OLLSCOIL NA hÉIREANN GAILLIMH

NATIONAL UNIVERSITY OF IRELAND GALWAY

---

SUMMER EXAMINATIONS 2002

---

Higher Diploma in Applied Science (Software Design and Development)
Second University Examination in Electronic Engineering

**ALGORITHMICS AND LOGICAL METHODS (CT853)**

Professor D. Bell
Professor G. Lyons
Dr. M. Mc Gettrick

Time allowed: *two* hours.
Attempt *three* questions.

1. (a) Using each of the following methods, write down (step by step) the position of each letter in the word "gdansk" when sorted alphabetically using
   (i) insertion sort
   (ii) quicksort
   (b) State the worst case and best case complexities of the algorithms in part (a). Under best case, how many comparisons does each make to sort 64 items?

2. 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)
y ← 1
WHILE (n>0) DO
   BEGIN y ← x*y; n ← n-1; END
OUT(y)
END
```

(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 $n$.

3. (a) Suppose we define $s_0 = 0, s_1 = 1, s_2 = 2$ and $s_n = s_{n-1} + s_{n-2} + s_{n-3}$ for $n > 2$, so that we have the sequence 0,1,2,3,6,11,20,37,.... Write an algorithm to calculate $s_n$ using

   (i) Iteration

   (ii) Recursion

   Which of these algorithms is more efficient?

   (b) Explain the Divide and Conquer strategy, and give an example of an algorithm using this strategy.

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

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

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

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

   (b) Given the predicates

   H(x) : "x is an Hawk"
   F(x) : "x is a Fish"
   E(p,q) : "p Eats q"

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

   (i) All Fish do not Eat Hawks

   (ii) Some Hawks Eat some Fish.

   (iii) Some Hawks Eat no Fish.