CST0
COMPUTER SCIENCE TRIPOS Part IA

Tuesday 9 June 2026 14:00 to 17:00

COMPUTER SCIENCE Paper 1

*Answer **one** question from each of Sections A, B, C, D, and E.*

*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

SPECIAL REQUIREMENTS
Approved calculator permitted

SECTION A

1 Foundations of Computer Science

A Lindenmayer system is a type of formal grammar that can be used to generate self-similar fractals. It consists of three parts: an alphabet of symbols that can be assembled to form a 'sentence', an initial sentence S_0, and a set of rewrite rules. The rewrite rules are applied to sentence S_0 to produce the next sentence S_1, then they are applied to S_1 to produce S_2 and so on.

In this question we consider a Lindenmayer system with symbols $F, G, +, -$, and the following rewrite rules:

$$F \to F + G$$
$$G \to F - G$$

The symbols + and − are unchanged. The initial sentence is $S_0 = F$ so the first few sentences are therefore $S_1 = F + G$, $S_2 = F + G + F - G$.

(*a*) We will use a type `t` to represent the symbols and `t list` to represent a sentence. Define a suitable type `t` and a 'rewrite function' `rewrite : t -> t list` that applies the rewrite rules to a single symbol. [4 marks]

(*b*) Define a function `apply : ('a -> 'a list) -> 'a list -> 'a list` that takes a rewrite function and a sentence, and applies the rewrite function to obtain the next sentence. [4 marks]

(*c*) The symbols describe the movement of a robot on a grid. The state of the robot is completely defined by its current position and the direction of travel. It starts at the origin (0,0) and is facing in the positive y direction, or 'Up', such that moving forward one unit would leave it at position (0,1).

Symbol	Action
F, G	Move forward one unit in current direction
+	Turn right (90° clockwise)
-	Turn left (90° anticlockwise)

Using the types

```
type pos = int * int
type dir = Up | Left | Right | Down
type state = pos * dir
```

define `step : state -> t -> state` that performs one action. [6 marks]

(*d*) Define a function `run : t list -> (int * int) list` that, given a sentence, returns the coordinates visited executing the corresponding actions, including the start and end points. For example, the sentence $[F, +, G]$ should produce `[(0,0); (0,1); (1,1)]`. [6 marks]

2 Foundations of Computer Science

A certain encoding of data treats bit sequences as paths down a tree, where 0 means taking the left node, and 1 means taking the right node. When a leaf is reached, the token associated with the leaf is emitted and decoding starts again at the root for the next input digit.

Consider for example the following trees:

The bit sequence `11100` is decoded as "charlie","bob","alice" using the first tree and 2.718, 2.718, 2.718, 1.414 using the second.

Using the `seq` and `tree` types defined as follows

```
type 'a seq = Nil | Cons of 'a * (unit -> 'a seq)
type 'a tree =
  | Node of 'a tree * 'a tree
  | Leaf of 'a
```

the input bits are supplied as a `bool seq` where `true` represents 1 and `false` represents 0.

(*a*) Write a function `decode : 'a tree -> bool seq -> 'a seq` that decodes the supplied bit sequence and returns a sequence of the decoded tokens. You should assume that the input stream is infinitely long. [10 marks]

(*b*) Write a function `scan : ('a -> 'b -> 'b) -> 'a seq -> 'b -> 'b seq`, and show how this would be used with a suitable function and a sequence of type `float seq` to produce a running total, where each float in the output sequence is the sum of the floats seen so far in the input sequence. [6 marks]

(*c*) What do the types of `decode` and `scan` tell you about the relationships between the inputs and the results of those functions? [4 marks]

SECTION B

3 Object-Oriented Programming

(*a*) (*i*) Explain the practical motivation for introducing `default` methods within interfaces in Java. [2 marks]

(*ii*) Compare and contrast a Java interface containing `default` methods with an abstract class. Give a scenario where an abstract class would still be the preferred design choice. [4 marks]

(*b*) You have an existing class hierarchy consisting of an interface `Shape` and two concrete implementations: `Circle` (atomic) and `Group` (composite). A `Group` contains other `Shape`s in a `List<Shape>`. You wish to allow arbitrary operations to be defined **outside** of the `Shape` classes, such that new operations can be added without modifying the `Shape` source code later.

