THE UNIVERSITY OF ADELAIDE
AUSTRALIA

Primary Examination, Semester 2, 2013

Algorithm and Data Structure Analysis
COMPSCI 2201, 7201

Official Reading Time:	10 mins
Writing Time:	120 mins
Total Duration:	130 mins

Questions	Time	Marks
Answer all 7 questions	120 mins	120 marks
		120 Total

Instructions

- Begin each answer on a new page
- Examination material must not be removed from the examination room
- Only Simple Calculators With No Alpha-Numeric Memory Allowed

Materials

- 1 Blue book
- 1 Dictionary for translation purposes only

DO NOT COMMENCE WRITING UNTIL INSTRUCTED TO DO SO

Right or Wrong?

Question 1

(a) Indicate whether each of the following statements is true or false. There is one mark for each correct answer and zero marks for each incorrect answer.

	Statement
1	$n^{8/7} \in \Omega(n)$
2	$2.1 \cdot n^{2.1} + 1500n^2 \in \Theta(n^3)$
3	The Floyd-Warshall all shortest path algorithm takes time $\Theta(nm + n^2 \log n)$. (where m is the number of edges and n is the number of nodes in the graph).
4	Kruskal's algorithm for minimum spaning tree takes time $\Theta(m \log m)$
5	It is known that there are problems in P that are not in NP
6	The problem to decide whether a given graph contains an Eulerian cycle is NP-hard
7	All optimisation problems are in NP
8	AVL trees have $O(\log n)$ time for *delete* operations.
9	The Dijkstra single-source shortest path algorithm cannot work on graphs containing negative-weight cycles.
10	Unbounded arrays always have $O(1)$ time access regardless of how we decide to reallocate space for them – as long as we always allocate sufficient space of course.

[10 marks]

[Total for Question 1: 10 marks]

Invariants, Proofs and Sequences

Question 2

(a) Describe the invariant maintained by binary search for a value in a sorted array of values?

[4 marks]

(b) Prove by induction that the sum:

$$2^0 + 2^1 + 2^2 + \ldots + 2^n = 2^{n+1} - 1$$

[5 marks]

(c) Show the binary heap structure that is equivalent to the following list (the root is first element).

[5, 9, 8, 12, 15, 11, 19, 14, 20, 18, 17, 13]

[4 marks]

(d) Show the resulting tree after the value 6 is added to the heap in the part (c). Note that the binary heap properties must be restored after insertion. Show your working; you may show the data structure in tree or array form.

[3 marks]

[Total for Question 2: 16 marks]

Hashing and Skiplists

Question 3

(a) A hash function, $h(x)$, hashes a name, x (the hash key), onto a hash value (the index into an array) as listed in the following table. The hash table has a size of 8 (indexed from 0 to 7).

The keys are inserted into a hash table in the order listed in the following table. Use the following collision resolution approaches as described in the lectures/textbooks.

y	h(y)
Lime	6
Mandarin	5
Mango	5
Grapefruit	1

Show the hash table contents after insertion with:

i. chaining

[2 marks]

ii. linear probing

[2 marks]

iii. quadratic probing

[2 marks]

iv. Discuss how to handle the deletion of `Mango` from the above linear probed table. Note, you have a choice in your answer because there are multiple correct ways to handle deletions in linear probed tables.

[3 marks]

(b) Derive the expression for the expected height H of an element inserted into a skiplist.

[3 marks]

(c) What is the probability that, upon insertion of an element in a skiplist, the height of that element is greater than a given height x? Write your solution in terms of x.

[2 marks]

[Total for Question 3: 14 marks]

Trees

Question 4

(a) Draw a maximally balanced binary search tree that can be produced from the elements: $1, 2, 3, 4, 5, 6, 7, 8, 9$. *Hint:* a maximally balanced binary search tree minimises the average depth of its elements.

[3 marks]

(b) Derive a *recurrence* for the number of *degenerate* binary search trees that can be generated from a given sequence of n distinct elements. Recall that degenerate binary search tree contains no branches and thus is structurally similar to a linked list.

