Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Introduction to Algorithms & Data Structures

Date: Friday 29\textsuperscript{th} May 2009

Time: 14:00 – 16:00

Please answer THREE Questions from the FOUR questions provided

Use a SEPARATE answerbook for each SECTION

This is a CLOSED book examination

The use of electronic calculators is permitted provided they are not programmable and do not store text.
Section A

1. a) Explain clearly what is meant by the time complexity of an algorithm. (2 marks)

   b) What is meant by the rate of growth of a time complexity measure? Explain in detail the big “O” notation used to express rate of growth. (3 marks)

   c) For each of the following properties of time complexity, give an example of an algorithm with the property. In each case, give the relevant time complexities and explain clearly how the complexities are derived from the algorithm.

      i) Where the rate of growth of the best-case and worst-case time complexities are the same.

      ii) Where the rate of growth of the best-case and worst-case time complexities differ, with the average-case being the worst-case.

      iii) As (ii), but with the average-case being the best-case. (5 marks)

   d) Consider lists whose elements are integers without any element repeated. For two such lists, their intersection is a list of all the elements which occur in both lists. For example, for lists [3, 1, 4, 2, 6] and [7, 4, 3, 8, 2], their intersection is list [3, 2, 4] (any order of these elements is equally acceptable).

      i) Describe an algorithm to compute the intersection of any two such lists without sorting the lists. You need not write a program, but your algorithm should be clearly explained. In terms of the lengths of the two lists, what is the time complexity of your algorithm? Explain your answer. (5 marks)

      ii) Consider now first sorting each list into ascending order and then calculating the intersection. Is there a more efficient algorithm for intersection when the lists are in ascending order? If so, describe such an algorithm and give its time complexity. If not, explain why not. (5 marks)
2. a) Describe precisely the Insert Sort algorithm for sorting an array of integers into ascending order. Your description should include:

i) a program in a programming language or pseudocode, or a clear step-by-step explanation of the algorithm; (5 marks)

ii) a correctness argument; (2 marks)

iii) an analysis of its best-case, average-case and worst-case time complexity. Explain how these are derived. (3 marks)

b) Consider a modification of Insert Sort in which binary search is used to locate where to insert each item. Does this give an algorithm with an improved performance? Explain your answer in terms of time complexity measures. (4 marks)

c) Consider the following idea for a sorting algorithm: at the current position in a list, if there is a preceding item, compare this with the current item, if they are in order, move the current position right one place; if not, swap them and move the current position left one place.

This algorithm can be implemented with one loop. Explain clearly how this can be done, by presenting the algorithm as a program or in pseudocode.

What is the worst-case time complexity of the algorithm? Explain your answer. (6 marks)
Section B

3. a) Describe a pointer-free representation of a forest of trees. Explain how this representation can be used to represent a partition of a finite set, including operations to determine whether two set elements are in the same member of the partition and to merge members of the partition. (8 marks)

b) What is the heap property of a binary tree? (2 marks)

Describe how a heap can be constructed in linear time, illustrating your answer by making the following sequence into a heap

[32, 15, 16, 29, 11, 27, 14] (6 marks)

Describe how a heap can be used to represent a priority queue, including a brief description of insertion and removal, making clear how the heap property is maintained. (4 marks)

4. a) Define the term minimum-cost spanning set for a connected, undirected graph. Explain why this set must be a tree. (4 marks)

b) Describe Kruskal’s algorithm for finding the minimum spanning tree. Illustrate your answer by applying the algorithm to find a minimum spanning tree for the following graph. (10 marks)

Outline an implementation strategy for this algorithm, describing carefully any data structures involved. (6 marks)

END OF EXAMINATION