UNIVERSITY OF CAMBRIDGE

COMPUTER SCIENCE TRIPOS Part IA

NATURAL SCIENCES TRIPOS Part IA (Paper CS/1)

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

Monday 1 June 2015 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 about a tree representation of functional arrays, subscripted by positive integers according to their representation in binary notation. How efficient are the lookup and update operations? [6 marks]

(*b*) Write an ML function `arrayoflist` to convert the list $[x_1, \ldots, x_n]$ to the corresponding functional array having x_i at subscript position i for $i = 1, \ldots, n$. Your function should not call the update operation. [6 marks]

(*c*) Consider the task of finding out which elements of an array satisfy the predicate `p`, returning the corresponding subscript positions as a list. For example, the list $[2, 3, 6]$ indicates that these three designated array elements, and no others, satisfy `p`. Write an ML functional to do this for a given array and predicate, returning the subscripts in increasing order. [8 marks]

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

2 Foundations of Computer Science

(*a*) Write brief notes on programming with lazy lists in ML. Your answer should include the definition of a polymorphic type of infinite lazy lists, a function to return the tail of a lazy list, a function to create the infinite list of all positive integers, and an apply-to-all functional analogous to the list functional `map`. [6 marks]

(*b*) Write a function `diag` that takes a lazy list of lazy lists,

$$\begin{array}{l}[\,[z_{11}, z_{12}, z_{13}, \ldots],\\ \;[z_{21}, z_{22}, z_{23}, \ldots],\\ \;[z_{31}, z_{32}, z_{33}, \ldots], \ldots\,]\end{array} \qquad (*)$$

and returns the diagonal, namely the lazy list $[z_{11}, z_{22}, z_{33}, \ldots]$. [3 marks]

(*c*) Write a function that takes two lazy lists $[x_1, x_2, x_3, \ldots]$ and $[y_1, y_2, y_3, \ldots]$ and a function `f` of two arguments; it should return a lazy list of lazy lists like $(*)$ above, with $z_{ij} = \mathtt{f}\, x_i\, y_j$. [3 marks]

(*d*) Write a function that converts a lazy list of lazy lists like $(*)$ above to a lazy list whose elements are all of the z_{ij}, enumerated in some order. [8 marks]

SECTION B

3 Object-Oriented Programming

The Java class below defines a queue data structure with a fixed capacity.

```
public class FixedCapacityQueue {
        int[] mData = null;
        int   mHead=0; // index of first full cell
        int   mTail=0; // index of next empty cell

        public FixedCapacityQueue(int capacity) {
                mData = new int[capacity];
        }

        public boolean enqueue(int x) {
                if (mTail==mData.length) return false;
                mData[mTail]=x;
                mTail++;
                return true;
        }

        public int dequeue() {
                if (mTail==mHead) return -1;
                int v=mData[mHead];
                mHead++;
                return v;
        }
 }
```

(a) Explain the risks posed by the lack of access modifiers specified on the state. Which access modifier is appropriate for these entities? [3 marks]

(b) Discuss the choice to signal enqueue and dequeue errors using return values of false and -1, respectively. Suggest a better alternative and modify `enqueue` and `dequeue` to use it. [5 marks]

(c) Explain how an enqueue operation on a `FixedCapacityQueue` object could fail with an error even when there are fewer items in the queue than the assigned capacity of the object. Show how to fix this. [3 marks]

(d) Rewrite the class to use Java's Generics so that it can be used to represent queues of arbitrary objects (`Integer`s, `String`s, etc). Use the `java.util.ArrayList` class instead of `int[]` for the type of `mData`. [4 marks]

(e) Explain why it was necessary to replace the `int[]` type of `mData` in part (d), and why `ArrayList` does not suffer from the same issue. [5 marks]

4 Object-Oriented Programming

A developer wishes to create a photo library application that can auto-detect faces in images to assist labelling. They wish to use some commercially available Java face detection software. The source to this software is unavailable—only the Java bytecode and documentation are supplied. The two key classes are as follows:

Class name	Method prototype

`Image`	// Constructor
	`public Image(byte[] imageData, int w, int h)`
	// Get the image value at pixel (x,y)
	`public byte getPixel(int x, int y)`
	// Set the image pixel at (x,y) to value val
	`public void setPixel(int x, int y, byte val)`
	// Get the image width
	`public int getWidth()`
	// Get the image height
	`public int getHeight()`

`Algorithm`	// Search for and mark the faces in Image im `public static void markFaces(Image im)`

(*a*) Explain what is meant by source code, machine code and Java bytecode. Give two advantages of distributing Java software as bytecode rather than source code. [5 marks]

(*b*) Explain why `markFaces()` is declared `static`. [1 mark]

(*c*) The developer wishes to add filename information in the form of a `String` to all of the `Image` instances. Show how to achieve this efficiently using inheritance. [4 marks]

(*d*) The developer wishes to know whether the `markFaces()` method visits every pixel in the image or uses a more intelligent search strategy. Provide a new class definition that can be used to determine the search sequence. Your class should be able to *temporarily* augment an `Image` object with logging capabilities while it is being processed by `markFaces()`.

