COMPUTER SCIENCE TRIPOS Part IA

Wednesday 1 June 1994 9 to 12

Paper 2

Answer **five** *questions.*
At least **one** *question from each section is to be answered.*
Submit the answers in five **separate** *bundles each with its own cover sheet.*
Write on **one** *side of the paper only.*

SECTION A

1 A partition of n into r is a set $\{S_i\}$ of r strictly positive integers such that

$$\sum_{i=0}^{r-1} S_i = n$$

Derive a recurrence from which one could tabulate the Stirling numbers of the second kind $S(n, r)$, which are the numbers of distinct partitions of n into r. [8 marks]

Show how the tabulation may be started, assuming $S(0, 0) = 1$, and giving other boundary values. [4 marks]

$T(n, r)$ are similar numbers where some of the r integers are allowed to be zero (n.b. they can't all be!). Relate $T(n, r)$ to the Stirling numbers. [8 marks]

2 A bipartite graph is one where the vertices can be partitioned into two disjoint sets A and B such that every edge of the graph has one end at a vertex in A and the other at a vertex in B. Are there bipartite graphs for which the sets A and B are not uniquely defined? [2 marks]

Given an arbitrary graph how would you find out whether it was bipartite or not? [6 marks]

Define a *matching* in a bipartite graph and show how to find a maximum matching in any given bipartite graph. Prove any results you use about augmenting paths. [12 marks]

3 State carefully the conditions for a relation R on a set S to be:

(a) a partial order;

(b) a total order;

(c) a well-order, as applied to both (a) and (b). [6 marks]

Let S consist of ordered pairs of positive integers (i, j) such that $j > i$. Define relations on S that are

(a) totally and well-ordered;

(b) partially (not totally) and well-ordered;

(c) totally but not well-ordered. [14 marks]

In each case, explain why.

SECTION B

4 State and prove Bayes's Theorem for obtaining the probability of A given B from the probability of B given A and the probabilities of A and B. [4 marks]

A box of computer components contains items made to the same specification from two different manufacturers, Proctor and Co. and Constable and Co., in the ratio 3:2. It is known that 1 in 1000 of Proctor's components are faulty, but only 1 in 1500 of Constable's. A component is drawn at random from the box and found to be faulty. What is the probability it was manufactured by Proctor and Co.? [16 marks]

5 In $2n$ independent trials the probability of success is p_1 in each of the first n trials and p_2 in the remaining n. Prove that the mean and variance of the total number of successes are $n(p_1 + p_2)$ and $n(p_1 + p_2) - n(p_1^2 + p_2^2)$ respectively. [10 marks]

Hence show that unless $p_1 = p_2$ the variance of the number of successes is less than it would be in a binomial distribution with the same number of trials and the same mean number of successes. [10 marks]

6 A customer asks you, a programmer, to write a program to generate random deviates from a chi-squared distribution, and you recall that this is the same as the gamma distribution with probability density function

$$f(x) = x^{a-1}\mathrm{e}^{-x}/\Gamma(a).$$

For small integral values of the parameter a (say up to 10) how would you do this? [20 marks]

SECTION C

7 A small electronic mail system has been installed in your organisation, allowing the dozen or so workers in your group (all users of a single central server computer) to send each other messages about their work. There is now talk of extending this mail system to provide contact with several hundred other sites, each of which has from a dozen to a few hundred workers. For political reasons it will be necessary to construct a new mail system for this enlarged group, rather than buy in services from some commercial organisation that provides existing hardware or software.

