COMPUTER SCIENCE TRIPOS Part IA

Monday 6 June 2005 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 Coversheets
Tags

SECTION A

1 Foundations of Computer Science

It is required to implement polynomials, which may contain any number of different variables, with the operations of *addition* and *equality test*.

- Computing that the sum of $xy^2 + 2xz - y$ and $y + 4xw - xz$ is $xz + xy^2 + 4xw$ is an example of polynomial addition.

- Determining that $xy + z^2$ equals $zz + yx$ is an example of an equality test.

(*a*) Design a suitable representation of polynomials, in ML. Justify the choices you make. [6 marks]

(*b*) Describe how you would implement addition and the equality test, and express their efficiency using O-notation. It is not necessary to present ML code. [4 marks]

2 Discrete Mathematics

(*a*) State the Fermat–Euler theorem, and deduce that $p \mid (2^p - 2)$ for any prime p. [5 marks]

(*b*) A composite number m that satisfies $m \mid (2^m - 2)$ is known as a *pseudo-prime*.

Show that $2^{10} \equiv 1 \pmod{11}$ and $2^{10} \equiv 1 \pmod{31}$. Deduce that 341 is a pseudo-prime. [5 marks]

3 Programming in Java

(*a*) Describe the four access regimes from `public` to `private` that may be applied to Java fields and methods. Why are they useful? [4 marks]

(*b*) When you extend a class, the constructor for your new class will reference the constructor of the parent class, and this latter constructor may have any of the four possible access regimes. Comment on the consequences of each of the four possibilities. [4 marks]

(*c*) If the only constructor for a class is marked as `private`, is it ever possible to have an instance of that class or any subclass of it? Explain why or why not. [2 marks]

4 Operating Systems

(*a*) What is the *address binding* problem? [1 mark]

(*b*) The address binding problem can be solved at compile time, load time or run time. For *each* case, explain what form the solution takes, and give *one* advantage and *one* disadvantage. [3 marks each]

SECTION B

5 Foundations of Computer Science

(*a*) Explain the operation of the Quicksort algorithm. Illustrate your answer by applying it to the list $[8, 3, 6, 12, 2, 9, 20, 1, 5, 0, 7, 13, 4, 11, 10]$. [6 marks]

(*b*) Write a Standard ML function for finding the median of three integers. [3 marks]

(*c*) A variant of Quicksort uses a novel method of choosing the pivot element. Instead of using the head of the list, it uses the median of the first, middle and last elements of the list. Express this algorithm in ML. You may assume the existence of the function `length`, but you may not assume that the items being sorted are distinct. [7 marks]

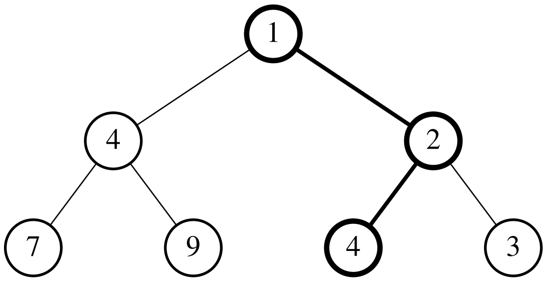
(*d*) What is the average-case execution cost, measured in terms of the number of comparisons, for the version of Quicksort described in part (*c*) above? Justify your answer carefully. [4 marks]

6 Foundations of Computer Science

Consider a datatype of binary trees where both leaves and branches carry labels:

```
datatype 'a tree = Twig of 'a
                 | Br of 'a * 'a tree * 'a tree;
```

A *path* in a binary tree is a series of labels proceeding from the root to a leaf, as shown in the diagram:



Consider the problem of finding a path in a binary tree such that the integer sum of the labels satisfies a given property. (In the example above, the highlighted path sums to a prime number.)

(*a*) Write an ML function `find_path` such that `find_path p t` returns some path in `t` whose sum satisfies the boolean-valued function `p`. If no such path exists, the function should raise an exception. [5 marks]

(*b*) Write an ML function `all_paths` such that `all_paths p t` returns the list of all paths in `t` whose sums satisfy the boolean-valued function `p`. [6 marks]

(*c*) Write an ML function `all_pathq` that is analogous to `all_paths` but returns a lazy list of paths. For full credit, your function should find paths upon demand rather than all at once. [Hint: try adding solutions to an accumulating argument.] [9 marks]

SECTION C

7 Discrete Mathematics

(*a*) State and prove the Chinese Remainder Theorem concerning the simultaneous solution of a pair of congruences to co-prime moduli, and the uniqueness of that solution. [10 marks]

(*b*) Define U_n (the set of units modulo n) and $\varphi(n)$ (Euler's totient function). [2 marks]

(*c*) Given natural numbers m and n with no common factors, define $f : \mathrm{U}_{mn} \longrightarrow \mathrm{U}_m \times \mathrm{U}_n$ by $f(u) = (u \bmod m, u \bmod n)$. Prove carefully that f is a bijective function. [6 marks]

(*d*) Deduce that φ is multiplicative, and calculate $\varphi(175)$. [2 marks]

8 Discrete Mathematics

(*a*) (*i*) Define the terms *injection*, *surjection* and *bijection*. [2 marks]

(*ii*) Let A, B and C be sets. Define a bijection from $[(A \times B) \to C]$ to $[A \to (B \to C)]$. Define its inverse. [2 marks]

(*b*) Prove that

(*i*) if A and B are countable sets, then their product $A \times B$ is also countable; [2 marks]

(*ii*) if A and B are countable sets, then their union $A \cup B$ is countable; [2 marks]

(*iii*) the powerset of the natural numbers $\mathcal{P}(\mathbb{N})$ is uncountable; [5 marks]

(*iv*) the set of finite subsets of $\mathbb{N}$ is countable. [2 marks]

[You may assume that a set B is countable iff there is an injection from B into $\mathbb{N}$.]