[*Hint:* You may find it useful to apply a common design pattern.]

[10 marks]

SECTION C

5 Numerical Methods

(a) Consider the iteration:

$$x_{n+1} = (2x_n + N/x_n^2)/3$$

(i) The iteration converges to give what useful property of the constant argument N? [2 marks]

(ii) Examine whether the above iteration should work for all possible values of N and x_0? [6 marks]

(iii) Find the order of convergence for the above iteration. You may use standard results but do not simply state an order without justification. [3 marks]

(b) Cholesky provides an approach to solving certain systems of simultaneous equations. His method (and similar methods) perform upper/lower triangle decomposition of the equation coefficient matrix A such that $A = LU$ and $U^T = L$.

(i) Under what circumstances can Cholesky's method be used? Can it be used if A is already a triangular matrix and, if not, what should be done instead? [3 marks]

(ii) Give expressions for two of the four values in the upper-left 2×2 sub-matrix of L in terms of the elements of A. [2 marks]

(iii) When is Cholesky's method preferred over general Gaussian Elimination and what advantage does it provide? [3 marks]

(iv) Why might the decomposition $A = LDU$ be preferable to an LU decompostion given that D is a diagonal matrix? [1 mark]

6 Numerical Methods

(*a*) (*i*) How and why is normalisation performed on a floating-point number at the end of an arithmetic operation? [4 marks]

(*ii*) Assuming both operands are normalised and nonzero, what is the maximum degree (number of shifts) of normalisation required in the result of a floating point multiplication? Ignore zero and denormal results. [5 marks]

(*b*) A watertight capsule containing scientific instruments is dropped from a high-altitude plane into the ocean where it must approach close to the sea bottom before floating up for retrieval by a boat. You are provided with subroutines that compute water and air resistance from velocity. Collision with the ocean bottom is a mission failure.

(*i*) Set up a state vector and sketch code for a finite-difference, time-domain simulation of the trajectory in one dimension. Include a check for mission failure. [5 marks]

(*ii*) What three issues will affect your choice of simulation timestep? [3 marks]

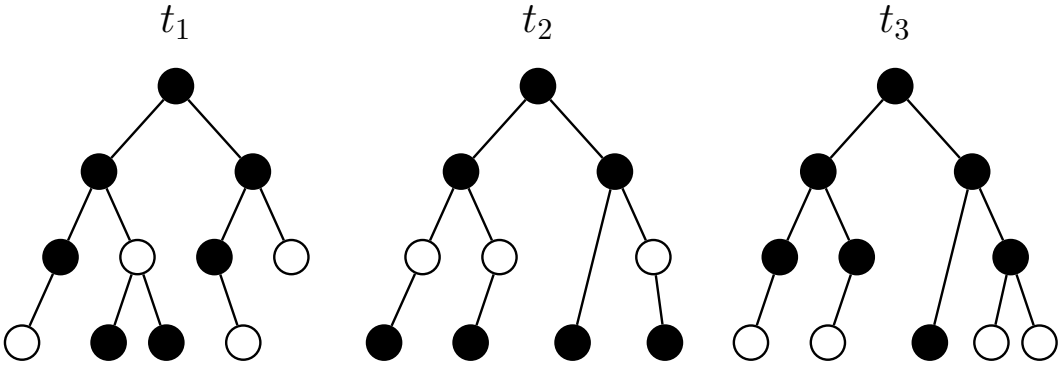
(*iii*) Your simulation is now re-coded using interval arithmetic. New subroutines for resistance are provided with interval domain and range giving a pair with the maximum and minimum resistance likely to be encountered. All input parameters are likewise coded as pairs. Summarise how this will affect the state vector and your code. [3 marks]

SECTION D

7 Algorithms

Reminders: A red-black tree has leaf nodes (black) and may have non-leaf nodes (red or black). The height of a red-black tree is the number of edges on the longest path from the root to any leaf.
Mathematical hint: $\sum_{j=1}^{k} 2^{-j} = 1 - 2^{-k}$

(a) Indicate whether each of the following trees is or is not a valid red-black tree. Justify your answers with reference to the defining invariants of red-black trees. You may, but do not have to, redraw the trees if it helps you clarify a point. [8 marks]



(b) Let $r(h), b(h), l(h)$ respectively represent the number of red non-leaf nodes, the number of black non-leaf nodes, and the number of leaf nodes in a red-black tree as a function of the height h of the tree. Under each of the conditions stated below, and assuming that the tree has as few red nodes as possible, derive mathematical expressions for $r(h), b(h), l(h)$, preferably in closed form. Clearly justify your answers, with drawings if appropriate. Expressions that are valid only for even (or odd) values of h can still earn full marks if properly derived and explained.

