THE UNIVERSITY OF AUCKLAND

TEST FOR BSc BScHons ETC 2004

COMPUTER SCIENCE: COMPSCI.220.S1T
Algorithms and Data Structures

(Time allowed: ONE hour)

Family Name (*please, print clearly*): ....

Given Name(s): ....

Degree (BSc, BSc(Hons), etc.): ....

Student Identification Number: ....

Signature: ....

Attempt *all* questions. 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.

Section:	A	B	Total
Possible marks:	25	25	50
Awarded marks:			

QUESTION/ANSWER SHEETS FOLLOW

Family Name: ....

Given Names: ....

A. (Algorithm Analysis)

1. Assume that each of the expressions below gives the processing time $T(n)$ spent by an algorithm for solving a problem of size n. Select the dominant term and specify the Big-Oh complexity of each algorithm. **[6 marks]**

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

   ANSWER:

Expression	Dominant term	$O(\ldots)$
$50n + n^{1.5} + 0.01n^{1.75}$	$0.01n^{1.75}$	$O(n^{1.75})$
$300 + 5n^{0.1} + \log_2 n$	$5n^{0.1}$	$O(n^{0.1})$
$3n + 5n \log_2 n + n^{1.5}$	$n^{1.5}$	$O(n^{1.5})$
$n \log_4 n + n(\log_2 n)^2$	$n(\log_2 n)^2$	$O(n(\log n)^2)$
$2^6 + n^6 + 6^n$	6^n	$O(6^n)$
$(\log_4 n)^2 + (\log_2 \log_2 n)^2$	$(\log_4 n)^2$	$O(\log n)^2)$

2. Prove that $f(n) = 30n^{1.5} + 0.03n^{2.5}$ is $O(n^{2.5})$ **[2 marks]**

   Hint: $f(n)$ is $O(g(n))$ if there exist a positive factor $c > 0$ and a positive threshold $n_0 > 0$ such that $f(n) \leq c \cdot g(n)$ for all $n \geq n_0$.

   ANSWER:

$f(n) = 30n^{1.5} + 0.03n^{2.5} = n^{2.5}\left(\frac{30}{n} + 0.03\right)$, or

$$f(n) \leq (30 + 0.03)n^{1.5} = 30.03n^{1.5} \text{ if } n \geq 1$$

More general proof:

$$f(n) \leq \left(\frac{30}{n_0^{1.5}} + 0.03\right) n^{1.5} \text{ if } n \geq n_0$$

Family Name: ....

Given Names: ....

3. Show the basic steps for solving the recurrence $T(n) = \frac{1}{n}(T(0) + T(1) + \ldots + T(n-1)) + c$; $T(0) = 0$, and specify the Big Oh complexity of $T(n)$. **[3 marks]**

   Hint: A useful formula: $H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n} \approx \ln n + 0.577$

   ANSWER:

$$\begin{array}{lcl} nT(n) & = & T(0) + T(1) + \ldots + T(n-2) + T(n-1) + cn \\ (n-1)T(n) & = & T(0) + T(1) + \ldots + T(n-2) + c(n-1) \end{array}$$

   or $nT(n) - (n-1)T(n-1) = T(n-1) + c$,
   or $T(n) = T(n-1) + \frac{c}{n}$.
   By telescoping,

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

   or $T(n) = c\left(1 + \frac{1}{2} + \ldots + \frac{1}{n}\right)$,
   or $T(n) = c \ln n - 0.577c$, *that is,* $T(n)$ *is* $O(\log n)$.

4. Two students are playing the game "How old am I?" where one player has to find the age (in years) of the second player by asking questions of the form: "Are you less than x years old?" The goal is to ask as few questions as possible, assuming that nobody cheats and the players are at most 31 years old.

