THE UNIVERSITY OF AUCKLAND

FIRST SEMESTER, 2005
Campus: Tamaki

COMPUTER SCIENCE

Algorithms and Data Structures

(Time allowed: ONE hour)

NOTE: Attempt *all* questions. Write answers in the boxes below the questions. You may continue your answers onto the "overflow" page provided at the back if necessary. Marks for each question are shown below the answer box. The use of calculators is NOT permitted.

SURNAME:

FORENAME(S):

STUDENT ID, UPI:

Section:	A	B	Total
Possible marks:	40	20	60
Awarded marks:			

QUESTION/ANSWER SHEETS FOLLOW

Surname: _____

Forename(s): _____

A. (Algorithm Analysis)

1. The expressions below give the processing time $T(n)$ of an algorithm for solving a problem of size n. Give dominant terms and Big-Oh complexities of these algorithms.

   Hint: The dominant term has the steepest increase in n.

Expression	Dominant term	$O(\ldots)$
$50n^2 + n^{1.5} + \sqrt{n}$	$50n^2$	$O(n^2)$
$(n + 2n^2)(5n \log_2 n + n^3)$	$2n^5$	$O(n^5)$
$n^2 \log_2 n + n(\log_2 n)^2 + n \cdot \log_2 n^2$	$n^2 \log_2 n$	$O(n^2 \log_2 n)$
$n(\log_4 n) + n(\log_2 n) + n(\log_{10} n)$	$n(\log_2 n)$	$O(n(\log_2 n))$
$2^6 + n^6 + 6^n$	6^n	$O(6^n)$
$1000n + 0.0005n^2$	$0.0005n^2$	$O(n^2)$

[6 marks]

2. Prove that $T(n) = a_0 + a_1 n + a_2 n^2 + a_3 n^3$ is $O(n^3)$ where a_0, a_1, a_2 and a_3 are positive real constants.

   Hint: $f(n)$ is $O(g(n))$ if there exists a real $c > 0$ and an integer $n_0 > 0$ such that $f(n) \leq c \cdot g(n)$, for all $n \geq n_0$.

It is obvious that $T(n) \leq a_0 n^3 + a_1 n^3 + a_2 n^3 + a_3 n^3 = (a_0 + a_1 + a_2 + a_3) n^3$. If $n > 0$ then $T(n) \leq cn^3$ where $c = a_0 + a_1 + a_2 + a_3$.

[5 marks]

Surname: _____

Forename(s): _____

3. Consider the recurrence relation $t(n) = 4 \cdot t(\lceil \frac{n}{4} \rceil) + n$, with $t(1) = 0$.

   (a) Calculate $t(n)$ for $n = 2, 3, 4$.

   $t(2) = 2$, $t(3) = 3$, $t(4) = 4$

   [3 marks]

   (b) Assume that $n = 4^k$ and derive a closed formula for $t(n)$ by telescoping. What is the Big-Oh complexity?

   We replace n by 4^k and we get $t(4^k) = 4t(4^{k-1}) + 4^k$. We divide the equation by 4^k and apply telescoping:

   $$\begin{array}{lcl} \frac{t(4^k)}{4^k} & = & \frac{t(4^{k-1})}{4^{k-1}} + 1 \\ \frac{t(4^{k-1})}{4^{k-1}} & = & \frac{t(4^{k-2})}{4^{k-2}} + 1 \\ \cdots & \cdots & \cdots \\ \frac{t(4)}{4} & = & \frac{t(1)}{1} + 1 \end{array}$$

   It follows that $t(4^k) = k \cdot 4^k = n \cdot log_4 n$. The Big-Oh complexity is $O(n \cdot log_2 n)$.

   [4 marks]

   (c) Apply the "divide and conquer theorem" to obtain the Big-Oh complexity of $t(n)$. Hint: The recurrence relation $T(n) = aT(n/b) + cn^k$, where $a \geq 1$, $b \geq 2$, $c > 0$, and $k \geq 0$ are constants, has the following solution:

   $$T(n) = \begin{cases} O(n^{\log_b a}), & \text{if } b^k < a \\ O(n^k \log n), & \text{if } b^k = a \\ O(n^k) \quad \text{otherwise.} \end{cases}$$

   In our example we have that $a = 4$, $b = 4$ and $k = 1$. This means that $b^k = a$ and it follows that the Big-Oh complexity of $t(n)$ is $O(n \cdot log_2 n)$.

   [1 mark]

Surname: _____

Forename(s): _____

4. For the following three code segments, find the simplest and slowest-growing function g such that the time complexity of the segment is $O(g)$ in the worst case. In all three segments, S represents a sequence of statements in which there are no loops depending on n.

