UNIVERSITY OF OTAGO EXAMINATIONS 2011

COMPUTER SCIENCE

Paper COSC242

ALGORITHMS & DATA STRUCTURES

Semester 2

(TIME ALLOWED: THREE HOURS)

This examination consists of 5 pages including this cover page.

Candidates should answer **all** questions.

Questions are worth various marks each and submarks are shown thus: (5)

The total number of marks available for this examination is 100.

No supplementary material is provided for this examination.

Candidates may **not** bring calculators, reference books, notes, or other written material into this examination room.

TURN OVER

1. **Complexity classes**

Listed below are some algorithms that you have encountered in this course. For each one, state whether the **worst** case has $\Theta(1), \Theta(\log n), \Theta(n), \Theta(n \log n), \Theta(n^2)$, or $\Theta(2^n)$ time complexity for n items.

(a) Insertion Sort (1)

(b) Merge Sort (1)

(c) Counting Sort (1)

(d) Searching an unsorted array (1)

(e) Binary Search (1)

(f) Searching a Perfect Hash Table (1)

(g) Searching a Red-Black Tree (1)

(h) Finding all the subsets of a set. (1)

2. **Recurrences**

Use the iteration method to solve the recurrence

$f(1) = 5$
$f(n) = f(n-1) + 2.$

You do **not** need to prove that your solution is correct. (4)

3. **Big-O and Induction**

Using the definition of big-O and induction, prove that $n^3 = O(2^n)$. (7)

4. **Sorting**

(a) Carefully describe the differences between Merge Sort and Quicksort. (5)

(b) Mention one way to improve Merge Sort. (2)

(c) Describe one way to improve the partitioning in Quicksort. (2)

(d) If an algorithm that sorts items by comparing key values cannot do better than $\Theta(n \log n)$, how is it possible that one can sometimes sort keys in $O(n)$ time? (3)

(e) What is the difference between a stable sort and an unstable sort? Mention one situation in which you really need a stable sort. (3)

5. **Hash Tables**

(a) Given a table of size 7, a hash function h(k), and input keys 27, 47, 81, 74, 11, 50, and 64 (in that order), draw the hash table that results from:

(i) Chaining, with $h(k) = k\%7$. (4)

(ii) Open addressing with double hashing. Use $h(k) = k\%7$ as the primary hash function, and $g(k) = 1 + (k\%6)$ as the secondary hash function. (5)

(iii) Chaining, with universal hash function $h_{(10,10)}(k) = ((10k + 10)\%101)\%7$. (6)

(b) Suppose you were using a perfect hashing scheme to create a hash table from the keys above. Would $h_{(10,10)}$ be acceptable as the primary hash function? Show your reasoning. (3)

(c) Suppose you are using double hashing with the secondary hash function g as described above. Explain why a hash table of size 10, 100, or 1000 would be a poor choice, regardless of the number of keys. (2)

6. **Binary search trees**

(a) Draw the final binary search tree T that results from successively inserting the keys 5, 4, 3, 2, 1 into an initially empty tree. (1)

(b) Write down the keys of T in the order in which they would be visited during a postorder traversal. (1)

(c) Draw the results of deleting 4 from T. (1)

(d) Give one reason why you might choose to store data in a binary search tree instead of a hash table. (1)

7. **Red-black trees**

(a) Show all the red-black trees that result after successively inserting the keys 5, 4, 3, 2, 1 into an initially empty red-black tree. State which cases apply. (6)

(b) Show all the red-black trees that result from the deletion of 4. State which cases apply. (4)