(*i*) Define the additional methods necessary in the `Shape` interface and the `Circle` and `Group` classes to support this new mechanism via the generic interface:

```
public interface ShapeOp<R> {
  R opShape(ShapeCircle s);
  R opGroup(Group g);
}
```

Your answer should define only the additional methods and not the original fields and methods, which you can assume are complete. [5 marks]

(*ii*) Explain why `opCircle` and `opGroup` should not have default implementations. [3 marks]

(*iii*) Using the mechanism you created in part (*b*)(*i*), implement a concrete class `BlueFilter` that implements `ShapeOp<List<Circle>>`. This class must traverse the shape hierarchy and return an immutable list containing only the `Circle` objects that return `Color.BLUE` when their `getColour()` method is called. [5 marks]

(*iv*) Explain why the `ShapeOp` interface cannot be instantiated using a Java Lambda expression (e.g., `x -> ...`). [1 mark]

4 Object-Oriented Programming

(*a*) Consider the following class definition.

```
public class User {
   private String username;
   public User(String uname) { username = uname; }
   public void setUsername(String uname) {username = uname; }
   @Override public boolean equals(Object o) { ... }
   @Override public int hashCode() { ... }
}
```

You may assume `equals()` and `hashCode()` are correctly implemented and both refer to `username`.

(*i*) Explain the contract between `equals()` and `hashCode()` and provide an example of why it is necessary. [3 marks]

(*ii*) Identify a critical flaw in the `User` class design regarding its use as a `Map` key. Illustrate your answer with an example of the flaw. [3 marks]

(*b*) Consider the following Java code snippet.

```
List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);
System.out.println("Start");
Integer result =
    numbers.stream()
           .filter(n -> {
              System.out.print("Filter(" + n + ") ");
              return n % 2 == 0; })
            .map(n -> {
              System.out.print("Map(" + n + ") ");
              return n * 10; })
            .findFirst()
            .orElse(0);
System.out.println("\nResult: " + result);
```

(*i*) Explain what is means for a Java Stream to be lazy. Distinguish between intermediate and terminal operations, with examples, and explain how laziness affects their execution. [6 marks]

(*ii*) The `findFirst()` method returns an `Optional<Integer>` rather than a raw `Integer`. Explain the purpose of the `Optional` class and why `orElse(0)` is used in this snippet. [2 marks]

(*iii*) Provide a detailed explanation of the code execution and state the exact text printed to the console when this code runs. [6 marks]

SECTION C

5 Introduction to Probability

Two persons have a meeting at a given time, and each will arrive at the meeting place with a delay between 0 and 1 hour, where all delays are equally likely and independent of each other. Let $X_1, X_2 \in [0,1]$ be the continuous random variables representing the delays of person 1 and 2, respectively.

(*a*) Calculate $\mathbf{E}[X_1 + X_2]$ and $\mathbf{V}[X_1 + X_2]$, justifying each step in your derivation carefully. [4 marks]

(*b*) What is $\mathbf{P}\left[X_1 \geq 0.5 \text{ hours} \,\middle|\, X_1 \geq 0.2 \text{ hours}\right]$? [2 marks]

(*c*) Assume now that the first person to arrive will wait for 15 minutes and will leave if the other has not yet arrived. We are interested in the event that the two persons meet.

(*i*) Formalise this event using mathematical notation. [2 marks]

(*ii*) Compute the probability of this event. [3 marks]

Consider now an arbitrary pair of random variables (X_1, X_2).

(*d*) State two different equivalent definitions for $\mathbf{Cov}[X_1, X_2]$. [2 marks]

(*e*) What is the relationship between $\mathbf{Cov}[X_1, X_2] = 0$ and X_1, X_2 being independent? [2 marks]

Assume now that X_1, X_2 are independent samples from the same continuous distribution (not necessarily uniform) over non-negative real numbers with finite mean μ. Let $U = \min(X_1, X_2)$ and $V = \max(X_1, X_2)$.

(*f*) Prove that U and V cannot be negatively correlated.
Hint: You may want to use the following identity among sets: $\{U, V\} = \{X_1, X_2\}$. [5 marks]

6 Introduction to Probability

An internet service provider has installed c modems to serve the needs of a population of n users ($n \geq c$). It is estimated that at a given time, each user will need a connection with probability p, independently of the others. Let X be the total number of users needing a connection at a given time.

