UNIVERSITY OF EAST ANGLIA

School of Computing Sciences

Main Series UG Examination 2014-15

DATA STRUCTURES AND ALGORITHMS

CMP-5014Y/CMPC2M11

Time allowed: 3 hours

Section A (Attempt any 4 questions: 60 marks)

Section B (Attempt any 2 questions: 60 marks)

Notes are not permitted in this examination.

Do not turn over until you are told to do so by the Invigilator.

SECTION A

1. (a) Outline a general strategy for describing the complexity of an algorithm. [4 marks]

   (b) A matrix is said to be lower triangular if all the elements above the diagonal are zero. The algorithm below tests whether an input matrix is lower triangular or not. Use the strategy you described in the first part of this question to ascertain the worst case complexity of `isLowerTriangular`.

```
Input:  Matrix M[1..n][1..n]
Output:  boolean true if M is lower triangular
 begin isLowerTriangular(Matrix[][] M, size n)
    for i:=1 to n−1  loop
        for j:=i+1 to n  loop
            if M[i][j] != 0
                return false
    return true
end isLowerTriangular
```

   Show your working. [11 marks]

2. Given the following set of keys:

$$S = \{514, 5141, 5148, 516, 823, 8235, 82356, 971, 9718\}$$

   (a) draw the trie corresponding to S; [7 marks]

   (b) represent the trie from (a) using a two-dimensional array. [8 marks]

3. (a) What is a list data structure and how does it differ from a set data structure? [2 marks]

   (b) What are the differences between an array based and a linked list implementation of a list data structure? Include a statement of the complexity of the standard operations to access and structurally modify the data structures. [5 marks]

   (c) Describe the stack and queue abstract data types, including a description of the standard operations that can be performed. [4 marks]

   (d) Explain how you could use an array to implement a queue, including how to perform the standard queue operations. Include diagrams where necessary. [4 marks]

4. Consider the following set of disjoint sets:

$$S = \{\{2,3,4\},\{1,7\},\{5,6,10,11\},\{8,9\}\}.$$

   (a) Represent S by an appropriate forest of trees. [3 marks]

   (b) Explain how your forest may in turn be represented by a one-dimensional array and give the array. [4 marks]

   (c) Suppose that it is required to merge the set containing 4 with the set containing 10. Draw the resulting forest of trees and give the corresponding one-dimensional array. [3 marks]

   (d) Give an algorithm that uses the array representation of a set of disjoint sets to find the set containing item i, where i is specified. Illustrate the application of this algorithm using the above example. [5 marks]

5. (a) In the context of hashing, briefly describe what is meant by each of the following terms:

    (i) hash function;

    (ii) collision;

    (iii) probe increment;

    (iv) quadratic probing;

    (v) double hashing

[5 marks]

(b) Let $h(k) = k\%13$, where $\%$ is the remainder operator (e.g. $17\%13 = 4$). Insert the following set of keys into a hash table of size 13, using quadratic probing to resolve collisions. For each key inserted, give the sequence of locations probed during that insertion.

$$19, 27, 20, 33, 13, 24, 28, 14, 16, 7.$$

[5 marks]

(c) Repeat the exercise in part (b) but this time using double hashing with the following secondary hash function:

$$p(k) = 5 - (k\%5).$$

[5 marks]

SECTION B

6. (a) Give an informal description of the selection sort algorithm. [3 marks]

   (b) Describe how selection sort could be improved by the utilization of a heap. Give informal and formal pseudo-code descriptions of the heap sort algorithm. You may use the operations `heapify` and `sift-down` without a description of how they work. [8 marks]

   (c) When might it make sense to sort an array by grouping objects rather than by comparison and swapping? [3 marks]

   (d) Give informal and formal pseudo-code descriptions of the counting sort algorithm. [6 marks]

   (e) What are the time and space complexities of counting sort? [2 marks]

   (f) Give informal and formal pseudo-code descriptions of the bucket sort algorithm. [8 marks]

7. (a) Describe the quicksort algorithm both informally and formally with pseudo-code. Include a description of how to partition. [12 marks]

   (b) Quicksort the following array into ascending order, using the median of the first three elements as a pivot

$$A = [13, 4, 10, 23, 15, 6, 8, 17, 18, 7, 12]$$

   If the array is of size two pivot on the first element. [6 marks]