Discuss some of the problems that will have to be considered in the design of the new mail system. Concentrate on ways that electronic mail will interact with the body of users rather than on any fine technical details of a design. In particular consider:

(a) Mail redirection when a user moves to a new location; [5 marks]

(b) Circulation lists and multiple copies of mail; [5 marks]

(c) Recovery of the cost of setting up and running the mail service; [5 marks]

(d) The possible alternative of using existing telephone lines and installing a large number of FAX machines. [5 marks]

8 Suppose that *take* and *drop* are ML functions such that $take(n, s)$ returns the first n elements of the list s, while $drop(n, s)$ returns all but the first n elements of s. Let $length(s)$ be the function to compute the length of the list s. Consider the following ML function

```
fun front s = take(length s div 2, s);
fun back s  = drop(length s div 2, s);

fun bsum [ ]   = 0.0
  | bsum [x]   = x
  | bsum s     = bsum front s + bsum back s;

fun sum [ ]    = 0.0
  | sum (x::s) = x + sum s;
```

Give a formal proof that $sum(front\ s) + sum(back\ s) = sum(s)$ for all lists s, explaining what properties of arithmetic you are assuming. [9 marks]

Describe a proof of $bsum(s) = sum(s)$ for all s using the lemma that you have just established. Do not give a detailed proof but instead outline the main argument. State any additional lemmas required and indicate how they might be proved. [6 marks]

Does proving $bsum(s) = sum(s)$ for all s in this way ensure that *bsum* and *sum* are completely interchangable in ML programs? Discuss. [5 marks]

9 Recall that a *binary search tree* is a binary tree whose labels are ordered from left to right, while a *priority queue* is a collection whose elements are inserted in any order but are removed in increasing order. Priority queues can be represented by binary search trees using the normal insert operation; they require additional functions *least* and *del_least*. Given a binary search tree, *least* returns the node with the smallest label, while *del_least* returns the binary search tree with that node removed.

Using a suitable `datatype` for binary trees, code *del_least* as a recursive (and side-effect free) ML function. Illustrate its operation using an example. [4 marks]

Consider now the cost of emptying an n-label binary search tree by applying *del_least* to it n times. The cost will be measured by the number of calls to *del_least*, including recursive calls.

Which binary search trees constitute the best and worst cases for this operation? State the cost of each as a function of n. [4 marks]

If the tree is perfectly balanced and of depth d then $n = 2^d - 1$. Show that the cost of emptying the tree in this case is $d \times 2^{d-1}$. [8 marks]

A functional program cannot update the binary tree 'in place' but must do some copying. To what extent does the number of calls to *del_least* reflect the true cost of emptying the tree? Discuss. [4 marks]

SECTION D

10 "Avoidable monopolies are objectionable because they lead to higher prices and/or lower supply than would be the case in a free market. Patents confer avoidable monopolies. Therefore, patents are objectionable." How would you defend the patent system against this argument? [20 marks]

11 You are in charge of the design of both hardware and software for a new (but fairly conventional) workstation which will have its peripherals (for example a disc drive and a printer) directly connected to it. Your workstation will be required to support an operating system that allows its user(s) to run several independent programs at once.

Explain and justify the method that you will provide for an application program to gain access to kernel services and physical input/output devices. [12 marks]

For both a printer and a disc drive, indicate additional functions (beyond those provided by the physical device itself) that an operating system typically provides in software. [8 marks]

12 If a program is written cautiously and in a suitable high level language then it can be "portable", and one set of source files can be used with little or no change on a wide variety of computers and with many different operating systems. Explain the steps that are taken to turn the source version of such a program into a runnable version of the application that it represents. Indicate all the places where programs or pieces of code not derived from the portable sources are involved. [4 marks]

Identify which parts of the software preparation path (if any) will need to be altered in each of the following cases, commenting on just where programs and code (not being directly part of the portable source) can be used unaltered and where different versions are called for:

(a) The program is to be run on different hardware configurations of the various models of the same computer; [4 marks]

(b) The program is to be run on computers which share the same hardware design and have the same processor, but under different operating systems (for example some PCs will run MSDOS, Unix and several other operating systems); [4 marks]

(c) The program is to be run on two machines that share a common operating system (such as Unix) but which have different processor designs. [4 marks]

Suppose the portable programming language involved is implemented using an interpreter rather than a compiler, and the language is already implemented on all the computer systems involved. How does this affect the amount of work and the number of changes needed when building an executable version of a program for use in many different environments? [4 marks]