   (a) for (int $i = 0; i \cdot i < n; i++$)
   S

   $O(\sqrt{n})$

   (b) for (int $i = 0; \sqrt{i} < n; i++$)
   S

   $O(n^2)$

   (c) int $k = 1$;
   for (int $i = 0; i < n; i++$)
   $k* = 2$;
   for (int $i = 0; i < k; i++$)
   S

   $O(2^n)$

[6 marks]

5. The running time of algorithm A and B is given by $f(n) = (\log_2 n)^2 + n!$ and $g(n) = 2^n + n^2$, respectively. For which input sizes (positive integers) is algorithm B faster than algorithm A, assuming that both algorithms run on identical hardware?

n	1	2	3	4	5
A	1	3	< 8	28	< 125
B	3	8	17	32	57

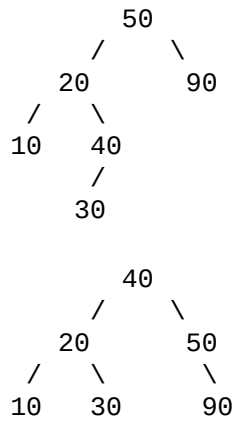
For $n \geq 5$ algorithm B is better than algorithm A.

[5 marks]

Surname: _____

Forename(s): _____

6. Convert the following binary search tree into an AVL-tree of height 2:



[5 marks]

7. Convert the following array $[50, 30, 35, 40, 32, 25, 18, 31, 2]$ into a minimum heap (tree or array is fine).

~~[2,18,25,30,32,35,50,31,40]~~ [2 30 18 31 32 25 35 50 40]

[5 marks]

Surname: _____

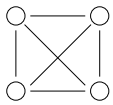
Forename(s): _____

B. (Graph Algorithms)

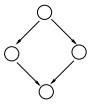
8. For each of the following, draw an example or explain why no such example exists:

[5 marks]

(a) A graph with 4 vertices where the distance between every pair of vertices is 1.



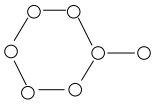
(b) A DAG whose underlying graph is connected but not a tree.



(c) A connected graph with 100 vertices and 98 edges.

IMPOSSIBLE - $e \geq n - 1 = 99$ for connected graphs

(d) A graph with diameter 4 and girth 6.



(e) A digraph with 4 nodes and 14 arcs.

IMPOSSIBLE - $n(n-1) = (4)(3) = 12$ is maximum

Surname: _____

Forename(s): _____

9. Consider the digraph G with nodes $0, \ldots, 6$ whose adjacency matrix representation is given below.

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

(a) Write down the adjacency lists representation of G.

[1 mark]

0:	1	4	5
1:	0	3	
2:	0	6	
3:	0	5	
4:	5		
5:			
6:	5		

(b) Suppose that BFS is run on G, with the rule that whenever there is a choice of node to visit, the one with smallest label is chosen. List all tree arcs, forward arcs, back arcs, and cross arcs of G. [4 marks]

Tree: $(0,1)(0,4)(0,5)(1,3)(2,6)$. Forward: none. Back: $(1,0)(3,0)$. Cross: $(2,0)(3,5)(4,5)(6,5)$.

Surname: _____

Forename(s): _____

10. (a) Perform the DFS algorithm on the digraph G whose adjacency lists representation is given below. List the seen and done times for each node.

0:	2		
1:	0		
2:	0	1	
3:	4	5	6
4:	5		
5:	3	4	6
6:	1	2	

Node:	0	1	2	3	4	5	6
Seen:	0	2	1	6	7	8	9
Done:	5	3	4	13	12	11	10

(b) List the strong components of G. Show your work.

Answer is $\{0, 1, 2\}, \{3, 4, 5\}, \{6\}$. Obtained by running DFS on G_r choosing root as white node with latest done time above. Roots are $3, 6, 0$ in that order.

[5 marks]

Surname: _____

Forename(s): _____

11. Answer each question TRUE or FALSE. Correct answers receive 1 mark; incorrect ones receive -0.5 marks.

    (a) If BFS is run on a graph, and there is a cross edge, then the graph has a cycle.

    TRUE

    (b) If a graph has a cycle, then when BFS is run, there will be a cross edge.

    TRUE

    (c) When DFS is run on a graph, every edge is a cross edge or a tree edge.

    FALSE

    (d) If DFS is run on a digraph and (v, w) is a cross arc, then v is seen before w.

    FALSE

    (e) If DFS is run on a digraph and v is visited before w, then w finishes processing before v.

    FALSE

[5 marks]

Surname: _____

Forename(s): _____

Additional work page