COMPUTER SCIENCE TRIPOS Part IA

Monday 5 June 2000 1.30 to 4.30

Paper 1

Answer **two** *questions from Section A, and* **one** *question from* **each** *of Sections B, C, D and E.*
Submit the answers in six **separate** *bundles, each with its own cover sheet. On each cover sheet, write the numbers of* **all** *attempted questions, and circle the number of the question attached.*
Write on **one** *side of the paper only.*

SECTION A

1 Foundations of Computer Science

Code the curried function `exf`, which takes as arguments the function f and the list l. The result must consist of those elements x of l such that $f(x)$ is also a member of l. The elements of the result must be distinct from each other but may appear in any order. For example, if $f(x) = x + 1$ and $l = [9, 3, 2, 2, 8]$ then the result should be $[2, 8]$ or $[8, 2]$. [9 marks]

State, with justification, the type of `exf`. [1 mark]

2 Discrete Mathematics

State the conditions for a relation to be a partial order. [3 marks]

A *partition* of a natural number n is a collection of natural numbers (possibly including duplicates and in any order) whose sum is n. Let P_n be the set of partitions of n; for example, $P_4 = \{(4), (3,1), (2,2), (2,1,1), (1,1,1,1)\}$. Order the partitions in P_n as follows:

$$
\begin{aligned}
(a_1, a_2, \ldots a_r) \leqslant (b_1, b_2, \ldots b_s) \text{ if the } (a_i) \text{ and } (b_j) \text{ can be rearranged so that} \\
b_1 &= a_1 + a_2 + \cdots + a_{k_1} \\
b_2 &= a_{k_1+1} + a_{k_1+2} + \cdots + a_{k_2} \\
&\vdots \\
b_{s-1} &= a_{k_{s-2}+1} + a_{k_{s-2}+2} + \cdots + a_{k_{s-1}} \\
b_s &= a_{k_{s-1}+1} + a_{k_{s-1}+2} + \cdots + a_r
\end{aligned}
$$

Note that $(2,1,1) \leqslant (3,1)$, and $(2,1,1) \leqslant (2,2)$ but $(3,1)$ and $(2,2)$ cannot be compared.

Show that $\leqslant$ is a partial order on P_n. [4 marks]

P_5 has seven elements; draw the Hasse diagram for $(P_5, \leqslant)$. [3 marks]

3 Operating Systems

Operating systems need to be able to prevent applications from crashing or locking up the system, or from interfering with other applications. Which *three* kinds of hardware support do we require to accomplish this? Justify your answer. [6 marks]

How do applications request that the operating system performs tasks on their behalf? [2 marks]

What could we do if we did not have the requisite hardware support? [2 marks]

4 Programming in Java

The following attempt at Java code might have been written by a beginner. Identify (but do not correct) as many of the mistakes as you can. Explain how each oddity you identify is either something that would prevent the program from compiling, something that would cause it to stop abruptly reporting failure at runtime, a reason why the program might not do anything sensible or just a stylistic oddity.

```
//* A comment to start with: Exam:2000 */
public Class mycode.java
{
void static public fun main(String [junk])
  begin
    Leaf tr = null;
    for (i=1; i>10; ++i) tr = new Node(i, tr)
    tr.print();
  end;
}

class Leaf
{
    integer value;
    Leaf(int value)
    {   this = value; }
    public void print()
    {   System.out.println(value); }
}

class Node extends Leaf
{
    Leaf left, right;
    Node(leaf l, Leaf r)
    {   left = l, right = r;
    }
    void print()
    {   left.print();
        System.out.println("val=" @ value);
        right.print();
    }
}
```

[10 marks]

SECTION B

5 Foundations of Computer Science

State the time complexity of the *lookup* and *update* operations for each of the following:

(*a*) association lists

(*b*) binary search trees

(*c*) functional arrays (implemented as binary trees)

Use O-notation and include both the average-case and worst-case complexity. [6 marks]

You are provided with the ML code for binary search trees, including the *lookup* and *update* operations. Use these operations to code a sorting function that works by repeatedly inserting elements of a list into a binary search tree, then converting the final binary search tree back into a list. [4 marks]

Consider the following methods of sorting a list:

(*a*) Locate the smallest element of the input. The output is this element followed by the result of recursively sorting the remaining elements.

(*b*) Take the first 16 elements of the input and sort them using special hardware. Sort the remaining elements recursively. The output is the result of merging the two sorted lists.

(*c*) Take the first 20% of the input elements. Sort them and the remaining 80% recursively. The output is the result of merging the two sorted lists.

