ENGINEERING TRIPOS PART IIA

Wednesday 11 May 2011 9:00 – 10:30

Module 3I1

DATA STRUCTURES AND ALGORITHMS

*Answer **all** of Section A (which consists of short questions) and **two** questions from Section B.*

All questions carry the same number of marks.

*The **approximate** percentage of marks allocated to each part of a question is indicated in the right margin.*

There are no attachments.

STATIONERY REQUIREMENTS	SPECIAL REQUIREMENTS
Single-sided script paper	none

You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator

Version: 2

SECTION A

1 (a) Suppose algorithm A has cost O(n) and algorithm B has cost O(n^2). Could there be circumstances when it would be rational to use algorithm B? Explain. [10%]

(b) In an ideal world each time Quicksort selects a pivot this would lead to the data being partitioned into two equal sections. When that happens Quicksort can complete in time proportional to $n\log(n)$. Suppose instead that the pivot always splits the data at the 1% position, in other words the two sections after partitioning have sizes in the ratio 1:99. What are the cost consequences? [10%]

(c) You wish to find the item that is at rank $N/100$ from a set of N items. You have two possible methods to use. If you use the guaranteed linear-cost median finding algorithm you can assume the time taken will be around $10N$. As an alternative you can start using heap-sort on the N items and stop as soon as you have found the information you need. Which is liable to be better, and does you answer depend on N? [10%]

(d) Now you wish to identify the median, in other words the item whose rank is $N/2$. Is your answer still the same as in part (c) above? Explain. [10%]

(e) What will be the worst case, and hence what will be the worst case cost for releasing a block of memory back into the control of a Buddy system allocator? [10%]

(f) What advantages do Red-Black trees have over ordinary binary-search trees, and what disadvantages or extra costs do they have? [10%]

(g) You have set up a hash table (which stores everything within the array that represents the table) and added items to it so it is now just two-thirds full. You now wish to remove one of the items that you inserted earlier. Explain how you do this and estimate the cost involved. [10%]

(h) One way of implementing Shell's Sort uses a sequence of strides using a recurrence $s_{k-1} = 2s_k + 1$ and arranging that the final stride in the sequence is just 1. Would it be reasonable instead to use a sequence of strides that was just powers of 3, as in one that finished $\ldots 27, 9, 3, 1$ instead of $\ldots 40, 13, 4, 1$? If not what disadvantage might arise from the slightly simpler sequence? [10%]

(i) The word "engineers" has 3 instances of the letter "e", two of "n" and one of each other letter it uses. You will also need a code ω that will mark the end of file and which should be treated as being much less common than anything else. Using just the letters present in "engineers" plus ω with the relative frequencies indicated here create a Huffman coding tree that could be used to send streams of letters. [10%]

(j) Why is it useful to consider analysis in terms of amortised computing time when discussing garbage collection? Define any technical terms you need to use, and explain any limitations that balance the strengths that you identify. [10%]

2 (a) Explain how the data structure known as a Heap and as used in Heapsort treats a sequence on N items stored in an array as if they were arranged in a very well balanced binary tree. [15%]

(b) A programmer seeks to impose essentially perfect balance on all the binary search trees they use, and at the same time avoid the need for pointers. They decide to use the heap representation as in part (a). Explain in detail how to look up a key in a binary search tree stored in this manner (supposing that the tree already exists) and explain the costs involved. [15%]

(c) Suppose N items are to be formed into a binary seach tree in this form. Can you tell which item will end up at the top of the tree (i.e. at the first position in the array)? If so which item will it be, or if there is flexibility comment about any items that could *not* end up in the top position. [15%]

(d) Given your data as a sorted list, hence or otherwise develop a procedure that can form it into a properly arranged search tree filling just the N initial entries in the array. How long will the process take? [15%]

(e) Now the programmer has established a perfectly balanced search tree, but wants to add a new item. Are there circumstances in which no other data will need moving, and are there cases in which every other item must be moved? [15%]

(f) Invent and describe an algorithm that inserts a new item such that the insertion procedure performs approximately $\log(N)$ comparisons. You may disregard all cost of data movement and other administration. [25%]

3 Data will be transmitted and the raw channel can cope with a set of 10 symbols, 0-9. The messages to be sent, however, use a much larger alphabet, so the engineer who is setting everything up designs a variable length encoding scheme very loosely based on the idea of Huffman Encoding. The digits 0-4 are used to represent the 5 most commonly used characters. Two digit codes of the form 50-59, 60-69 and 70-79 provide a less compact way of denoting the next 30 characters. Then 800-899 and 900-998 provide 199 more. The final code 999 is used to mark the end of the data.

The engineer now wishes to use a further layer of file compression technology to convert this stream of symbols into a stream of bits.

Two schemes are under consideration – Lempel Zif and Arithmetic Coding. In each case the compression will work with one symbol at a time, i.e. it is not considered proper to decode the existing stream into the string of items it represents. You must handle it digit by digit, but if relevant you may maintain some status or history information.

(a) For Lempel Zif comment on how the behaviour of compression is liable to be affected by the variable length encoding. Describe how the method works, giving sufficient detail to reveal any special behaviour which may arise. [40%]

(b) For arithmetic coding, what other information would you need, if any, and how would you use it? Describe the coding process, commenting on how the statistical modelling involved is affected by the nature of the raw data. [40%]

(c) Which of these two compression methods would be preferable here, and why? [20%]

4 You are given an undirected graph that has integer-valued weights associated with each edge.

(a) Define a *minimum spanning tree* for such a graph and explain an algorithm that will find one. Your answer should include discussion of particular bottlenecks or ways in which special sorts of graphs could impact costs. You should also provide an upper bound on the cost of your method in terms of the number of vertices and edges involved. It will be sufficient to use simple techniques for any sub-tasks that need handling. [35%]

(b) In the same graph a user identifies two vertices, A and B, and your task is to find a shortest path from A to B through the graph, treating the weights on the edges as distances. You are expected to report both the length of this path and the chain of edges that make it up. Present and analyse an algorithm for solving the problem. [35%]

(c) You are now told that around two thirds of the edges in such a graph will have weight 1, while the remaining edges will have weights of 2 or 3. Discuss whether, and if so how, you would alter your algorithms of parts (a) and (b) to take advantage of this new information. [30%]

END OF PAPER