COMPUTER SCIENCE TRIPOS Part IA

Monday 4 June 2007 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.*

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

STATIONERY REQUIREMENTS
Script paper
Blue cover sheets
Tags

SPECIAL REQUIREMENTS
None

SECTION A

1 Foundations of Computer Science

(*a*) Code the ML function `merge`, which combines two ordered lists to form an ordered list containing the elements of both. [2 marks]

(*b*) Code a top-down merge sort function in ML. You may assume that basic functions on lists are given, provided you describe them briefly. [3 marks]

(*c*) State the time complexities of merge and merge sort, justifying your answer carefully. [5 marks]

2 Operating Systems

(*a*) In relation to scheduling of processes, describe the concept of a working set and briefly outline how it can be used within an operating system. [3 marks]

(*b*) Briefly explain why context switching between processes is inherently more costly than switching between threads of a process. [3 marks]

(*c*) Give *two* reasons why operating system designers often choose to make code in the kernel non-preemptive. [2 marks]

(*d*) Why would it be bad for a Unix file owned by `root` with the `setuid` bit set also to have read, write and execute access permissions granted to all users? [2 marks]

3 Programming in Java

(*a*) What is meant by a *generic* in the context of Java? Explain the main purpose of generics and the most important syntax associated with them. [3 marks]

(*b*) Explain how any program written using generics could be re-written to avoid them. Give a fragment of code illustrating the conversions needed to achieve this. Discuss why many people will view the version that does use generics as displaying better style and robustness. [4 marks]

(*c*) One of your fellow students puts forward the proposition "`String` is a sub-class of `Object`, therefore `Vector<Object>` is a sub-class of `Vector<String>`". Discuss. [3 marks]

4 Algorithms

For *each* of the following, indicate whether the statement is true or false, and explain why this is the case (you do not need to derive the correct asymptotic complexity yourself, but no marks will be awarded for an answer with no explanation).

(*a*) Heapsort executes in worst-case $O(n^2)$ time. [2 marks]

(*b*) Insertion sort executes in worst-case $\Theta(n^2)$ time. [2 marks]

(*c*) Shellsort executes in worst-case $\Omega(n^{1.5})$ time. [2 marks]

(*d*) All sorting algorithms are $\Omega(n \log n)$. [2 marks]

(*e*) A linked list is faster to access than a splay tree. [2 marks]

SECTION B

5 Foundations of Computer Science

(*a*) Consider the following piece of ML code:

```
datatype 'a tree = Lf | Br of 'a * 'a tree * 'a tree;
exception Blair;

fun tony p Lf = true
  | tony p (Br(x,t1,t2)) = if not (p x) then raise Blair
                           else tony p t1 handle Blair => tony p t2;

fun gordon p t = tony p t handle Blair => false;
```

(*i*) Code a function that returns the same results as `gordon` but makes no use of exceptions. [4 marks]

(*ii*) What property of binary trees does `gordon` express? [3 marks]

(*b*) Write brief notes on the ML type `exn`. [3 marks]

(*c*) Consider the following piece of ML code:

```
datatype 'a result = Ian of 'a | Cherie of exn;

fun what f x = Ian (f x) handle e => Cherie e;
```

We ask ML to evaluate the expression

```
map (what (tony (fn x => x <> 0))) [ta,tb]
```

and the response is as follows:

```
val it = [Ian true, Cherie Blair] : bool result list
```

What is the type of `what (tony (fn x => x <> 0))`, and what can we infer about the binary trees `ta` and `tb`? Justify both answers carefully.

[5+5 marks]

6 Foundations of Computer Science

(*a*) Write brief notes on reference types in ML and on control structures for imperative programming. [6 marks]

Consider the following ML datatype:

```
datatype 'a meal = Snack of 'a
                 | Lunch of 'a meal * 'a meal
                 | Feast of 'a meal * 'a meal * 'a meal;
```

(*b*) Write a function that is equivalent to snacker below but makes no use of references. Briefly explain why the two functions are equivalent.

```
fun snacker m =
  let val l = ref []
      fun munch (Snack x) = (l := x :: !l)
        | munch (Lunch (m1,m2)) = (munch m1; munch m2)
        | munch (Feast (m1,m2,m3)) =
            (munch m1; munch m2; munch m3)
  in munch m; !l end;
```

[5 marks]

(*c*) Write a function gluttony such that gluttony m1 m2 makes a copy of m1, replacing every Snack node with m2. [3 marks]

(*d*) Write a function glut such that glut k m1 m2 makes a copy of m1, replacing the *k*th Snack node with m2. Nodes are counted from left to right, with the leftmost node being number one. [6 marks]

SECTION C

7 Operating Systems

(*a*) Describe carefully how a 32-bit virtual address could be translated to a physical address during the execution of a memory reference instruction on a typical modern CPU that supports paged virtual memory. You should assume that the page size is 4096 bytes and that the system uses two-level paging with page tables at both levels holding 1024 entries. [5 marks]

