COMPUTER SCIENCE TRIPOS Part IA

NATURAL SCIENCES TRIPOS Part IA (Paper CS/1)
PSYCHOL. AND BEHAVIOURAL SCIENCES TRIPOS Part I (Paper CS 1)

Monday 2 June 2014 1.30 to 4.30 pm

COMPUTER SCIENCE Paper 1

*Answer **one** question from each of Sections A, B and C, and **two** questions from Section D.*

*Submit the answers in five **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
Rough work pad

SPECIAL REQUIREMENTS
Approved calculator permitted

SECTION A

1 Foundations of Computer Science

(a) Write brief notes on polymorphism in ML, using lists and standard list functions such as `@` (append) and `map`. [4 marks]

(b) Explain the meaning of the following declaration and describe the corresponding data structure, including the role of polymorphism.

```
datatype 'a se = Void | Unit of 'a | Join of 'a se * 'a se;
```

[4 marks]

(c) Show that ML lists can be represented using this datatype by writing the functions `encode_list` of type `'a list -> 'a se` and `decode_list` of type `'a se -> 'a list`, such that `decode_list (encode_list xs) = xs` for every list `xs`. [3 marks]

(d) Consider the following function declaration:

```
fun cute p Void = false
  | cute p (Unit x) = p x
  | cute p (Join(u,v)) = cute p u orelse cute p v;
```

What does this function do, and what is its type? [4 marks]

(e) Consider the following expression:

```
fn p => cute (cute p)
```

What does it mean, and what is its type? Justify your answer carefully. [5 marks]

2 Foundations of Computer Science

(a) Write brief notes on the queue data structure and how it can be implemented efficiently in ML. In a precise sense, what is the cost of the main queue operations? (It is not required to present ML code.) [6 marks]

(b) Run-length encoding is a way of compressing a list in which certain elements are repeated many times in a row. For example, a list of the form $[a, a, a, b, a, a]$ is encoded as $[(3, a), (1, b), (2, a)]$. Write a polymorphic function `rl_encode` to perform this encoding. What is the type of `rl_encode`? [6 marks]

(c) The simple task of testing whether two lists are equal can be generalised to allow a certain number of errors. We consider three forms of error:

- *element mismatch*, as in [1,2,3] versus [1,9,3] or [1,2,3] versus [0,2,3]
- *left deletion*, as in [1,3] versus [1,2,3] or [1,2] versus [1,2,3]
- *right deletion*, as in [1,2,3] versus [1,3] or [1,2,3] versus [1,2]

Write a function `genEquals n xs ys` that returns `true` if the two lists `xs` and `ys` are equal with no more than `n` errors, and otherwise `false`. You may assume that n is a non-negative integer. [8 marks]

All ML code must be explained clearly and should be free of needless complexity.

SECTION B

3 Object-Oriented Programming

(*a*) (*i*) Explain the purpose of access modifiers in OOP languages. [2 marks]

(*ii*) Copy and complete the table below to show the access restrictions for the four access modifiers in Java. [2 marks]

Access Modifier				
Defining class Class in same package Subclass in different package Non-subclass in different package				

(*b*) A Java game designer wishes to store all the game preferences (e.g., player name, screen size, music volume, etc.) within a custom `Preference` class.

(*i*) Assuming each preference is stored as a unique `String` key mapping to a `String` value, give a simple implementation of `Preference` that allows for efficiently setting or updating preferences and retrieving previously set ones. Your implementation should define an exception that is thrown when a preference key is requested but not present. [5 marks]

(*ii*) It is important that only one `Preference` object exists in a running game. Show how to apply access modifiers and the Singleton design pattern to ensure this. Your implementation should lazily instantiate the object. Is it necessary to make your class `final` or `Cloneable`? Explain your answer. [6 marks]

(*c*) The designer also implements other Singleton classes in the game and proposes to create a `SingletonBase` base class from which all such classes would inherit the singleton behaviour. By providing example Java code, explain why this is not viable. [5 marks]

4 Object-Oriented Programming

A Lecturer wishes to create a program that lists his students sorted by the number of practical assignments they have completed. The listing should be greatest number of assignments first, sub-sorted by name in lexicographical order (A to Z).

A class `StudentInfo` stores the name and number of assignments completed for a student. Amongst other methods, it contains a `void setCompleted(int n)` method that allows changes to the number of completed assignments.

