Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Introduction to Algorithms & Data Structures

Date: Wednesday 4th June 2008
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.
1. a) Explain clearly what is meant by the time complexity and the space complexity of an algorithm. (2 marks)

b) How is time complexity related to the running time of an algorithm? What criteria are used to choose a suitable time complexity measure? (3 marks)

c) What is meant by the best-case, average-case, and worst-case time complexity of an algorithm? Illustrate your answer with two examples of algorithms. (4 marks)

d) Explain what is meant by a time complexity lower-bound for a computational task and a computational gap. Give an example of a task for which there is no computational gap and an example of a task for which, with current state of knowledge, there is a computational gap. (3 marks)

e) What is meant by a polynomial-time algorithm and an exponential-time algorithm? In what circumstances do exponential-time algorithms arise? (3 marks)

f) Give an example of an exponential-time algorithm. You need not write a program but your description should make the steps of your algorithm clear. (5 marks)
2. a) Explain clearly what is meant by a divide-and-conquer algorithm. (2 marks)

b) Describe ONE divide-and-conquer algorithm for sorting lists of integers into ascending order. Your description should include

i) a precise presentation of the algorithm, either as a program that you explain, or a clear step-by-step account;

ii) a correctness argument for the algorithm; and

iii) an analysis of its worst-case, average-case and best-case time complexity of the algorithm. (8 marks)

c) Consider sorting a list of integers, where there are few different integers compared to the length of the list i.e. some integers are repeated many times. Thus, consider lists of length N containing M distinct integers, with M considerably less than N.

Consider a sorting algorithm for such lists based on the following idea: With one pass through the list we can find the minimum and maximum elements in the list. Now iterate through the list putting all minimum elements at the front (by interchanging elements) and, at the same time recording the next smallest item for the next pass through the list.

Describe clearly this sorting algorithm. You may present an outline program or a clear step-by-step account. (6 marks)

What is the worst-case and average-case time complexity of the algorithm? Under what conditions on M and N is this an improvement on mergesort? (4 marks)
Section B

3. a) What is dynamic programming? Describe circumstances under which the use of dynamic programming is appropriate. (4 marks)

b) Describe the Floyd-Warshall algorithm for finding the minimum cost paths between all pairs of points in a directed graph. Illustrate your answer by applying the algorithm to the graph with adjacency matrix

\[
\begin{pmatrix}
0 & 15 & 26 & \infty \\
40 & 0 & 25 & 15 \\
30 & \infty & 0 & 95 \\
15 & \infty & 85 & 0 \\
\end{pmatrix}
\]

(13 marks)

What effect does the introduction of edges with negative costs have on the applicability of this algorithm? (3 marks)

4. a) Describe the circumstances in which hash tables might be used, and outline the different approaches taken by open hashing and closed hashing. What criteria should be used to choose a suitable hash function? (8 marks)

b) Describe three approaches which may be adopted when using closed hashing to solve the collision resolution problem, illustrating your answer with a hash table of size 10, integer keys, the hash function \( h(x) = x \mod 10 \), and data to insert 421, 153, 623, 489, 44, 99, 199. (12 marks)

END OF EXAMINATION