883

Ollscoil na hÉireann, Gaillimh
National University of Ireland, Galway

Semester II Examinations, 2004/2005

Exam Code(s)	0MS1, 1MF1
Exam(s)	MS1 M.Sc.Degree
	MF1 M.Sc.Degree(Software Design & Development)
Module Code(s)	CT518
Module(s)	Algorithms And Logical Methods
Paper No.	1
Repeat Paper	Special Paper
External Examiner(s)	Professor D. Bell
Internal Examiner(s)	Professor G. Lyons
	Dr. M. Mc Gettrick
Instructions	Answer 4 questions. All questions will be marked equally.
Duration	2hrs
No. of Answer Books	1
Requirements	
Handout	
MCQ	
Statistical Tables	
Graph Paper	
Log Graph Paper	
Other Material	
No. of Pages	
Department(s)	

1. (a) Using each of the following methods, write down (step by step) the position of each letter in the word "harlem" when sorted using

   (i) selection sort

   (ii) merge sort

   (b) State (in "Big Oh" notation) the worst case complexity of both algorithms in part (a). For merge sort, explain why you would expect a factor of approximately 2 between worst case and best case performance.

2. (a) Write the pseudocode for an algorithm which calculates x^n, given as input x and n where n is a positive integer, using

   (i) Iteration

   (ii) Recursion

   (b) Write the pseudocode for a recursive algorithm which checks whether or not a word is a palindrome (a palindrome is a word which reads the same written in either direction, e.g. "deed", "abba").

3. Consider the following two algorithms:

(I)

```
BEGIN {algorithm to calculate x^n}
IN(x,n)
y ← 1
WHILE (n>0) DO
   IF (n mod 2 = 0) THEN
      BEGIN x ← x*x; n ← n/2; END
   ELSE
      BEGIN y ← y*x; n ← n-1; END
OUT(y)
END
```

(II)

```
BEGIN {algorithm to calculate x^n}
IN(x,n)
OUT(POW(x,n))
END

POW(i,j)
IF (j=1) THEN POW ← i
ELSE POW ← POW(i,FLOOR(j/2))*
     POW(i,FLOOR((j+1)/2))
```

(note that p mod q returns the remainder after dividing p by q, e.g. 7 mod 2 is 1)

   (a) For large n, which algorithm should be faster?

   (b) For algorithm (I), determine the number of times the while loop is executed for the two inputs $n = 64$ and $n = 63$. Hence calculate the worst case and best case complexity of algorithm (I).

   (c) Say that for certain fixed values of x and n both algorithms run in equal time t_1 seconds. Estimate (in terms of t_1) the time for each algorithm if we keep x constant but double n.

4. (a) Write the pseudocode for an $\mathbf{O}(\log_2(n))$ algorithm to calculate the nth Fibonacci Number.

   (b) Explain by giving an example each of the following:

   (i) Idempotency of $\wedge$

   (ii) The Law of Double Negation

   (iii) Distribution Law of $\vee$ over $\wedge$

   (iv) Associativity of $\vee$

5. (a) Use truth tables to determine whether each of the following well formed formulae (wff) are tautologies, contradictions, or neither.

   (i) $\neg(A \vee B) \leftrightarrow \neg A \wedge \neg B$

   (ii) $A \vee (B \wedge C) \leftrightarrow (A \vee B) \wedge (A \vee C)$

   (iii) $\neg(A \wedge B) \rightarrow (A \vee \neg B)$

   (b) Given the predicates

   E(x) : "x is an Eagle"
   M(x) : "x is a Mouse"
   A(x,y) : "x Attacks y"

   write statements in First Order Predicate Calculus to express each of the following.

   (i) All Eagles Attack Mice.

   (ii) Some Eagles Attack all Mice.

   (iii) Only Eagles Attack Mice.