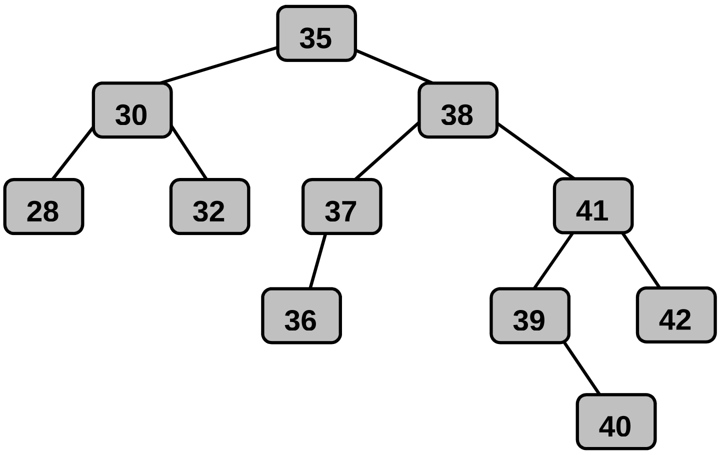
   (c) Perform a worst-case analysis of the quicksort algorithm, assuming the pivot is always the last term in the segment to be sorted. State the average case complexity. [8 marks]

   (d) Describe the refinements that are usually introduced into the quicksort algorithm in real world implementations. [4 marks]

8. (a) (i) Define what is meant by the *balance* of a node in a binary tree. [1 mark]

When is a binary search tree *height-balanced*? [2 marks]

(ii) Determine the balance of each node in the following binary search tree.



[2 marks]

(iii) Draw diagrams to illustrate the effect in a binary tree of

A. a left rotation,

B. a right rotation. [4 marks]

(iv) State what must be done to re-balance the binary search tree given in part (c). Draw diagrams to illustrate your answer. [6 marks]

(b) (i) Define what is meant by a B-tree of order m. [4 marks]

(ii) Compare and contrast the use of a B-tree data structure with that of a height-balanced binary search tree data structure. [3 marks]

(iii) Draw diagrams to illustrate the insertion of the following keys, in the order given, into a 2-3 tree.

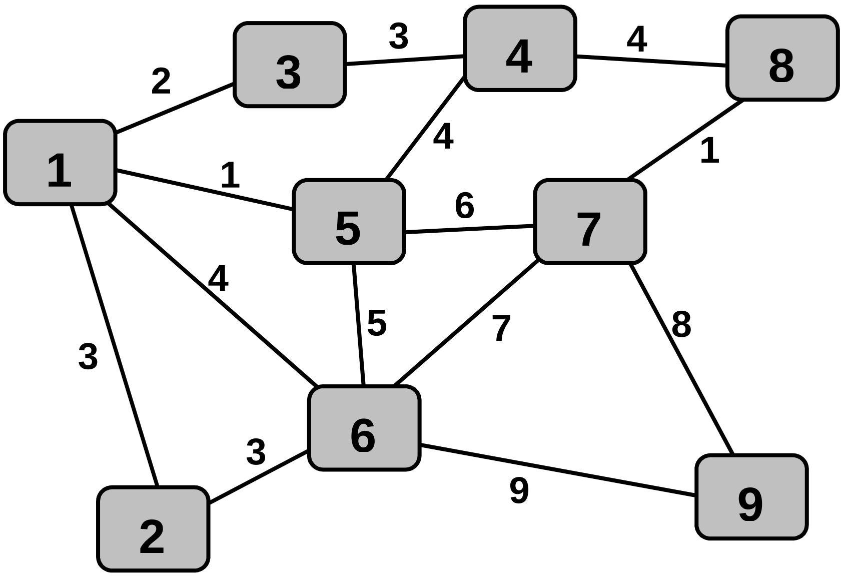
$$38, 59, 46, 28, 17, 54, 75, 34, 21, 71, 47, 68, 27.$$

[8 marks]

9. (a) Let $G = (V, E)$ be a connected graph with a positive weight associated with each edge in E.

   (i) Describe Kruskal's algorithm for finding a minimum spanning tree of G. [8 marks]

   (ii) Illustrate the application of Kruskal's algorithm on the following weighted graph. [7 marks]



(b) (i) Define what is meant by a (max) heap. Describe the main ideas underlying the `deleteMax()` method for such an object. State the worst case run-time complexity for the `deleteMax()` method on a heap containing n elements, briefly justifying your answer. [7 marks]

(ii) Describe a linear-time algorithm for constructing a heap. Draw diagrams to illustrate your answer by creating a max heap from the following sequence of integers:

$$34, 41, 63, 40, 55, 36, 76, 66, 43, 51, 67.$$

[6 marks]

(iii) Use your heap to illustrate an application of the `deleteMax()` operation. [2 marks]

END OF PAPER