A31759 *No calculator permitted in this examination*

UNIVERSITY OF BIRMINGHAM

School of Computer Science

First Year - BSc Artificial Intelligence and Computer Science
First Year – BSc Computer Science
First Year – MSci Computer Science
First Year – MEng Computer Science/Software Engineering
First Year – BSc Mathematics & Computer Science
First Year – MSci Mathematics & Computer Science
First Year – BSc Computer Science with Study Abroad
First Year – MSci Computer Science with Study Abroad
First Year – BSc Computer Science with Business Management
First Year – MSci Mathematics & Computer Science with Industrial Year
First Year – BSc Computer Science with Industrial Year
First Year – MEng Computer Science/Software Engineering with Industrial Year
First Year – BSc Artificial Intelligence and Computer Science with Industrial Year
First Year – BSc Computer Science with Business Management with Industrial Year
First Year – MSci Computer Science with Industrial Year
First Year BA/BSc Liberal Arts & Sciences

06 28345

Data Structures and Algorithms

Main May/June Examinations 2017

Time allowed: 1 hour 30 minutes

[Answer ALL Questions]

No calculator

Answer all questions

1. (a) Give inductive definitions of a *Binary Tree* and of a *Quad Tree*. [4%]

   (b) What additional conditions must a *Binary Tree* satisfy for it to be a valid *Binary Search Tree (*BST*)*? [3%]

   (c) Draw the BST that results from inserting the items [6 9 7 3 1 8 5 2] in that order starting from an initially empty tree. Then draw the tree that results from removing the root node from that tree using the standard *BST* delete algorithm. [5%]

   (d) Write in pseudocode an efficient BST-based procedure `printArray(A)` that prints in ascending order the values held in an arbitrarily-ordered array `A`. You may call any of the standard primitive binary tree constructors, selectors and conditions, a procedure `size(A)` that returns the size of array `A`, a procedure `insert(v,bst)` that returns the BST formed by inserting the value `v` into an existing BST `bst`, a procedure `print(v)` that prints the value `v`, and any other pseudocode procedures you choose to write in the same way. [8%]

   (e) Explain the overall average-case and worst-case time complexities of your procedure in terms of the number of elements *n* in the array. [5%]

2. (a) Outline what is meant by the terms *Priority Queue* and *Binary Heap Tree*, and how they may be used for sorting a collection of items. [6%]

   (b) Explain the general idea of *divide and conquer* approaches to sorting, and how the *Quicksort* algorithm applies that idea to sort a collection of items stored in an array. [7%]

   (c) What are the average and worst case time complexities of the *Quicksort* algorithm? Give a simple example of how the worst case can easily arise, and how it can be avoided. [6%]

   (d) What does it mean to say that a sorting algorithm is *stable*? Discuss the stability of the *Quicksort* algorithm and how that relates to the efficiency of the algorithm. [6%]

No calculator

3. (a) Explain what a *hash table* is, and what it means to say *a hash table has had a hash collision*. [4%]

   (b) What are the computational advantages and disadvantages of using hash tables for data storage compared to other approaches? [4%]

   (c) Outline the three principal approaches for dealing with hash collisions, together with the main advantages and disadvantages of each. [9%]

   (d) Suppose a university wants to create a hash table to store approximately 70,000 student records identified by seven-digit ID numbers. Describe some sensible hash table approaches for doing that, with reasons why they would be appropriate, and suggestions how to avoid common problems. [8%]

4. (a) Explain what *traversal of a graph* means, and the main difference between *breadth first traversal* and *depth first traversal*. Which data structures would be best used to implement those two traversal types? [6%]

   (b) How can graph traversal be used to determine whether a graph is *connected*? [2%]

   (c) Explain what is meant by a *minimal spanning tree* of a weighted graph, and suggest a practical example of when one may be useful. [6%]

   (d) Describe an efficient greedy vertex-based algorithm for finding a minimal spanning tree of a weighted graph. State in what sense is your algorithm is greedy. [5%]

   (e) Assuming you keep your algorithm simple by *not* using a priority queue, what is its time complexity? Comment on how the graph's connectivity proportion affects the relative speed of your algorithm compared to Kruskal's edge-based algorithm for the same problem. [6%]

Do not complete the attendance slip, fill in the front of the answer book or turn over the question paper until you are told to do so

Important Reminders

- Coats/outwear should be placed in the designated area.
- Unauthorised materials (e.g. notes or tippex) must be placed in the designated area.
- Check that you do not have any unauthorised materials with you (e.g. in your pockets, pencil case).
- Mobile phones and smart watches **must** be switched off and placed in the designated area or under your desk. They must not be left on your person or in your pockets.
- You are not permitted to use a mobile phone as a clock. If you have difficulty seeing a clock, please alert an Invigilator.
- You are not permitted to have writing on your hand, arm or other body part.
- Check that you do not have writing on your hand, arm or other body part – if you do, you must inform an Invigilator immediately
- Alert an Invigilator immediately if you find any unauthorised item upon you during the examination.

Any students found with non-permitted items upon their person during the examination, or who fail to comply with Examination rules may be subject to Student Conduct procedures.