THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2006
Campus: City

COMPUTER SCIENCE

Algorithms and Data Structures

(Time allowed: ONE hour)

NOTE: Attempt *all* questions!

Put the answers in the boxes below the questions.

Marks for each question are shown just before each answer box.

Use of calculators is NOT permitted.

Section:	**A**	**B**	Total
Possible marks:	30	30	60
Awarded marks:			

Section A: Analysis of algorithms

1. (a) Prove directly from the definition that if a and b are positive real constants, then $T(n) = an \ln n + b$ is $O(n^2)$,

[5 marks]

(b) True or false: $n^{\log n}$ is in $O(\sqrt{n})$. Give full explanation.

[5 marks]

2. For each question, circle the (unique) correct answer. Note: correct answers score +1 point, incorrect ones -0.25 so the expected score for completely random guessing is 0.

   (a) When the entries $7, 4, 6, 1, 2, 3, 8, 5$ are successively inserted into an initially empty binary search tree, what is the height of the resulting tree?

   A. 4 B. 3 C. 7 D. 2 E. 1

   (b) Which of the following inputs will make the standard implementation of quicksort (always choose the leftmost element as the pivot) work the hardest?

   A. 1,3,5,6,4,2 B. 6,5,4,3,2,1 C. 1,6,2,5,3,4 D. 2,4,6,1,3,5 E. 1,3,5,2,4,6

   (c) Which of the following inputs will make the standard implementation of quicksort (always choose the leftmost element as the pivot) work the least?

   A. 1,2,3,4,5,6,7 B. 7,6,5,4,3,2,1 C. 2,4,6,7,5,3,1 D. 2,1,3,4,6,5,7
   E. 1,3,5,7,6,4,2

   (d) When hashing with separate chaining is used, and the elements $1, 2, 3, 4, 5, 11, 12, 13, 21, 22, 31$ are inserted into an initially empty hash table of size 5 using the hash function $h(x) = 2x + 1 \mod 5$, what is the maximum chain size required?

   A. 2 B. 0 C. 1 D. 4 E. 3

   (e) If algorithm A has time complexity $\Theta(n \log n)$ and runs for 1000 ms on a problem of size 10000, about how long would you expect it to take to solve a problem of size 1000000?

   A. 100000 ms B. 2000000 ms C. 150000 ms D. 1000000 ms
   E. 240000000 ms

   (f) You know that algorithm B runs in exponential time $\Theta(10^n)$. If your computer can process a problem of size 1000 in one year using an implementation of this algorithm, what size problem would you expect to be able to solve in one year with a computer 100 times faster?

   A. 100000 B. 1100 C. 2000 D. 1002 E. 1000.2

(g) I am thinking of a well-known algorithm (***-sort) that is stable, in-place, and runs in linear time in the best case. What is ***?

A. merge B. insertion C. heap D. selection E. quick

(h) If, in a given computing environment, data moves are very expensive and comparisons are cheap, which sorting method is likely to be best for a medium-large file?

A. selection B. heap C. insertion D. merge E. quick

(i) If your life depended on correctly sorting data in a reasonable time and you had very little memory available in your application, your best choice would be

A. quick B. merge C. insertion D. Shell E. heap

(j) The number of inversions in the permutation 135246 is

A. 2 B. 3 C. 6 D. 1 E. 0

(k) The average running time of quickselect when looking for the median of a list of size n is of order

A. 1 B. n^2 C. $\log n$ D. $n \log n$ E. n

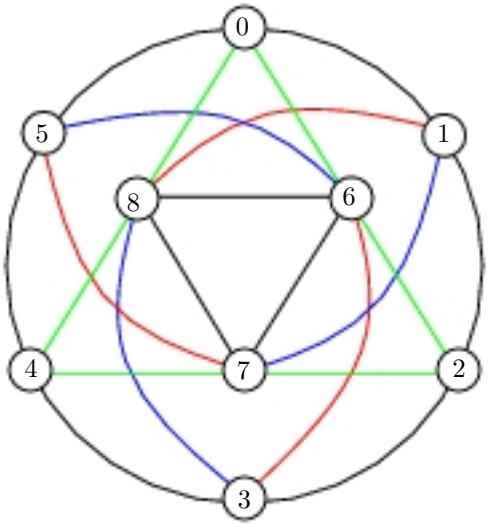
3. Consider the recurrence relation $t(n) = t(\lfloor \frac{n}{2} \rfloor) + t(\lceil \frac{n}{2} \rceil) + \lg n$, with $t(1) = 0$.

   Assume that $n = 2^k$ and derive an exact closed formula for $t(n)$. What is the asymptotic growth rate of $t(n)$ for such values of n?

