UNIVERSITY OF OTAGO EXAMINATIONS 2010

COMPUTER SCIENCE

Paper COSC242

ALGORITHMS & DATA STRUCTURES

(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.

1. **Complexity classes**

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

- Quicksort
- Merge sort
- Searching an unordered list
- Binary Search
- Searching a Perfect Hash Table
- Searching a Binary Search Tree
- Searching a Red-Black Tree (7)

2. **Recurrences**

Show by the **iteration method** that the recurrence

$f(1) = 1$
$f(n) = n * f(n-1)$

defines the function "n factorial" (i.e. n!). You do **not** need to prove the solution to be correct. (5)

3. **Big-O and Induction**

Using induction, prove that $n^2 = O(n!)$ (7)

4. **Sorting**

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

(b) Suppose you are to implement a sorting algorithm for a search engine. It will sort all of the words in all of the documents in a very large collection: this will take quite a long time and memory will be tight. What sorting algorithm would you implement, and what would you do to make it perform as well as possible? (4)

(c) Draw the bucketing structure produced by Bucket Sort when sorting the following data: 0.34, 0.54, 0.88, 0.81, 0.11, 0.50, 0.71. (3)

(d) What is the difference between a stable sort and an unstable sort? Describe a situation in which a stable sort is necessary. (3)

5. **Hash Tables**

(a) Given a table of size 7, a hash function h(k), and input keys 34, 54, 88, 81, 11, 50, and 71 (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. (4)

(iii) Chaining, with universal hash function $h_{(10,10)}(k) = ((10k + 10) \% 101) \% 7$. (4)

(b) Suppose you were using a perfect hashing scheme to create a hash table from the keys above. What would be the *cost* of using $h_{(10,10)}$ as the primary hash function? Why is that cost considered unacceptable? (2)

(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 1, 5, 2, 4, 3 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 1, then 5, then 2 from T. (2)

(d) Give one reason why you might choose to use 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 1, 5, 2, 4, 3 into an initially empty red-black tree. (6)

(b) Show all the red-black trees that result from the successive deletion of 1, then 5, then 2. (4)

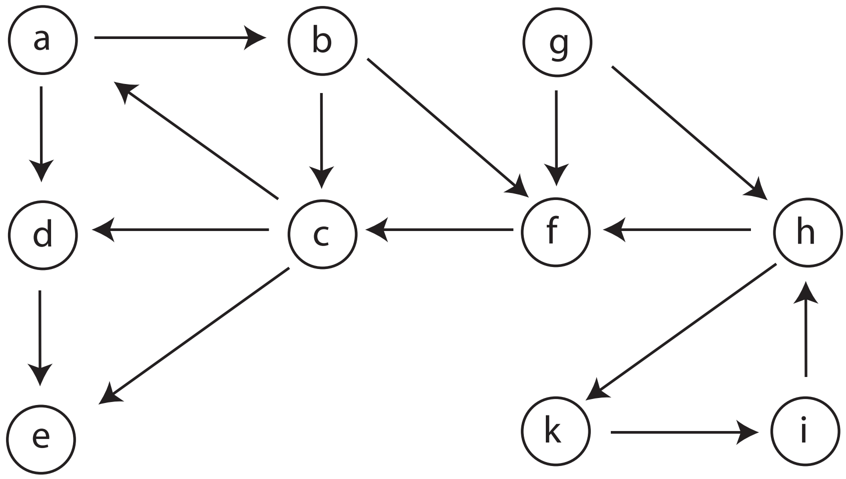
8. **B-trees**

(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 10, 9, 8, 7, 6, 5, 4, 3, 2, 1 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)

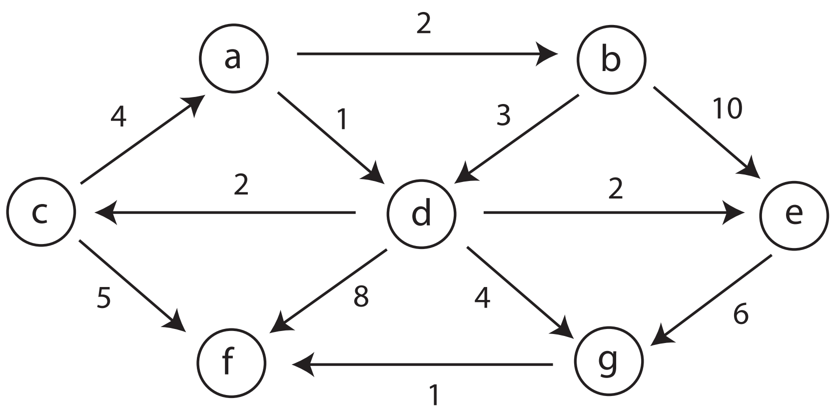
(b) Show what happens when you delete first key 4 and then key 1. (5)

9. **Graphs**

(a) Copy the following directed acyclic graph into your answer book. Do a depth-first search starting at vertex b, 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)

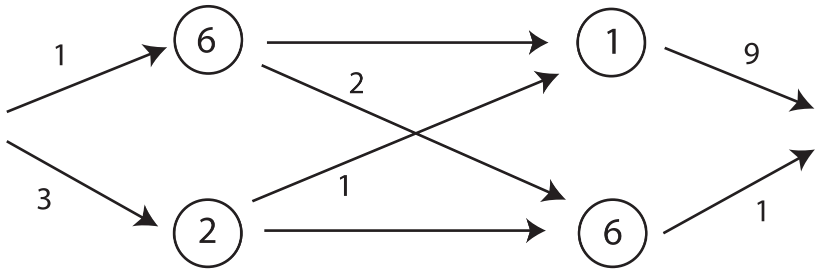


(b) Copy the following directed graph into your answer book. Show how Dijkstra's algorithm works, with *a* as source. Show clearly how the priority values change, and the order in which vertices are extracted from the priority queue. (7)



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)