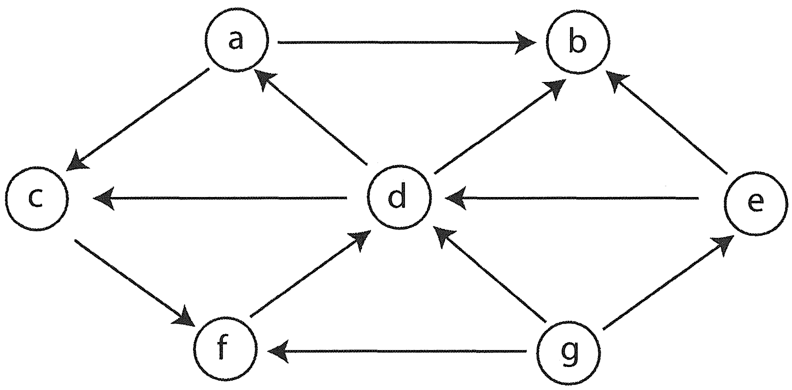
8. **B-trees**

(a) By a 2-3-4 tree we mean a B-tree of minimum degree t = 2. Show the results of successively inserting the keys 7, 3, 4, 9, 8, 11, 14, 16, 18 into an initially empty 2-3-4 tree. You should at least draw the trees just before some node must split and just after the node has split. (5)

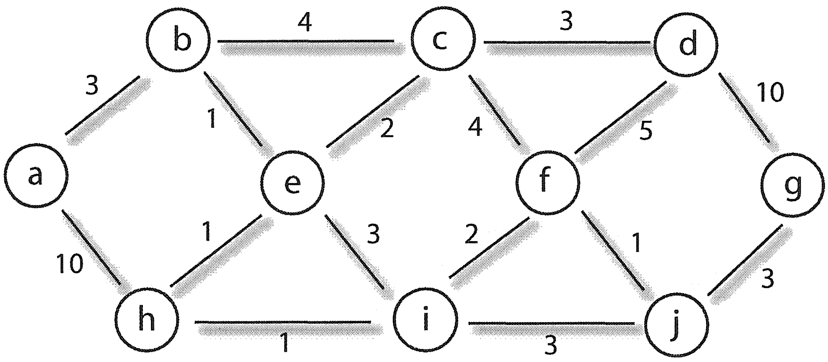
(b) Show what happens when you delete key 9. (3)

9. **Graphs**

(a) Copy the following directed graph into your answer book. Do a depth-first search starting at vertex a, showing time stamps, and label the edges with T, F, B, or C according to whether each is a tree edge, forward edge, back edge, or cross edge. (8)

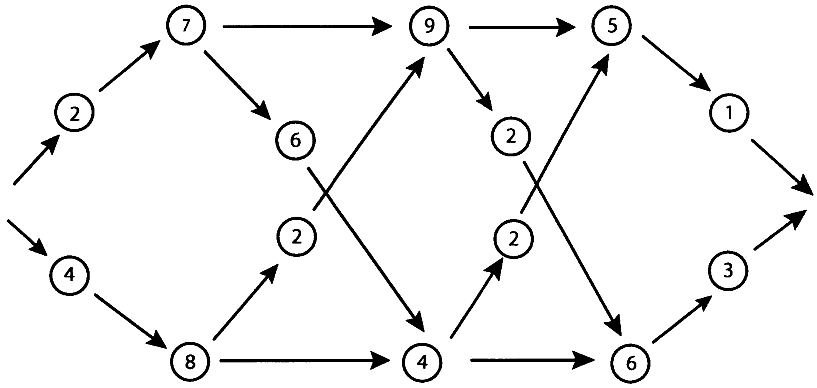


(b) Copy the following undirected graph into your answer book. Show how Dijkstra's algorithm works, with a as source. Show clearly how the priority values change, and the order in which vertices are extracted from the priority queue. (6)



10. **Dynamic programming**

Consider the assembly line scheduling problem below. Give a dynamic programming solution. Show any bottom-up tables used in your solution and any calculations you perform. Explain what the entries in your tables mean. (5)



11. **P and NP**

In a few well-chosen sentences, tell Aunt Maud what the classes P and NP are, and what it means to say that a problem is NP-complete. Give her one example of an NP-complete problem. (5)