(*b*) List the protection bits that you would expect to find in a page-table entry and briefly explain how they are used. [5 marks]

(*c*) Outline the main differences between paging and segmentation, and show how a segmentation scheme can be implemented with reasonable efficiency in a system that supports paging. [4 marks]

(*d*) Outline how you would implement, on a machine with 64-bit virtual addresses, the MULTICS-like view of files in which open files are mapped onto positions in virtual memory. [6 marks]

8 Operating Systems

(*a*) Most modern processors have a status register which include bits that specify whether it is running in supervisor or user state, and whether interrupts are enabled. Explain why this information is useful and what effect they have on instruction execution. [4 marks]

(*b*) Suggest *three* situations that cause the settings of these status bits to change. [3 marks]

(*c*) Discuss whether it is ever useful to run

(*i*) in supervisor state with interrupts enabled;

(*ii*) in user state with interrupts disabled. [4 marks]

(*d*) Carefully describe the structure of a Unix *inode*. Assuming that the block size is 4096 bytes in a Unix file system and that indirection blocks can hold 512 entries, calculate the size in bytes of the largest file that does not need to use triple indirection. You may give your answer as a formula. [5 marks]

(*e*) State *four* reasons why the Windows NTFS file system is superior to the FAT32 file system. [4 marks]

SECTION D

9 Programming in Java

(*a*) Explain how to set up a 2-dimensional array in Java. [2 marks]

(*b*) A simple spreadsheet is a grid of *cells*. Each cell can contain one of four possible things:

1. Nothing – the cell might be empty;

2. A fixed string, used as a label;

3. A fixed numerical value, represented as a *double*;

4. A formula, as discussed below, which will evaluate to a number.

The sorts of formulae to be supported to start with are very limited, but it is expected that later developments will add more options. For now a formula can indicate that the value in a cell is the sum of two other values whose coordinates are specified relative to the cell being considered. If one of the cells so addressed is empty, contains a string or is off the edge of the grid then it will be treated as if it contains zero.

Somewhere in the spreadsheet program there will need to be methods that make it possible to set the type of content of a cell, and to process the formulae until all values are up to date. They may of course need a number of additional fields and methods not explicitly noted in this specification.

(*i*) Design a set of Java classes that you can use to represent this set-up. Explain what fields and methods each will have, and what needs to be public and what can be kept private. At this stage you do not need to implement any elaborate methods, but you should explain what your methods must achieve. [9 marks]

(*ii*) Sketch an implementation of the methods involved in causing the spreadsheet to bring all its values up to date after the user alters the value in one cell. [9 marks]

10 Programming in Java

(*a*) A restricted variant on ML has types that are such that a type either is denoted by a type variable, α, β, etc; or is a function type of the form $\tau_1 \rightarrow \tau_2$ where τ_1 and τ_2 are simpler types. Design a Java class or set of classes that can be used to represent ML type expressions. Ensure that you provide a static method `createNewTypeVar` that makes a new ML type variable that is different from all the ones you have had before. Rather than giving your type variables Greek letters for their names you may call them `t1`, `t2`, `t3`, ... [7 marks]

(*b*) Suppose that your types are represented by a class called `MLType`. Explain all the changes you need to make to your code so that any `MLType` object has a method with signature

```
public void mustNotDependOn(TypeVar a) throws ItDoesDependOn;
```

that will check whether the type variable passed as an argument is present within the type. If it is, then an exception must be thrown. For instance if you passed the `mustNotDependOn` method of (the representation of) $(\alpha \rightarrow \beta) \rightarrow (\beta \rightarrow \gamma)$ the type-variable δ the method would just return, while if you passed it α, β or γ there would be an exception. [9 marks]

(*c*) Give an implementation of a method

```
toString()
```

which returns a text representation of the `MLType` object concerned. [4 marks]

SECTION E

11 Algorithms

(*a*) Describe how Quicksort works, using the following array of numbers as an example: $16, 42, 22, 7, 15, 3$. [8 marks]

(*b*) What are the worst-case time and space complexities of Quicksort, and how can this worst-case behaviour occur? [3 marks]

(*c*) Explain how the worst-case behaviour of Quicksort can be avoided to yield an improved time complexity. Would the changes you describe be useful in practice? [9 marks]

12 Algorithms

(*a*) How do insertions and deletions in a 2-3-4 tree retain the structure's perfect balance? [2 marks]

(*b*) Explain the structural relationship between 2-3-4 trees and red-black trees. [4 marks]

(*c*) Draw diagrams to illustrate left and right rotations at the root node of a binary search tree. Label the positions of all subtrees before and after the rotation. [4 marks]

(*d*) Write pseudocode for a recursive function `move_to_root(x,k)` which, given a binary search tree with root node x and a key value k, uses a sequence of rotations to move the node with key value k to the root of the tree and returns a pointer to the new root node. [10 marks]

END OF PAPER