(*c*) Explain why assuming that the collection $\{x \mid x \text{ is a set}\}$ is a set would lead to a contradiction. [3 marks]

(*d*) By considering the set $B = \mathbb{R} \cup \{b\}$, where $b \notin \mathbb{R}$, define a well-founded relation $\prec$ on B such that $\{x \in B \mid x \prec b\}$ is uncountable. [2 marks]

SECTION D

9 Programming in Java

The following code has been written by a novice Java programmer. You are not required either to understand or to debug the details of how this code draws some particular pattern (a "Dragon").

The programmer finds that the Java compiler complains. Identify any errors and comment on any issues that (even if not strictly invalid Java) are liable to cause problems. You are not required to provide corrections.

```
**********************************
** Supervison work for June 6th. **
**********************************
upper class Dragon extends JApplet throws Exception
{
  import javax.swing.*;

  public paint(Graphics g)
  {
    this.g = g;
    drawDragon(DRAWDEPTH,100,200,300,200);
  }

  /** @title: drawDragon
    Function to draw a dragon curve between too points
    (x1,y1) and (x2,y2) with depth 'depth'.
    */
  void protected drawDargon(int depth,
      int x1,int y1,int x2,int y2)
  {
    if (x1 < 0 | x2 < 0)
       if (y1 < 0) raise new Exception("X & Y < 0");
    else assert("Ok so far");
    if (depth = 0)    // bottom of recursion
    {   g.drawLine(x1,y1,x2,y2);
        continue;
    int mpx=(x2+x1+y2-y1)/2; /* X coord of a new point...
    int mpy=(y2+y1-x2+x1)/2;    ... and Y coord. */
    printf("DEBUG: x=    drawDragon(depth-1,mpx,mpy,x1,y1);
    drawDragon(depth+1,mpx,mpy,x2,y2);
  }

  static secret int DRAWDEPTH=15,
                Graphics g;
}
```

[20 marks]

10 Programming in Java

The naïve way of computing a value to the power n performs $n-1$ multiplications. A much better algorithm, discussed in some detail in the Foundations of Computer Science course, can involve repeated squaring:

```
pow(a, 1) = a
pow(a, n) = pow(a*a, n/2)       [n even]
          = a * pow(a*a, (n-1)/2) [odd]
```

Even this is not always optimal. For example, it computes a^{15} via the sequence $a, a^2, a^3, a^6, a^7, a^{14}, a^{15}$ in 6 steps while it is possible to use $a, a^2, a^4, a^5, a^{10}, a^{15}$ and manage in 5 steps.

Design a Java program that can be given an integer n and will find (by some form of exhaustive search) the smallest number of multiplications that could be used to raise a value to the power n.

The strategy to use is as follows:

Start with the value a before undertaking any multiplication. The first multiplication must therefore be of a by itself, so after one multiplication you have a, a^2 calculated. The second multiplication could multiply a^2 either by itself or by a (there is no point in multiplying a by itself again!) so the situation you reach has either a, a^2, a^4 or a, a^2, a^3 calculated. You will carry on this building up a collection of sets representing the collections of values you might have available after $3, 4, 5, \ldots$ multiplications, and you will use exceptions to exit as soon as you find that one of these collections includes the target exponent n.

The standard Java library class `HashSet` may prove helpful. It supports a method `add` for adding an item, `contains` for checking membership, `equals` for comparing `HashSet`s, and the Java `for (var : collection)` style of iteration can be used to look at all its elements in turn.

It is not expected that your code is fast enough for seriously large values of n.

Minor syntax errors or mistakes about the Java libraries will not be heavily penalised, but code that is without any explanation and that is poorly laid out will not gain many marks.

[20 marks]

SECTION E

11 Operating Systems

(*a*) Describe with the aid of a diagram the on-disk layout of a UNIX V7 filesystem. Include in your description the role of the *superblock*, and the way in which free inodes and data blocks are managed. [6 marks]

(*b*) Describe with the aid of a diagram a UNIX V7 *inode*. [6 marks]

(*c*) Estimate the largest file size supported by a UNIX V7 filesystem. [2 marks]

(*d*) Suggest *one* reliability enhancement and *two* performance enhancements which could be made to the UNIX V7 filesystem. [2 marks each]

12 Operating Systems

(*a*) Describe with the aid of a diagram the structure of the Windows XP operating system. Sketch the functions of each component, and clearly indicate which parts execute in kernel mode and which in user mode. [6 marks]

(*b*) Compare and contrast the *object namespace* in Windows XP with the *directory namespace* of UNIX. [4 marks]

(*c*) Compare and contrast *blocking*, *non-blocking* and *asynchronous* I/O. [6 marks]

(*d*) Give *four* techniques which can improve I/O performance. [1 mark each]

END OF PAPER