UNIVERSITY OF BIRMINGHAM

School of Computer Science

First Year - UG Affiliate Biosciences
First Year - BSc Artificial Intelligence and Computer Science
First Year - UG Affiliate Computer Science/Software Engineering
First Year - BSc Computer Science FT
First Year - MSci Computer Science
First Year - MEng Comp Science/Software Engineering
First Year BSc Mathematics + Computer Science
First Year - MSci Mathematics + Computer Science
First Year - BSc Computer Science w Study Abroad
First Year - BSc Computer Science w Business Management
First Year - BSc Computer Science w Foundation Year
First Year - BSc Mathematics + Computer Science w Industrial Year
First Year - MSci Mathematics + Computer Science w Industrial Year
First Year - BSc Computer Science w Industrial Year
First Year - MEng Comp Science/Software Engineering w Industrial Year
First Year - BSc Artificial Intelligence and Computer Science w Industrial Year
First Year - BSc Computer Science w Business Management w Industrial Year
First Year - MSci Computer Science w Industrial Year
Second Year - BA/BSc Liberal Arts + Sciences

06 28345

Data Structures Algorithms

Main May/June Examinations 2018

Time allowed: 01:30

[Answer ALL Questions]

No Calculator

[Answer all questions]

1. (a) Give an inductive definition of a *Binary Tree.* [2%]

   (b) What additional conditions must a *Binary Tree* satisfy for it to be? (i) *Complete*, and (ii) a *Binary Heap Tree (BHT)*? [4%]

   (c) Explain what the *Bubble Up* and *Bubble Down* operators do in the context of heap trees, and what they are used for. [4%]

   (d) The items [6 9 4 8 12 5] need to be inserted one at a time into a *BHT*, starting from an empty tree. Show the state of the tree after each item has been inserted. [3%]

   (e) What is the *BHT* remaining after removing the root node from that tree using the standard *BHT* delete algorithm? [2%]

   (f) Describe in words a general purpose algorithm for merging any two array-based *BHTs* that is more efficient than simply inserting items one-at-a-time from one tree into the other. You may use the standard *BHT Bubble Up* and *Bubble Down* procedures without having to define them. [5%]

   (g) If both BHTs have size O(n), state in Big-O notation how the average time complexity of your merging algorithm compares with the simpler one-at-a-time insertion approach. [5%]

2. (a) One approach for sorting an array `a` of items is to first insert them into a *Binary Search Tree (BST)* `t` and then output them back into the array. Write in pseudocode a recursive procedure for filling the array from the BST, and specify what the initial call of the procedure is. You may call any of the standard primitive operators for binary trees, [5%]

   (b) What does it mean to say a sorting algorithm is *stable*? Explain how the stability of a BST-based sorting algorithm depends on its implementation. [4%]

   (c) Describe how one can establish the best possible average-case time complexity of any comparison-based sorting algorithm. Does BST-based sorting achieve that? [6%]

   (d) Show how *Two-Phase Radix Sort* can be used to sort the following set of dates given in day/month format:
   [17/7, 12/7, 8/4, 8/7. 9/3, 12/4, 17/3, 12/6]. [5%]

   (e) Explain how *Radix Sort* could be used to sort a set of integers without duplicates. Is that approach more efficient than the best possible comparison-based sorting algorithms? [5%]

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

   (b) How is the *load factor* of a hash table defined and why is it important to know? [3%]

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

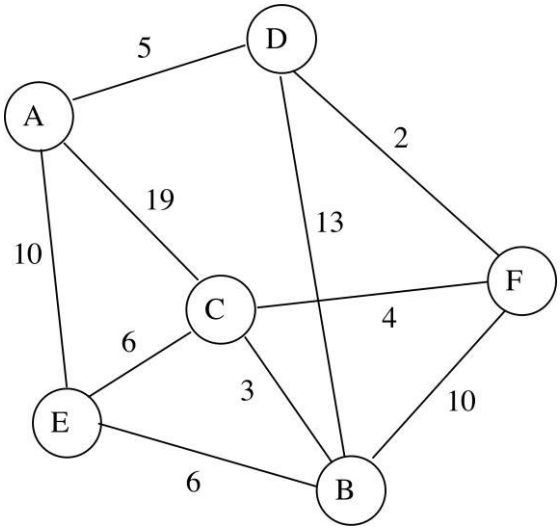
   (d) Explain what issue *Direct Chaining* and *Open Addressing* are designed to deal with, how they each work, and what their relative advantages and disadvantages are. [6%]

   (e) Suppose you have a hash table of size m, with a hash function that distributes the items as evenly as possible, and insert n items into the initially empty hash table using *Open Addressing*, where $n < m$. Derive the probability of there not being a collision when inserting the ith item. Then use that to derive a simple expression, containing only factorials and powers, for the probability of having had no collisions after inserting all n items. [8%]

4. (a) Describe the simplest *array-based*, *pointer-based* and *mixed* approaches for implementing weighted graphs. State the main advantages and disadvantages of each approach. [6%]

(b) Explain how an array-based *Dijkstra's algorithm* can be used to find the shortest path between two nodes in a given weighted graph. [6%]

(c) Use that array-based Dijkstra's algorithm to find the shortest path from A to B in the following weighted graph. Show the content of the relevant arrays at each stage and state the shortest path found. [8%]



(d) For sparse graphs, outline how the average time complexity of Dijkstra's algorithm depends on the implementational details. [5%]

A31759 Data Structures Algorithms

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.