. Let $T_d : A(d) \times \mathbb{N} \to \mathbb{R}$ be a function where $T_d(a,n)$ is the worst-case execution time (in seconds) of algorithm a operating on an instance of size n, for a particular machine.

a) A divide-and-conquer recursive algorithm has run-time $f(n)$, when operating on an instance of size n, when n is multiple of an integer $b > 1$. For this algorithm, $f(n) = af(n/b) + cn^d$, where $a \geq 1$, $c > 0$ are real numbers and $d \geq 0$ is an integer. Over what range of values for a, b, c, and d does this algorithm's run time fall into the following categories. Justify your answers in each case. *Hint:* Consider $d = 0$, $d = 1$, $d = 2$, and $d \geq 3$ separately.

i) $O(n)$

ii) $O(n^2)$

iii) $O(2^n)$ [18]

b) Briefly distinguish, in words, between the concepts of a polynomial-time algorithm, and a problem of polynomial complexity. [2]

c) Use symbolic logic and big-O notation to express the predicate $P(d)$, meaning 'problem d is of polynomial complexity' in terms of $A(d)$ and $T_d(a,n)$. [2]

d) Hence express in logic that there are some decision problems unsolvable in polynomial time. Is this proposition true? Briefly justify your answer. [2]

e) Write pseudo-code for a direct recursive implementations of the functions `func1(x)` and `func2(x)`, which return the values of $f_1(x)$ and $f_2(x)$, respectively, defined by the following recurrence relations.

i)

$$f_1(x) = f_1(x-1) + 2f_1(x-2) + 1, \text{ with } f_1(1) = 1. \qquad (2.1)$$

ii)

$$f_2(x) = f_2(\lfloor x/3 \rfloor) + 2f_2(\lfloor x/4 \rfloor) + 1, \text{ with } f_2(1) = 1. \qquad (2.2)$$

[2]

f) Contrast the asymptotic execution times of the two implementations you have written. *Hint:* you may assume that the execution time of `func2(x)` is an increasing function of its argument `x`. [10]

g) Consider the problems $d_i \in D$ with parameter (x,k), to determine whether $f_i(x) > k$ for (2.1)-(2.2). State whether each decision problem is of polynomial computational complexity, justifying your answers. [4]

3. This question relates to a function $f : A \to B$, where A and B are finite sets.

a) Let R denote the range of the function. What relationship exists between R and B? In the case where f is a surjection, what more can be deduced about this relationship? [2]

b) Prove that the cardinality of the range of f is at most the cardinality of its domain. *Hint:* you may assume the pigeonhole principle without proof. [10]

c) Given that f is a surjection, and that A and B are finite sets, what can be deduced about the cardinalities of A and B? Prove this result. [3]

d) Prove that f has an inverse iff it is a bijection. [15]

4. a) Let P be the proposition $p \wedge (q \vee r) \vee \neg(p \vee (q \vee r))$. Replacing all occurences of $(q \vee r)$ by $(q \wedge r)$ gives the proposition $P^* = p \wedge (q \wedge r) \vee \neg(p \vee (q \wedge r))$.

Determine whether each of the following compound propositions is a tautology, and provide a suitable proof of your answer in each case.

i) $q \wedge r \rightarrow q \vee r$.

ii) $P \rightarrow P^*$.

iii) $P^* \rightarrow P$.

[14]

b) You ask two lecturers, G and T, for help, but they try to confuse you. G says 'If T is telling the truth, then so am I'. T says 'at least one of us is lying'. Let p be the proposition 'G is telling the truth'. Let q be the proposition 'T is telling the truth'.

i) Express G's statement using appropriate logical connectives.

ii) Express T's statement using appropriate logical connectives.

iii) By considering possibilities consistent with the truth values of p and q, and the statements made by G and T, deduce who, if anyone, is a liar. Fully explain your answer.

[16]

5. a) Define what is meant by the statements $f(x)$ is $O(g(x))$, $f(x)$ is $\Omega(g(x))$, and $f(x)$ is $\Theta(g(x))$, using appropriate symbolic logic. [3]

b) Prove that $f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_0$ is $\Theta(x^n)$ when $a_i \geq 0$ for $0 \leq i < n$ and $a_n > 0$. [9]

c) Derive the number of each of the following type of operation performed by a call to the procedure `proc1(n)` of Figure 5.1, in terms of n:

i) assignments (including for-loop initialization and incrementation assignments),

ii) multiplication,

iii) incrementation,

iv) comparison.