For each of these methods, state with justification the worst-case complexity in terms of the number of comparisons. [10 marks]

6 Foundations of Computer Science

Describe how ML lists are represented in storage. Your answer should include diagrams illustrating how the representation of $[a, b]@[c, d]$ is derived from those of the lists $[a, b]$ and $[c, d]$, indicating any sharing of memory. How efficient is the evaluation of $[a, b]@l$ if the list l is very long? [4 marks]

What are *cyclic lists* and how can they be created in ML? [2 marks]

Describe ML's reference types and their applications. In particular, compare mutable data structures with ordinary ML datatypes. [6 marks]

Code an ML function that takes a mutable list and returns true if the list is cyclic, otherwise returning false. Explain why your function is correct.
[Hint: in ML, the equality test $p = q$ is permitted on references and is true if p and q refer to the same location in memory.] [8 marks]

SECTION C

7 Discrete Mathematics

What does it mean for a partial order to be *well founded*? [3 marks]

Given two well founded, partially ordered sets $(A, \leqslant_A)$ and $(B, \leqslant_B)$, define the *lexicographic order* on $A \times B$ and show that it is well founded. [5 marks]

Two elements x and y of a partially ordered set are said to be *separated* if for all $k \geqslant 1$ there is a sequence of elements $z_1, z_2, \ldots z_k$ with $x < z_1 < z_2 < \cdots < z_k < y$.

Give an example of a well founded, partially ordered set that contains infinitely many pairs of separated elements. [5 marks]

Prove that no well founded, partially ordered set has every pair of elements separated. [7 marks]

8 Discrete Mathematics

The following fragment of ML implements Stein's algorithm for evaluating the Greatest Common Divisor, (a, b), of two natural numbers, a and b:

```
fun stein a b c =
  if a = b then a * c
  else
    if (a mod 2) = 0 then
      if (b mod 2) = 0 then stein (a div 2) (b div 2) (c * 2)
      else stein (a div 2) b c
    else
      if (b mod 2) = 0 then stein a (b div 2) c
      else
        if a > b then stein (a - b) b c
        else stein (b - a) a c;

fun gcd a b = stein a b 1;
```

Prove that, at each iteration within the Stein algorithm, the product $(a, b) \times c$ remains invariant. [8 marks]

Observing that the procedure starts with $c = 1$ and concludes by returning $a \times c$ when $a = b$, deduce that the algorithm correctly calculates the Greatest Common Divisor. [2 marks]

Show also that after two iterations the product $a \times b$ is reduced by at least a factor of 2. [6 marks]

Deduce that Stein's algorithm is at least as efficient as Euclid's algorithm. [4 marks]

SECTION D

9 Programming in Java

Write a Java class that provides support for arithmetic on the integers performed relative to some prime modulus p. An instance of the class should be constructible specifying the modulus, and then it should provide methods to create numbers and add, subtract, multiply and print them.

As a sample of the desired behaviour for your class, here is some test code for it:

```
Modular d = new Modular(7);  // work mod 7
ModInt a = d.reduceMod(10);  // create "10 mod 7"
ModInt b = d.reduceMod(20);  // create "20 mod 7"
ModInt c = a.add(b);         // work out a+b mod 7
c.print();
```

Note that I am suggesting a class called `Modular` that keeps track of the modulus p, and a second class `ModInt` to stand for numbers: these are created for the user via a method in `Modular`.

Your code should complain in some manner if, for example, an attempt is made to add a number that is defined modulo 7 to one that is defined modulo 11.

[14 marks]

Provide an additional method to perform division modulo a prime number.

[6 marks]

Note. Arithmetic modulo a prime number was explained in the lectures on Discrete Mathematics. In particular, the reciprocal of $a \pmod p$ can be found by solving the equation

$$ab = 1 \pmod p$$

10 Programming in Java

Explain *each* of the following: in relevant cases include up to four lines of Java code (but not more) to illustrate your point.

(*a*) How to print a table of integers in neat columns, e.g.

```
Heading1  Heading2
1234      989898976
76        -735743
8823674   66300
```
[4 marks]

(*b*) How to test if a floating point number is a NaN, an infinity or any other special cases that come to mind. [4 marks]

(*c*) How to arrange that, in an applet, a screen image is refreshed properly whenever the user moves, obscures or uncovers windows. [4 marks]

(*d*) Circumstances in which you might choose to declare one of your classes `final`, and cases when you might declare a method in a class `final` or `protected`. [4 marks]

(*e*) The keyword `finally` and its use. [4 marks]

SECTION E

11 Operating Systems

Describe with the aid of a diagram the life-cycle of a process. You should describe each of the states that it can be in, and the reasons why it moves between these states. [4 marks]

What information does the operating system keep in the process control block? [4 marks]

What information do the shortest job first (SJF) and shortest remaining time first (SRTF) algorithms require about each job or process? How can this information be obtained? [2 marks]

Give *one* advantage and *one* disadvantage of non-preemptive scheduling. [2 marks]

What steps does the operating system take when an interrupt occurs? Consider both the programmed I/O and DMA cases, and the interaction with the CPU scheduler. [4 marks]

What problems could occur if a system experienced a very high interrupt load? What if the device(s) in question were DMA-capable? [4 marks]

12 Operating Systems

In the context of memory management, under which circumstances do *external* and *internal* fragmentation occur? How can each be handled? [4 marks]

What is the purpose of a page table? What sort of information might it contain? How does it interact with a TLB? [3 marks]

Describe with the aid of a diagram a two-level page table. Explain the motivation behind the structure and how it operates. [5 marks]

What pieces of information make up the *meta-data* of a file? [4 marks]

Describe the basic access control scheme used in the Unix filing system. How does Unix support more advanced access control policies? [4 marks]

END OF PAPER