[10 marks]

Section B: Graph algorithms

4. Consider the following graph.



(a) What is the order of the graph? [1 mark]

(b) What is the size of the graph? [1 mark]

(c) What is the degree of vertex 0 of the graph? [1 mark]

(d) What is the diameter of the graph? [2 marks]

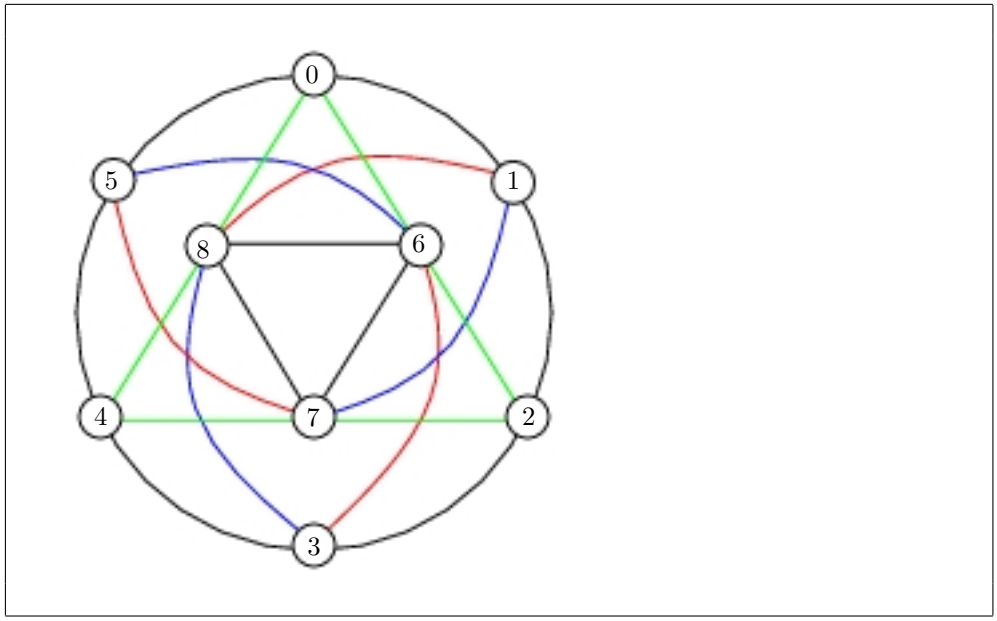
(e) What is the girth of the graph? [2 marks]

(f) What is the radius of the graph? [2 marks]

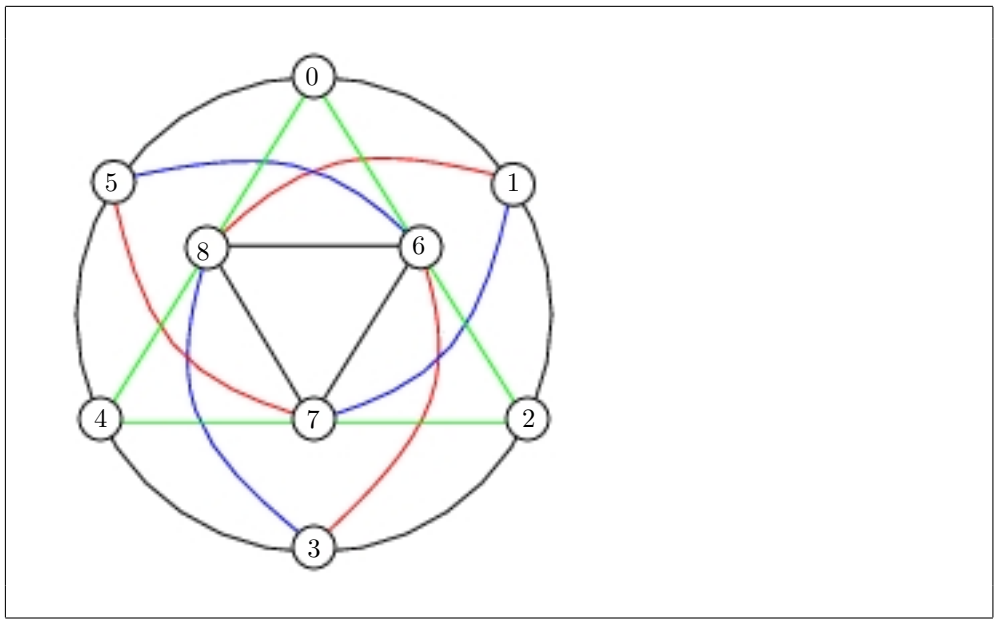
(g) Write the adjacency lists representation of this graph. [2 marks]

(h) Write the adjacency matrix representation of this graph. [2 marks]