[9]

d) Given that the above operation types are responsible for the run-time of this code, and that each such operation takes $\Theta(1)$ time, derive a big-Theta expression of the form $\Theta(n^k)$ for the execution time of `proc1`. Hence derive a suitable big-O expression for the execution time of `proc2`, expressing your answer in terms of `c`, an integer constant. [9]

```
proc1(n)
{
  for i = 1 to n {
    t = 2*i
    for j = 1 to t
      a[i][j] = a[i][j]*2;
  }
}

proc2(n)
{
  for i = 1 to c
    proc2( floor(n/2) )
  proc1(n)
}
```

Figure 5.1 Two procedures

E3.20

DISCRETE MATHS & COMPUTATIONAL COMPLEXITY 2006 MODEL ANSWERS

1. a) $\mathbb{Z}$

b) (i) $a \rightarrow b \rightarrow c$

(ii) Not symmetric
Not transitive
Not reflexive

(iii) (a, b) (b, c) (a, c)

c) (i) $f(x) = f(y)$
$|x| = |y|$
Consider $x = 1$, $y = -1$
Then $|x| = |y|$ but $x \neq y$, so not injective.

~~(ii) Let $y = f(x)$~~

(ii) There is no complex # with a negative magnitude $\Rightarrow$ not surjective.

d) True now : $p \rightarrow q$ TRUE $\rightarrow$ TRUE $\therefore$ TRUE
False then : $p \rightarrow q$ ~~FALSE~~ TRUE $\rightarrow$ ~~TRUE~~ FALSE $\therefore$ FALSE

e) $\exists x \, (P(x) \wedge Q(x))$

f) The disjunctive syllogism

g) (i) $f(x) + g(x)$ is $\cancel{O(\max(|f(x)|, |g(x)|))}$ $O(\max(x^2, x^2)) = O(x^2)$

(ii) $f(x)g(x)$ is $O(x^4)$

h) A polynomial time algorithm that transforms ~~one~~ a general instance of one decision problem into an instance of another, such that the answer to the 1st problem is "YES" iff the answer to the 2nd problem is also "YES".

(Q2 FOR CEC3 ONLY)
— HARDER —

2. a) $f(n) = a f(n/b) + c n^d \qquad a \geq 1,\ b > 1,\ c > 0,\ d \geq 0$

This is covered by the Master Theorem:

$a < b^d \Rightarrow O(n^d)$
$a = b^d \Rightarrow O(n^d \log n)$
$a > b^d \Rightarrow O(n^{\log_b a})$

(i) For $O(n)$

$d = 0$: $a = 1 \Rightarrow O(n^0 \log n)$
or $a > 1$ with $\log_b a \leq 1$, i.e. $a \leq b$

So $\underline{a \leq b}$ is sufficient.

$d = 1$: $a < b \Rightarrow O(n)$
$a = b \Rightarrow O(n \log n)$
$a > b$ with $\log_b a \leq 1$ — Not possible

So $\underline{\underline{a \leq b}}$

$d = 2$: $a > b^2$, $a \leq b$ — Not possible
(& $d \geq 3$ similar)

So we require $\underline{\underline{d \leq 1 \text{ with } a \leq b}}$

(ii) $d = 0$: $a = 1 \Rightarrow O(n^0 \log n)$
or $a > 1$ with $a \leq b^2$

So $\underline{\underline{a \leq b^2}}$ is sufficient

$d = 1$: $a < b \Rightarrow O(n)$
or $a = b \Rightarrow O(n \log n)$
or $a > b$ with $a \leq b^2$ — Not possible

So $\underline{\underline{a \leq b}}$

$d = 2$ $a < b^2 \Rightarrow O(n^2)$
$a > b^2$ with $a \leq b^2$ — Not possible

so $\underline{\underline{a < b^2}}$

$d \geq 3$ Not possible.

So we require $d = 0$, $a \leq b^2$
or $d = 1$, $a \leq b$
or $d = 2$, $\underline{\underline{a < b^2}}$

(iii) $f(n)$ is $O(2^n)$ in $\underline{\text{all}}$ cases.

2/10

2. b) A problem is of polynomial complexity if there exists an algorithm capable of solving the problem in polynomial time.

c) $\exists a \in A(d)\ \exists c \in \mathbb{Z}^+\quad T_d(a,n)$ is $O(n^c)$

d) $\exists d\ \neg P(d)$.

