THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2003
Campus: City

COMPUTER SCIENCE

Algorithmics

(Time allowed: ONE hour)

NOTE:

- Attempt all **6** questions.
- Please write precisely and clearly. Put the answers in the boxes below the questions. You may continue your answers onto the "overflow" pages provided at the back of the book if necessary.
- Marks for each question are shown below and just before each answer box.
- Use of calculators is NOT permitted (nor would it be useful).
- The test contributes 15% to your final grade for the course.

SURNAME:

FORENAME(S):

STUDENT ID:

Question #:	1	2	3	4	5	6	Total
Possible marks:	10	10	10	10	10	10	60
Awarded marks:							

Surname: ____________________

Forename(s): ____________________

1. (a) Complete the table by answering YES or NO to each question.

$f(n)$	$g(n)$	$f(n) \in O(g(n))$?	$f(n) \in \Omega(g(n))$?	$f(n) \in \Theta(g(n))$?
n^n	$n!$			
$\lg n$	$\lg(\lg n)$			
n^{100}	$(0.01)^n$			
$\log n$	$\lg(3n)$			

[6 marks]

(b) Prove formally that for any functions f and g that take positive values on the natural numbers, if $f \in \Theta(g)$ then $g \in \Theta(f)$. [4 marks]

Surname: ______________________________

Forename(s): ______________________________

2. Consider the mergesort recurrence

$$t(n) = \begin{cases} 0 & \text{if } n = 1; \\ t(\lceil n/2 \rceil) + t(\lfloor n/2 \rfloor) + n - 1 & \text{if } n > 1. \end{cases}$$

(a) What are the values of $t(3), t(4), t(5)$? [3 marks]

(b) Solve the recurrence exactly assuming that n is a power of 2. Express your answer in terms of a simple formula involving n only. [5 marks]

(c) Solve the recurrence asymptotically for general n. Justify your answer. [2 marks]

Surname: ______________________________

Forename(s): ______________________________

3. Recall that an *Egyptian fraction* is a rational number of the form $1/n$, where n is a positive integer. Consider the problem of writing a positive rational number m/n as a sum of distinct Egyptian fractions.

   (a) Show why the obvious greedy algorithm always finds a solution to the problem. You must show why the algorithm terminates. [7 marks]

   (b) Carry out the algorithm when $m = 14, n = 15$. [3 marks]

Surname: ______________________________

Forename(s): ______________________________

4. (a) True or false: the obvious greedy algorithm gives the optimal solution to the knapsack problem if we allow fractional objects. [2 marks]

(b) True or false: the obvious greedy algorithm gives the optimal solution to the integer knapsack problem. [2 marks]

(c) A particular divide-and-conquer algorithm divides a problem instance into 4 subinstances each roughly equal to one-third of the size of the original in size, and has overhead proportional to the size of the problem. What is its asymptotic running time? [2 marks]

(d) Multiply 23 by 15 using the "Russian peasant" multiplication algorithm. [2 marks]

(e) Explain why mergesort and quicksort are at opposite extremes in the class of divide and conquer algorithms. [2 marks]

Surname: ______________________________

Forename(s): ______________________________

5. (a) Briefly explain how the dynamic program given in the class/textbook works for the *Making Change* problem. [3 marks]

(b) Show the steps needed to solve the following instance of *Making Change*. Suppose we go to a party where we are given a supply of tokens of values $\{1, 4, 7\}$ and want to find the smallest set of tokens that sum to 20. [4 marks]

(c) Explain how we use/modify the *Making Change* dynamic program to find all possible combinations of tokens of minimum size. [3 marks]

Surname: ______________________________

Forename(s): ______________________________

6. We have the following **Digit Matching** problem. Input is two equal-length strings $A = a_1, a_2, \ldots, a_n$ and $B = b_1, b_2, \ldots, b_m$ of character digits $\{0, 1, \ldots 9\}$. Given a conversion cost matrix $C[0 \ldots 9, 0 \ldots 9]$, where $C[i][j]$ indicates the cost of exchanging digit i with digit j and $C[i][i]$ indicates the cost of inserting or deleting digit i, we want to find the minimum cost of converting both A and B to a common string $D = d_1, d_2, \ldots, d_l$. We want to solve this problem using dynamic programming.

   (a) Explain how we can define subproblems using substrings of the input A and B. [3 marks]

   (b) Show how you can solve larger instances when all of the solutions to smaller instances are known. [7 marks]

Surname: ____________________

Forename(s): ____________________

Additional work pages