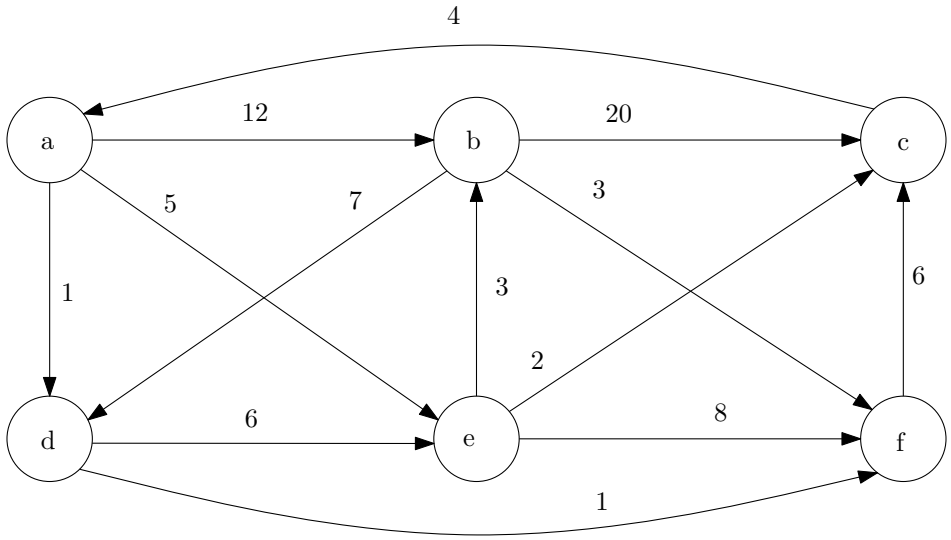
(i) Illustrate BFS on the graph starting from vertex 0 and draw the BFS tree, with the usual rule that when faced with a choice we take the lowest labeled vertex. [4 marks]



(j) Illustrate DFS on the graph and draw the DFS tree (with seen/done times) starting from vertex 0, with the usual rule that when faced with a choice we take the lowest labeled vertex. [4 marks]



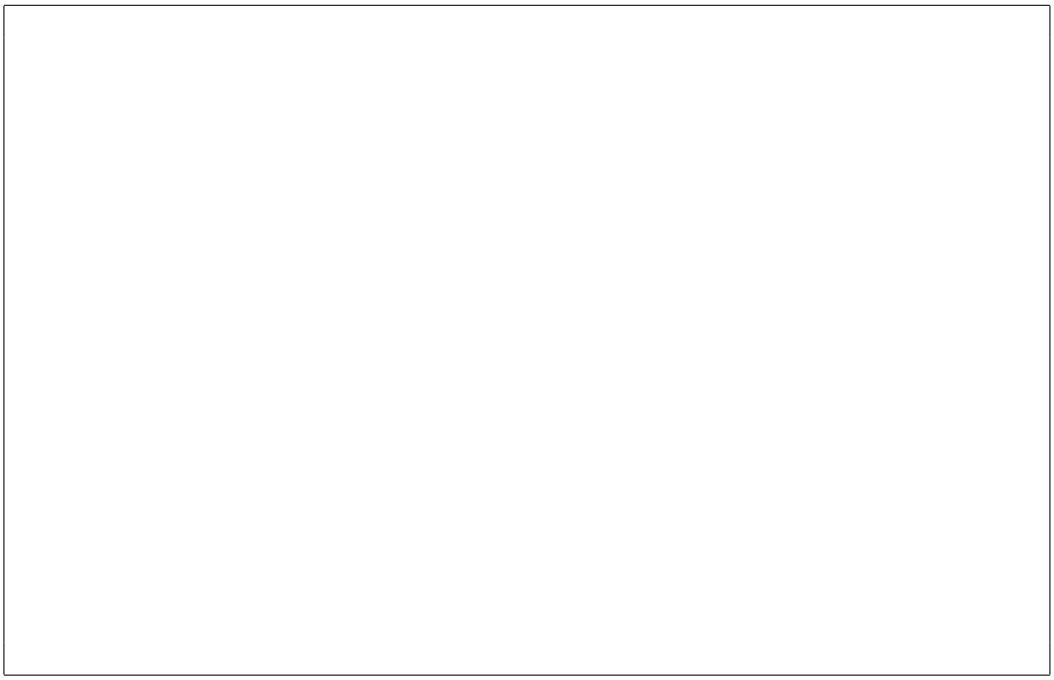
5. Consider the following edge-weighted digraph.



Illustrate Dijkstra's single-source shortest paths algorithm starting from node a by filling out the remaining entries of the following table. [6 marks]

Black nodes S	Distance to nodes (from node a) a	b	c	d	e	f
a	0	12	∞	1	5	∞
a, d						
a, b, c, d, e, f						

6. Give an example of a bipartite graph with 7 vertices, 8 edges and diameter 4. Justify your answer. [3 marks]



Overflow page 1

Overflow page 2