This is known to be true for some special problems, e.g. Turing machine halting. So the overall proposition is TRUE.

e)
```
func1(x)
   if x = 1 then
       return 1
   else
       return func1(x-1) + 2*func1(x-2) + 1
   end

func2(x)
   if x = 1 then
       return 1
   else
       return func2(⌊x/3⌋) + 2*func2(⌊x/6⌋) + 1
   end
```

f) Let us first consider the run time of func1 – we can count multiplications, although any other appropriate operation(s) are OK. We will call the number of mults $g_1(n)$.

We have $g_1(1) = 0$

$$g_1(n) = g_1(n-1) + g_1(n-2) + 1, \quad n > 1$$

This is a linear, non-homogeneous recurrence of degree 2. c.f.

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + K$$

We have $c_1 = 1,\ c_2 = 1,\ c_1 + c_2 \neq 1$.

$r^2 - r - 1 = 0$ has distinct roots

$$r = \frac{1 \pm \sqrt{5}}{2}$$

So recurrence has soln. $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$

$\Rightarrow$ EXPONENTIAL TIME.

3/10

2 f) [contd]

Now consider the number of multiplications in func2($\hat{x}$), which we shall denote $g_2(n)$.

$$g_2(1) = 0$$

$$g_2(n) = g_2(n/3) + g_2(n/4) + 1$$

when n is a multiple of 12.

$g_2(n)$ is increasing, so $g_2(n) \leq 2g_2(n/3) + 1$

Using the Master Theorem, $g_2(n)$ is $O(n^{\log_3 2})$

$\Rightarrow$ SUBLINEAR TIME

g) d_1 is of poly complexity, despite the exp nature of func1, as you could compute f1 in a more efficient way.

d_2 is of poly complexity – just use func2.

3. a) $R \subseteq B$.

when f is a surjection, $R = B$.

~~b) $|A| \le |B|$~~
~~$|A| \geq |B|$~~

b) We want to prove that $|f(A)| = |R| \le |A|$.

Assume $|f(A)| > |A|$
Let us define a function $g : f(A) \to A$ by $g(b) = a$ for some $a \in A$ such that $f(a) = b$,
so $f(g(b)) = b$.

g is an injection since $g(b) = g(c)$
$\Rightarrow f(g(b)) = f(g(c)) \Rightarrow b = c$.

But by the pigeonhole principle on g, it cannot be an injection.

c) $|A| \geq |B|$

Since f is surjective $f(A) = B$ so $|f(A)| = |B|$.
However $|f(A)| \le |A|$ from part (b). So
$$|B| = |f(A)| \le |A|.$$

d)
First, prove that if $f : A \to B$ is a bijection, then it has an inverse f^{-1}.

Construct $f^{-1} = \bigcup_{(a,b) \in f} \{(a,b)\} \subseteq B \times A$

here f^{-1} is a relation from B to A.

As f is an injection, no more than one element of A for each element of B.

As f is a surjection, no less than one element of A for each element of B.

So f^{-1} is a function.

Next, prove that if $f : A \to B$ has inverse $f^{-1} : B \to A$, then f is a bijection.

If $f(a) = f(b)$, $f^{-1}(f(a)) = f^{-1}(f(b)) \Rightarrow a = b$, so f is an injection.

Since for any $b \in B$ $a = f^{-1}(b) \in A$, we have $f(a) = f(f^{-1}(b)) = b$. f is a surjection.

5/10

4. a) (i) $q \wedge r \rightarrow q \vee r$

q	r	$q \wedge r$	$q \vee r$	$q \wedge r \rightarrow q \vee r$
F	F	F	F	T
F	T	F	T	T
T	F	F	T	T
T	T	T	T	T

✓ <u>TAUTOLOGY</u>

(ii) & (iii) $p \rightarrow p^*$, $p^* \rightarrow p$ — Neither are tautologies

p	q	r	p	p^*	$p \rightarrow p^*$	$p^* \rightarrow p$
F	F	F	F	T	T	F
F	F	T	T	T	T	T
F	T	F	T	T	T	T
F	T	T	T	F	F	T
T	F	F	F	F	T	T
T	F	T	T	F	F	T
T	T	F	T	F	F	T
T	T	T	T	T	T	T

As a counter-example, consider p, q & r false for $p^* \rightarrow p$, and p false, q & r true for $p \rightarrow p^*$.

b) (i) $q \rightarrow p$

(ii) $\neg p \vee \neg q$

(iii) ~~1st Consider the case that q & r are both telling the truth.~~

	p	q	$q \rightarrow p$	$\neg p \vee \neg q$	
(1)	F	F	T	T	
(2)	F	T	F	T	(*)
(3)	T	F	T	T	
(4)	T	T	T	F	

6/10

4. b) (iii) [contd]

Consider each possibility in the truth table.