(i) Derive the $r(h), b(h), l(h)$ expressions assuming that the red-black tree has the largest possible number of nodes for a given height h. [6 marks]

(ii) Derive the $r(h), b(h), l(h)$ expressions assuming that the red-black tree has the smallest possible number of nodes for a given height h. [6 marks]

8 Algorithms

(*a*) Explain the greedy strategy in algorithm design. To what problems does it apply? [3 marks]

(*b*) If a problem can be solved with both dynamic programming and a greedy algorithm, what are the advantages of using one or the other? [2 marks]

(*c*) An imaginary post office machine must issue decorative stamps adding up to a given amount of p pence. Its goal is to minimize the number of postage stamps issued, and the machine always has as many stamps as needed.

(*i*) Let the set of available denominations for the stamps be $D = \{1\text{p}, 5\text{p}, 25\text{p}, 50\text{p}, £1, £2\}$. Can this problem be solved using bottom-up dynamic programming? If so, clearly describe your algorithm and determine its complexity. If not, prove that it cannot be done. [5 marks]

(*ii*) Let $c_1 < c_2 < \cdots < c_n$ be n stamp denominations. Prove that if each c_i (a positive integer) is a multiple of c_{i-1} for every $i = 2, \ldots, n$ then the greedy strategy applied to the set $D = \{c_1, c_2, \cdots, c_n\}$ finds the optimal solution for any amount p that is a multiple of c_1. [7 marks]

(*iii*) Provide a set of denominations for stamps D and an amount of pence p for which the greedy strategy fails to give an optimal solution, p being a multiple of the smallest denomination in D. Show what solution the greedy strategy would find and what the optimal solution is. [3 marks]

9 Algorithms

(*a*) Consider the two standard representations of directed graphs: the adjacency-list representation and the adjacency-matrix representation. Find a problem that can be solved more efficiently in the adjacency-list representation than in the adjacency-matrix representation, and another problem that can be solved more efficiently in the adjacency-matrix representation than in the adjacency-list representation. [4 marks]

(*b*) Prove or disprove (by giving a counter-example) the following claim: If a directed graph G contains a path from a vertex u to a vertex v, then any depth-first search must result in $v.d \leq u.f$, where $.d$ is the discovery time and $.f$ the finishing time. [4 marks]

(*c*) We are given an undirected, connected graph $G = (V, E)$ with edge-weights $w : E \to \mathbb{R}^+$ and a minimum spanning tree T of G. How would you update your minimum spanning tree T in each of the following three cases? Specify the runtime of your algorithm and give a proof that the returned tree is indeed a minimum spanning tree.

(*i*) We increase the weight of an edge e which is not in T. [3 marks]

(*ii*) We decrease the weight of an edge e which is in T. [3 marks]

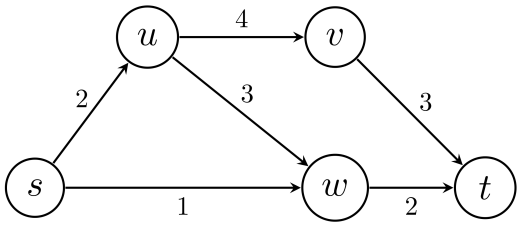
(*iii*) We add a new edge e with weight $w(e)$ to G. The weight $w(e)$ is arbitrary, but for simplicity you may assume that after adding the edge e no two edges in G have the same weight. [6 marks]

10 Algorithms

(*a*) State the Max-Flow Min-Cut Theorem. [2 marks]

(*b*) For an arbitrary integer $k \geq 1$, give an example of a flow network with at most five vertices on which the basic Ford-Fulkerson method takes at least k steps to terminate. [4 marks]

(*c*) Consider the following flow network G:



Given an initial flow f with $f(s,u) = f(u,w) = f(w,t) = 2$, perform one iteration of Ford-Fulkerson; that is, draw the residual graph G_f, specify an augmenting path in G_f, and update the flow f. Is this new flow a maximum flow? Justify your answer. [5 marks]

(*d*) Given an undirected, connected graph $G = (V, E)$, the edge-connectivity of G is the *size* of a smallest set of edges $X \subseteq E$ so that the graph $G' = (V, E \setminus X)$ becomes disconnected.

(*i*) Describe an algorithm that computes the edge-connectivity of G, and analyse its runtime and correctness. [7 marks]

(*ii*) Extend your algorithm so that it also returns a set X satisfying the conditions above. [2 marks]

END OF PAPER