You must make an argument to justify each component of your recurrence. Remember that all recurrences have base cases so do not forget to include a base case.

[6 marks]

(c) Derive a *recurrence* for the number of different binary search tree *shapes* that can be derived from a sequence of n distinct values. You must carefully explain your derivation in order to get full marks for this question. Again, you must remember to include at least one base case in your recurrence.

[6 marks]

(d) Draw a sequence of diagrams showing the insertion of the values:

`[1,9,8,7,2,3,6,4,5]`

into an empty AVL tree, in the order shown above.

You must:

- Show the resulting tree immediately after each insertion step (that is *before* any balancing has taken place).
- Indicate the node(s) at which each rotation is performed.
- Where there is a double rotation, show the tree after each single rotation.
- Show the resulting tree after balancing operation(s).

[10 marks]

[Total for Question 4: 25 marks]

Graph Representations and Traversals

Question 5

(a) Write pseudo-code for an algorithm that performs depth-first search of a directed graph: $G = (V, E)$ and prints out all of the **edges** as they are *checked*. Note: an edge is *checked* when it is inspected by the algorithm (not necessarily traversed).

[9 marks]

(b) State the storage requirements of a graph with n nodes and m edges using:

1. an adjacency list, and
2. an adjacency matrix

briefly justify each answer.

[4 marks]

[Total for Question 5: 13 marks]

Shortest Path Algorithms

Question 6

(a) Breadth-first search (BFS) can be used to perform single source shortest paths on any graph where all edges have the same weights. Write an algorithm using BFS to assign distances to all nodes in a graph and give the time complexity of this algorithm.

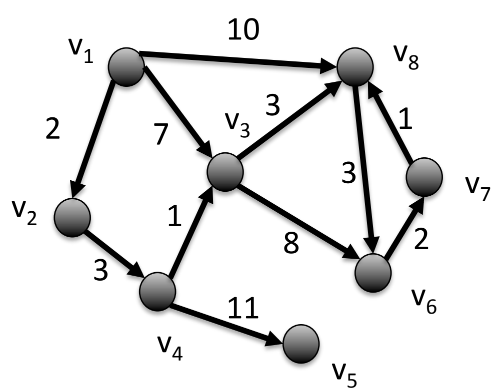
[7 marks]

(b) Dijkstra's algorithm for single-source shortest path on directed graphs, as given in lectures, assumed a heap-based priority queue was used to store tentative unscanned node distances. However, the original version of Dijkstra's algorithm used an *array* to store these distances.

Give the time complexity of this original version of Dijkstra's algorithm taking account of the cost of array access. Give a brief explanation of your answer.

[5 marks]

(c) Consider the following graph:



Solve the single-source-shortest path problem for the start node v_1 using Dijkstra's algorithm. List for each iteration which nodes becomes scanned and which edges are relaxed.

[12 marks]

(d) Briefly explain, with the use of an example, why Dijkstra's algorithm will not work on graphs with negative weight edges.

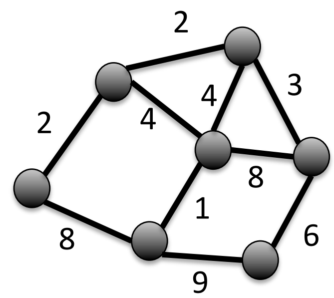
[4 marks]

[Total for Question 6: 28 marks]

Minimum Spanning Trees and P vs NP

Question 7

(a) Draw two different minimum spanning trees for the graph below.



In your answer show the final trees including the weights on the links of the trees.

[6 marks]

(b) What does it mean to say that a problem is not in NP? Give an example of a problem that is not in NP.

[4 marks]

(c) Briefly describe how a problem can be shown to be NP-Hard.

[3 marks]

(d) What was the first problem shown to be NP complete?

[1 mark]

[Total for Question 7: 14 mark]

End of exam