(1) Contradiction: G & T both lying but what they say is true.

(2) No contradiction

(3) Contradiction: T is lying but what he says is true.

(4) Contradiction: T is telling the truth but what he says is false.

Only one possibility is consistent: G is a liar, while T tells the truth.

~~4.~~ 5. a) $f(x)$ is $O(g(x)) \equiv \exists c \in R^+ \; \exists k \in R^+ \; \forall x \left((x > k) \rightarrow (|f(x)| \le c|g(x)|) \right)$

$f(x)$ is $\Omega(g(x)) \equiv \exists c \in R^+ \; \exists k \in R^+ \; \forall x \left((x > k) \rightarrow (|f(x)| \ge c|g(x)|) \right)$

$f(x)$ is $\Theta(g(x)) \equiv \exists c_1 \in R^+ \exists c_2 \in R^+ \; \exists k \in R^+$
$\forall x \left((x > k) \rightarrow (c_1|g(x)| \le |f(x)| \le c_2|g(x)|) \right)$

b) Need to prove

(i) $f(x)$ is $O(x^n)$
(ii) $f(x)$ is $\Omega(x^n)$

(i) $|f(x)| \le |a_n x^n| + |a_{n-1} x^{n-1}| + \dots + |a_1 x| + |a_0|$
$= |x^n|(|a_n| + |a_{n-1}|/|x| + \dots + |a_0|/|x^n|)$
$\le |x^n| (|a_n| + |a_{n-1}| + \dots + |a_0|)$ for $x > 1$

So with $c = |a_n| + \dots + |a_0| (>0)$ and $k = 1$
$f(x)$ is $O(x^n)$

(ii) $|f(x)| = |a_n x^n + \dots + a_0|$
$\ge |a_n x^n|$ for $x \ge 0$
$= |x^n| a_n$ since $a_n > 0$

So with $c = a_n$ and $k = 1$ (say)
$f(x)$ is $\Omega(x^n)$

Thus $\underline{f(x) \text{ is } \Theta(x^n)}$

8/10

4.5. 5 c) We can derive a $\Theta(\cdot)$ expression for all the operations - first count # ops

~~(i)~~ There are n iterations of the outer loop & $\sum_{i=1}^{n} 2i = 2\sum_{i=1}^{n} = n(n+1)$ iterations of the inner loop.

(i) assignments of a[i][j] : $n(n+1)$
assignments due to init of i : 1
" j : n
assignments due to increment of i : n
" j : $n(n+1)$
assignments due to t : n

TOTAL = $2n(n+1) + 3n + 1$
~~$= 2n(n+2)+1$~~
$= n(2n+5)+1$

(ii) mult of i : n
mult of a : $n(n+1)$

TOTAL = $n(n+2)$

(iii) increment of i : n
" j : $n(n+1)$

TOTAL = $n(n+2)$

(iv) comparison of i : $n+1$
j : $n(n+1) + n = n(n+2)$

TOTAL = $n(n+2) + n + 1$
$= n(n+3) + 1$

d) Total run time is

$$\Theta(1)\left[n(2n+5)+1\right] + \Theta(1)\left[n(n+2)\right] + \Theta(1)\left[n(n+2)\right] + \Theta(1)\left[n(n+3)+1\right] = \Theta(n^2)$$

9/10

~~4.4~~d) [cntd]

5 Denote the exec-time of proc(n) by g(n). Then

$$g(n) \le c\,g(n/2) + dn^2$$

This is a D&C recurrence.

If $c < 2^2 = 4$ run time is $O(n^2)$

If $c = 2^2 = 4$ run time is $O(n^2 \log n)$

If ~~c2~~ $= c > 2^2 = 4$ run time is ~~$O(n\log$~~ $O(n^{\log_2 c})$

10/10