THE UNIVERSITY OF AUCKLAND

FIRST SEMESTER, 2007
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:

Section:	A	B	Total
Possible marks:	30	20	50
Awarded marks:			

QUESTION/ANSWER SHEETS FOLLOW

Surname: ______________________________

Forename(s): ______________________________

A. (Algorithm Analysis)

1. Answer each question TRUE or FALSE. Correct answers receive 1 mark; incorrect ones receive -0.5 marks. [8 marks]

(a) The dominant term of the function $f(n) = 0.00001n^2 + 6000n + 3$ is the term $6000n$.

FALSE

(b) The function $f(n) = 12n^2 + 600n + 3000$ is $\Theta(n^2)$.

TRUE

(c) The function $f(n) = 12n^2 + 600n + 3000$ is $O(n^3)$.

TRUE

(d) If the function $f(n)$ is $O(n^2)$, then $f(n)$ is $O(n)$.

FALSE

(e) If the function $f(n)$ is $O(n)$ and the function $g(n)$ is $O(n)$, then $h(n) = f(n) \cdot g(n)$ is $\Omega(n)$.

FALSE

(f) If the function $f(n)$ is $\Omega(n)$, then $f(n)$ is $O(n)$ and $f(n)$ is $\Theta(n)$.

FALSE

(g) To sort an integer array of size N, the sorting algorithm `mergesort` needs an additional temporary array of size N.

TRUE

(h) The average time complexity bound for sorting an integer array of size N by either `quicksort` or `heapsort` algorithm is $O(N \log N)$.

TRUE

Surname: ____________________

Forename(s): ____________________

2. Prove directly from the definition of “Big Theta” notation that $f(n) = 2n^3 + 40n^2 + 20n + 200$ is $\Theta(n^3)$ [6 marks]

We have to show that there exist two positive constants c_1 and c_2 and a positive integer n_0 such that

$$c_1 n^3 \leq 2n^3 + 40n^2 + 20n + 200 \leq c_2 n^3 \text{ forall } n \geq n_0$$

Because all the addends in $f(n)$ are positive and $n^2 \leq n^3$, $n \leq n^3$, and $1 \leq n^3$ for $n \geq 1$, the above inequalities hold for the values $c_1 = 2$, $c_2 = 262$, and $n_0 = 1$.

Solutions with other valid values of c_1, c_2, and n_0 are equally correct.

3. Prove directly from the definition of “Big Oh” notation that $f(n) = 100n^{1.5} + 5n + 20$ is not $O(n)$ [4 marks]

We have to show that a positive constant c and an integer n_0, such that

$$100n^{1.5} + 5n + 20 \leq cn \text{ for all } n \geq n_0$$

or $c \geq 100n^{0.5} + 5 + \frac{20}{n}$, does not exist. But this is just the case because the right-hand side of the latter inequality tends to the infinity if n tends to the infinity.

Surname: ____________________

Forename(s): ____________________

4. Assuming $n = 3^m$; $m > 0$, and $T(1) = 0$, guess the solution of the recurrence $T(n) = 3T\left(\frac{n}{3}\right)+2$ from a sequence of values $T(1)$, $T(3)$, $T(9)$, $T(27)$, $\ldots$, and prove this solution by mathematical induction. [6 marks]

Starting from $T(3^0 = 1) = 0$, the successive values of the function under consideration are as follows: $T(3^1 = 3) = 3 \times 0 + 2 = 2$; $T(3^2 = 9) = 3 \times 2 + 2 = 8$; $T(3^3 = 27) = 3 \times 8 + 2 = 26$, etc. Therefore, one may guess that $T(3^m) = 3^m - 1$.

The base case holds: $T(1) = 1 - 1 = 0$.

By the inductive hypothesis, $T(3^{m-1}) = 3^{m-1} - 1$. Then $T(3^m) = 3 \times (3^{m-1} - 1) + 2 = 3^m - 3 + 2 = 3^m - 1$, i.e. just what should be proven.

5. Solve the recurrence $T(n) = \frac{1}{n}\left(T(0) + T(1) + \ldots + T(n-1)\right)+n$; $T(0) = 0$, by "telescoping". [6 marks]

Hint: A useful formula: $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n} \to \ln n$ if $n \to \infty$.

The recurrence can be represented as $nT(n) = T(0) + T(1) + \ldots + T(n-1) + n^2$ so that $(n-1)T(n-1) = T(0) + T(1) + \ldots + T(n-2) + (n-1)^2$. Subtracting the second relationship from the first relationship gives $nT(n) - (n-1)T(n-1) = T(n-1) + 2n - 1$, or $T(n) = T(n-1) + 2 - \frac{1}{n}$. By telescoping:

$$
\begin{array}{lcllclcl}
T(n) & = & T(n-1) & + & 2 & - & \frac{1}{n} \\
T(n-1) & = & T(n-2) & + & 2 & - & \frac{1}{n-1} \\
\ldots & & \ldots & & \ldots & & \ldots \\
T(2) & = & T(1) & + & 2 & - & \frac{1}{2} \\
T(1) & = & T(0) & + & 2 & - & 1
\end{array}
$$

Therefore, after summing the left-hand and right-hand parts of the equalities and reducing the same terms, we obtain:

$$T(n) = 0 + 2n - \left(1 + \frac{1}{2} + \ldots + \frac{1}{n}\right) = 2n - \ln n$$

Surname: ______________________________

Forename(s): ______________________________

B. (Graph Algorithms)

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

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

Suppose BFS is run on G, with the usual 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: $(01), (02), (06), (25), (34)$. Forward: none. Back: $(60), (4, 3)$. Cross: $(12), (45), (56)$.

7. (a) Perform the DFS algorithm on the digraph G whose adjacency lists representation is given below. Use the usual rule that whenever there is a choice of node to visit, the one with smallest label is chosen. List the seen and done times for each node. [4 marks]

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

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

Surname: ______________________________

Forename(s): ______________________________

(b) If G is acyclic, write down a topological order for G. Otherwise, write down a cycle in G. Explain your reasoning. [2 marks]

No back arcs detected in DFS, so acyclic. Topological order is given by reverse done times, namely 6, 2, 1, 0, 3, 4, 5.

8. Answer each question TRUE or FALSE. Correct answers receive 1 mark; incorrect ones receive -0.5 marks. [10 marks]

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

FALSE

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

TRUE

Surname: ______________________

Forename(s): ______________________

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

FALSE

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

FALSE

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

TRUE

(f) Floyd's algorithm solves the all-pairs shortest path problem in $O(n^2)$ time.

FALSE

(g) If DFS is run on a DAG and we list the nodes in increasing order of "seen" time, this gives a topological order of G.

FALSE

(h) If DFS is run on a graph, then there are no cross edges.

TRUE

(i) There exists a connected graph with 6 vertices and 4 edges.

FALSE

(j) If the digraph G has maximum outdegree 12, then we can represent it using $O(n)$ storage space.

TRUE

Surname: ________________________________

Forename(s): ________________________________

Additional work page