ENGINEERING TRIPOS PART IIA

Monday 1 May 2006 2.30 to 4

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.*

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: draft 1

SECTION A

Answer all parts of this question. The question in this section will be marked out of the same total as each question in section B, and each part carries the same weight.

1 (a) Explain precisely what it means to say $f(n) = O(g(n)$. [10%]

(b) In big-O notation, derive and justify a worst-case cost for Quicksort. [10%]

(c) Explain briefly what is meant by analysis in terms of *amortised* computing time. [10%]

(d) Given the recurrence formula $f(n) = 2f(n/2) + n$ with some suitable value for $f(1)$, find or quote and justify a closed-form solution for $f(n)$. [10%]

(e) A hash table stores all data within itself. The table has space to hold up to 1000 values, and 500 of these are present. On average how many probes of the table are performed when a new key different from the 500 that are present is looked up? [10%]

(f) What aspect of storage management that gives trouble for both first-fit and best-fit is addressed by Buddy allocation methods? [10%]

(g) Give a circumstance where a Buddy system of storage management might be less efficient than using Garbage Collection, and explain why this is so. [10%]

(h) Give a sequence of operations on a *splay tree* that will result in it holding n elements and being of height n. [10%]

(i) Explain how to construct a string that Huffman encoding will not manage to shrink. [10%]

(j) Explain what a minimum-cost spanning tree of a graph is. Does every graph have a minimum-cost spanning sub-tree? [10%]

SECTION B

*Attempt **two** questions from this section. Each question has the same weight in marks.*

2 (a) In the context of Heapsort, explain how to take initially unordered data and rearrange it so that the Heap Property applies. [34%]

(b) In the content of Quicksort you are given an array of length n containing arbitrary data in arbitrary order and a selected value p taken from that data (*the pivot*). Explain how to rearrange the data so that all values less than the pivot end up to its left, and all values greater than it to its right. [33%]

(c) In the context of binary insertion sort, which is expected to complete a sorting process using very close to the smallest possible number of comparisons (although it may use excessive data movement) explain how to find where an item must be inserted into that part of the data that has already been sorted. [33%]

In each case a full answer will give some explanation of context, will describe the procedure involved carefully (paying attention to any limiting or awkward cases) and will give and justify a cost estimate in big-O notation. Methods that are unnecessarily costly will gain less credit than those that maximise performance.

3 (a) What is a *red-black* tree? What are the key properties of red-black trees and what problem do they solve? [30%]

(b) The lectures explained the structure of red-black trees and talked through the procedure for adding a new item into such a tree. Discuss the problem of looking up a value that is stored in a red-black tree and removing it from the tree. Show cases where deletion is easy and note cases where it is harder, and get as far towards inventing, describing and justifying a complete deletion algorithm as you can. [70%]

4 (a) For arithmetic coding, explain how the recipient of a compressed string of bits can decode it to retrieve the original message. [50%]

(b) When encoding, special treatment is required when the input is such that the compressed output will be either of the form 01111111 or 10000000 with either a zero followed by a long run of ones or a one followed by a long run of zeros. Is there any special treatment that your decoder has to take in these cases? [10%]

(c) What is the largest number of bits the decoder must read before it can output a symbol? [20%]

(d) Arithmetic encoding is performed using limited precision integer arithmetic. For instance the examples in the notes used 3-bit arithmetic leading to a partition of the range [0 ... 1] into 8 regions. What effects on compression and on costs arise as that working precision changes? [20%]

END OF PAPER

Version: draft 1