   (a) Design and describe a good strategy for this game by specifying how to choose x in each successive question. **[3 marks]**

ANSWER:
The good strategy follows ideas of binary search: use in Question 1 $x = (31+1)/2 = 16$*, and depending on the answer, use the like mid-range* x *either in the left or the right half-range of the number values. For instance: let the desired number* k *is within the range* $[\mathrm{lo}, \mathrm{hi}]$ *such that* $1 \le \mathrm{lo} \le k \le \mathrm{hi} \le n$*. Then before asking the questions,* $\mathrm{lo} = 1$ *and* $\mathrm{hi} = n$*. The player finds the value* k *when the range is reduced to* $\mathrm{lo} = \mathrm{hi} = k$*. At each step* $t = 1, 2, \ldots$ *of the good questioning strategy, the question involves the current value* $x = \frac{\mathrm{lo}+\mathrm{hi}}{2}$*, and if the answer to the question is "yes", then to reduce the range, the right range border is set to mathrmhi* $= x$*, otherwise the left border is set to* $\mathrm{lo} = x$*. The process terminates when* $\mathrm{lo} = \mathrm{hi}$ *so that the desired answer is* $k = \mathrm{lo}(\equiv \mathrm{hi})$

(b) Give and solve the basic recurrence for the number of questions $T(n)$ in the proposed strategy. **[2 marks]**

ANSWER:
$T(n) = T(n/2) + 1$; $T(1) = 0$*, and the solution by telescoping:*

$$\begin{array}{lcl} T(2^m) & = & T(2^{m-1}) + 1 \\ T(2^{m-1}) & = & T(2^{m-2}) + 1 \\ \cdots & \cdots & \cdots \\ T(2) & = & T(1) + 1 \end{array}$$

so that $T(2^m) = m$*, or* $T(n) = \log_2 n$

(c) Specify the maximum number of questions to find the player's age. **[1 mark]**

ANSWER:
5

Family Name: ....

Given Names: ....

5. Show steps of adding one more node with the key 45 to the heap [50, 40, 35, 30, 32, 24, 18, 31, 20, 8] of size 10 and restoring the heap. **[3 marks]**

ANSWER:
To check whether you are careful or not, the given array violated the heap property for the pair 30 – 31. Those students who have found this and restored the heap, got 0.5 mark as a bonus.

The key insertion steps (after the heap was restored by swapping 30 and 31):

positions:

1	2	3	4	5	6	7	8	9	10	11
50	40	35	30	32	24	18	31	20	8	
										45
				45						32
	45			40						
50	45	35	31	40	24	18	30	20	8	32

6. Show steps of deleting the root node from the heap of size 11 obtained in Question 5 and restoring the heap. **[3 marks]**

ANSWER:

positions:

1	2	3	4	5	6	7	8	9	10	11
50	45	35	31	40	24	18	30	20	8	32
32	45	35	31	40	24	18	30	20	8	
45	32									
	40			32						
45	40	35	31	32	24	18	30	20	8	

Family Name: . . . .

Given Names: . . . .

7. Draw a binary tree representation of the heap of size 10 obtained in Question 6. **[2 marks]**

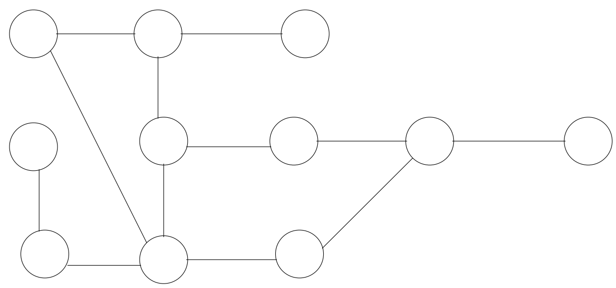
ANSWER:

```
          45
         /    \
      40        35
     /   \      /  \
   31      32  24    18
  / \     /
30   20  8
```

Family Name: . . . .

Given Names: . . . .

B. (Graph Algorithms)



1. For the graph shown

   (a) What is the girth?

   ANSWER:
   4

   (b) What is the diameter?

   ANSWER:
   5

   (c) What is the size?

   ANSWER:
   12

   (d) What is the order?

   ANSWER:
   11

CONTINUED

Family Name: . . . .

Given Names: . . . .

(e) What is the maximum degree of a vertex?

ANSWER:
4

Family Name: ....

Given Names: ....

2. Find the strong components of the digraph whose adjacency lists representation is given below. Show your working.

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

Family Name: ....

Given Names: ....