(*a*) Provide a definition of `StudentInfo` with an `equals()` method and a natural ordering that matches the given requirement. [9 marks]

(*b*) A `TreeSet` is used to maintain the `StudentInfo` objects in appropriate order. When `setCompleted(...)` is called on a `StudentInfo` object it is necessary to remove the object from the set, change the value and then reinsert it to ensure the correct ordering. This is to be automated by applying the Observer design pattern via classes `UpdatableTreeSet` and `SubscribableStudentInfo`. A partial definition of `UpdatableTreeSet` is provided below.

```
public class UpdatableTreeSet extends
             TreeSet<SubscribableStudentInfo> {
  // To be called just before the StudentInfo object is updated
  public void beforeUpdate(SubscribableStudentInfo s) {
    remove(s);
  }
  // To be called just after the StudentInfo object is updated
  public void afterUpdate(SubscribableStudentInfo s) {
    add(s);
  }
}
```

(*i*) Extend `StudentInfo` to create `SubscribableStudentInfo` such that: multiple `UpdatableTreeSet` objects can subscribe and unsubscribe to receive updates from it; and the `beforeUpdate(...)` and `afterUpdate(...)` methods are called appropriately on the subscribed `UpdatableTreeSet` objects whenever `setCompleted(...)` is called. [6 marks]

(*ii*) Give a complete definition of `UpdatableTreeSet` that overrides the inherited methods `boolean add(SubscribableStudentInfo)` and `boolean remove(Object)` to automatically subscribe and unsubscribe to their arguments, as appropriate. You may ignore all other methods inherited from `TreeSet`. [5 marks]

SECTION C

5 Numerical Methods

(a) Floating point representation:

(i) If a single-precision floating point number is added to a double-precision floating point number, what can we say about the expected and worst-case representation errors in the result? [2 marks]

(b) Floating point rounding:

(i) Would a *round-to-odd* rule introduce a different amount of bias compared with the normally used *round-to-even* rule? [2 marks]

(ii) What is it about counting in base 10 that causes bias in the standard approach used when rounding to a given number of significant figures? Explain whether a similar rule for numbers expressed in base 4 would introduce a bias. [4 marks]

(c) Matrix multiplication conditioning:

(i) When a pair of matrices of dimension $N \times N$ are multiplied, what is the expected and worst case representation error in the result? [2 marks]

(ii) Before using Gaussian Elimination to achieve triangle form when solving a system of simultaneous equations, what step or steps can be taken to ensure numerical stability? [4 marks]

(iii) What is meant by backwards stability regarding a numerical method? [2 marks]

(iv) What role can partial derivatives play in examining the stability of a computation compared with using a *condition number*? [4 marks]

6 Numerical Methods

(*a*) A lander for the planet Mars has initial mass $M_0 = m(0)$ kilograms which includes $F_0 = f(0)$ kilograms of fuel. It is released from an orbiter at time zero at height $H_0 = h(0)$ with an initial downwards velocity of zero. It must touch down at less than 1 metre per second. Its downward force is Mg (where g is constant) and this is countered by a rocket motor that is pre-programmed to generate a time-varying upwards force of $u(t)$. The motor burns fuel at a mass rate proportional to the force it develops. This is summarised in these equations:

$$\frac{dm(t)}{dt} = -\alpha u(t) \qquad \frac{dv(t)}{dt} = \frac{u(t) - gm(t)}{m(t)} \qquad \frac{dh(t)}{dt} = v(t)$$

A discrete-time computer simulation of the landing uses time steps Δt. Using a programming language of your choice or pseudo code:

(*i*) Give a suitable state vector for the system. Include setup code that suitably initialises the state vector. [1 mark]

(*ii*) Give state update assignments for one time step based on simple linear projections assuming a function `u(t)` has been provided. [2 marks]

(*iii*) Give code for the various stopping conditions. These include a safe landing, a fatal crash or running out of fuel. [1 mark]

(*iv*) Why does a simple linear projection lead to a velocity modelling error in every time step. What determines the error magnitude and does it compound over successive steps? [3 marks]

(*b*) (*i*) Briefly describe the bisection method (binary chop) for finding a root of an equation. Mention two possible stopping conditions. [3 marks]

(*ii*) Recall that the CORDIC algorithm uses successive approximation where the ith division of the interval is by $\arctan(2^{-i})$. Give a stopping condition for CORDIC. [2 marks]

(*iii*) The following approximation can be used for cosine: $\cos(x) \approx 1 - \frac{x^2}{2}$. Does it accurately deliver three significant decimal digits where the argument range is 0.0 to $\pi/4$? [2 marks]

(*iv*) Approximately how many iterations of CORDIC are required to ensure three significant decimal digits are accurate over the same range? [6 marks]

SECTION D

7 Algorithms

(*a*) Consider the radix sort algorithm.