ANSWER:

The strong components are $\{1,2\}, \{3,4,5\}, \{6\}$. *One can determine this by first peforming DFS on the digraph. Using the standard convention for making arbitrary choices (choose lowest numbered node) this gives postorder* $1, 2, 0, 6, 5, 4, 3$. *Running DFS on the reverse digraph, choosing roots according to reverse postorder gives* 3 *trees with nodes as above (roots are* $3, 6, 1$*).*

[5 marks]

Family Name: . . . .

Given Names: . . . .

3. The following statements are concerned with the DFS algorithm applied to a digraph G. State whether each is TRUE or FALSE (+1 mark will be given for each correct answer, -0.5 for each incorrect one). **[5 marks]**

   (a) If (x, y) is an arc, x was visited before y and x finished after y then (x, y) is a cross arc.

   ANSWER:
   FALSE

   (b) If (x, y) is a cross arc then x was visited after y finished.

   ANSWER:
   TRUE

   (c) If (x, y) is an arc, x was visited after y and x finished before y then (x, y) is a back arc.

   ANSWER:
   TRUE

   (d) If (x, y) is a forward arc then x was visited before y and x finished after y.

   ANSWER:
   TRUE

CONTINUED

Family Name: . . . .

Given Names: . . . .

(e) If G is a DAG, then a topological ordering of G can be obtained by listing the nodes in increasing order of their first visiting time.

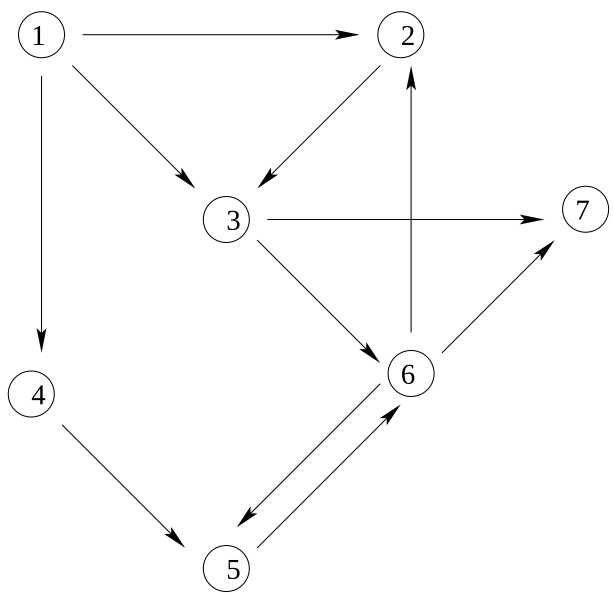
ANSWER:
FALSE

[5 marks]

Family Name: ....

Given Names: ....

4. Suppose that we run BFS on the digraph shown, using the convention that whenever there is a choice of node to visit, we choose the lowest numbered node.



(a) How many trees are in the BFS forest?

ANSWER:
1

(b) Write down the distance vector (ith entry is the distance from node 1 to node i, for $i = 1, \ldots, 7$).

ANSWER:
$[0, 1, 1, 1, 2, 2, 2]$

Family Name: . . . .

Given Names: . . . .

(c) List all cross arcs, back arcs and tree arcs.

ANSWER:
Cross: $(3, 7), (4, 5), (5, 6), (6, 2), (6, 7)$. *Back: none. Tree:* $(1, 2), (1, 3), (1, 4), (3, 6), (3, 7), (4, 5)$.

[5 marks]

5. Now suppose that we run again BFS on the digraph shown in the previous question, this time using the convention that whenever there is a choice of node to visit, we choose the HIGHEST numbered node.

(a) How many trees are in the BFS forest?

ANSWER:
4

(b) Write down nodes in the order in which they were visited (level order).

ANSWER:
$7, 6, 5, 2, 3, 4, 1$

(c) List all cross arcs, back arcs and tree arcs.

ANSWER:
Cross: $(1, 2), (1, 3), (1, 4), (4, 5), (3, 7), (6, 7)$. *Back:* $(3, 6), (5, 6)$.
Tree: $(6, 5), (6, 2), (2, 3)$.

[5 marks]