(*i*) Explain how radix sort works, to what inputs it can be applied and what its asymptotic complexity is. [5 marks]

(*ii*) Explain why running radix sort does not proceed from most to least significant digit, as would at first seem more intuitive. [4 marks]

(*iii*) Give a proof by induction of the correctness of radix sort. [4 marks]

(*b*) Clearly describe an algorithm, strictly better than $O(n^2)$, that takes a positive integer s and a set A of n positive integers and returns a Boolean answer to the question whether there exist two distinct elements of A whose sum is exactly s. Evaluate its complexity. [7 marks]

8 Algorithms

(a) Explain an efficient method to find the k-th smallest number in a set of n numbers (output: one number), without first sorting the n numbers, and discuss its complexity in terms of n and k. [4 marks]

(b) Explain an efficient method to find the k smallest numbers in a set of n numbers (output: k numbers), without first sorting the n numbers, and discuss its complexity in terms of n and k. How much extra work is needed compared to (a)? [4 marks]

(c) Draw four distinct binary search trees (BSTs) for the following set of keys: $\{1, 2, 3, 4\}$. [2 marks]

(d) Let a height-balanced BST (hBST) be a BST with the additional defining invariant that each node is the parent of subtrees whose heights differ by at most 1. Give an efficient procedure to insert into an hBST and prove that the defining invariant is preserved. [10 marks]

9 Algorithms

(a) Explain the terms *amortized analysis*, *aggregate analysis* and *potential method*. [6 marks]

(b) Consider an arbitrary sequence of n stack operations `PUSH()`, `POP()` and `MULTIPOP(x)` in which `POP()` or `MULTIPOP(x)` never attempt to remove more elements than there are on the stack. Assuming that the stack begins with s_0 items and finishes with s_n items, determine the worst-case total cost for executing the n operations as a function of n, s_0 and s_n. You may assume `PUSH()` and `POP()` cost 1 each and `MULTIPOP(x)` costs x. [5 marks]

(c) Suppose we want to store a number of items in an array, but we do not know in advance how many items need to be stored. The `INSERT(x)` operation appends an item x to the array. More precisely, if the size of the array is large enough, x is inserted directly at the end of the array. Otherwise, a new array of larger size is created that contains all previous items with x being appended at the end. The total cost of `INSERT(x)` is 1 in the first case, and the size of the new array in the second case.

(i) Devise a strategy which, for any integer n, performs any sequence of n `INSERT(.)` operations at a total cost of $O(n)$. [5 marks]

(ii) For the strategy described in (c)(i), give a proof of the cost of the algorithm using the potential method. [4 marks]

10 Algorithms

(*a*) Given any directed graph $G = (V, E)$ with non-negative edge weights, consider the problem of *all-pairs shortest path* (APSP). Give the asymptotic runtimes of the following four algorithms when applied (directly or iterated) to the APSP problem as a function of $|V|$ and $|E|$, and provide a brief justification for your answer: Bellman-Ford, Dijkstra, matrix multiplication and Johnson. [8 marks]

(*b*) Consider the problem of converting currencies modelled by a directed graph $G = (V, E)$ with $|V|$ vertices representing currencies and $|E|$ directed edges (u, v) each of which has a strictly positive weight $w(u, v) > 0$ representing the exchange rate. For instance, for any real number x, we have x USD $= w(\text{dollars, pounds}) \cdot x$ GBP. Our goal is, given a pair of currencies $s, t \in V$, to find the least expensive way of exchanging from s to t, possibly by using more than one exchange.

(*i*) How could you transform the graph by reweighting the edges so that the problem could be solved with a shortest path algorithm? Indicate which shortest path algorithm is used. [8 marks]

(*ii*) How would you deal with negative-weight cycles if they occurred in the transformed graph? Give the perspective of the currency trader as well as that of a computer scientist. [4 marks]

END OF PAPER