(*a*) What is the distribution of X, and what is its variance? [2 marks]

(*b*) If there are more than c users needing a connection, only c can be served while the rest of them will be blocked. Derive an expression for the expected number of blocked users. [2 marks]

Remark: For the following two questions, you do *not* need to compute an explicit, i.e., numerical value.

(*c*) Assuming $n = 200$ and $p = 1/100$, use a suitable approximation to estimate the probability that at a given time exactly 15 users need a connection. [3 marks]

(*d*) Assuming $n = 200$ and $p = 1/2$, use a suitable approximation to estimate the probability that at a given time at least 80 users need a connection. [5 marks]

Assume now that $p \in (0, 1]$ is an unknown parameter and that the provider records the number of required connections at m randomly chosen times. Let $X_1, X_2, \ldots, X_m$ denote these samples.

(*e*) Can we find an unbiased estimator for p? Justify your answer. [2 marks]

(*f*) Can we find an unbiased estimator for $\mathbf{P}\,[\,\text{at most } c \text{ users need a connection}\,]$? Justify your answer. [3 marks]

(*g*) Can we find an unbiased estimator for $1/p^2$? Justify your answer. [3 marks]

SECTION D

7 Algorithms 1

(*a*) Let $T(n)$ be the recurrence relation defined by

$$T(n) = \begin{cases} 1 & n \leq 10 \\ 4T(n/4 - 2) + kn & \text{otherwise} \end{cases}$$

Is $T(n) \in O(n)$? Justify your answer. [4 marks]

(*b*) Find asymptotically tight lower and upper bounds for the recurrence relation

$$T(n) = \begin{cases} 1 & n = 1 \\ 8T(n/2) + n^2 & \text{otherwise} \end{cases}$$

[6 marks]

(*c*) Find an asymptotically tight upper bound for the recurrence relation

$$T(n) = \begin{cases} 1 & n \leq 5 \\ T(n-5) + n^2 & \text{otherwise} \end{cases}$$

and justify that it is asymptotically tight. [3 marks]

(*d*) Show that the recurrence relation

$$T(n) = \begin{cases} 1 & n = 1 \\ 2T(n/4) + \frac{n}{\sqrt{n}} & \text{otherwise} \end{cases}$$

is in $\omega(\sqrt{n})$ and $O(n^2)$. Is $T(n) \in o(n \lg n)$? [7 marks]

8 Algorithms 1

(*a*) A haulage company uses electric trucks to move goods from one warehouse to another. There are recharging points at the warehouses A and B, and at $n-1$ intermediate points. The recharging points, $A = r_0, r_1, \ldots, r_n = B$, are at increasing distances $d_0 = 0, d_1, d_2, \ldots, d_n$ from A. Trucks can travel for distance c on a full charge, and the distance between adjacent recharging points is less than c. To maximise time on the road, the haulage company asks you to find a way of getting from A to B that minimises the number of occasions when the trucks have to stop to recharge. Trucks always recharge at A before setting off. They wish to know which recharging points to use, not only the minimum number of recharges required. You decide to solve the problem using dynamic programming.

(*i*) State and explain a formula that expresses an optimal solution in terms of the optimal solutions to subproblems. [5 marks]

(*ii*) Devise an algorithm based on dynamic programming to solve the problem and state whether it is top-down or bottom-up. [6 marks]

(*iii*) Derive the big-Theta space and time complexities of your approach. [4 marks]

(*b*) A friend believes that this optimisation problem can be solved with a greedy algorithm. Are they correct? If so, explain a greedy algorithm. If not, explain why no greedy solution can exist. [5 marks]

SECTION E

9 Algorithms 2

(*a*) Let T be a minimum spanning tree of an undirected graph G, and let T' be any other minimum spanning tree of G. Prove that T and T' must have the same sets of distinct edge weights. [5 marks]

(*b*) Say whether or not it is the case that, if the minimum spanning tree of an undirected graph G is unique then every cut of G has a unique light edge crossing the cut. Provide a proof or counterexample. Is the converse true? Again, provide a proof or counterexample. [5 marks]

(*c*) The Bellman–Ford algorithm is executed on a weighted, directed graph that contains a negative weight cycle. The cycle goes undetected by the algorithm. How can this occur? Provide an example graph for which this occurs. [4 marks]

(*d*) Bellman–Ford relaxes every edge in the input graph $|V| - 1$ times.

(*i*) For what purpose do we perform one extra relaxation of every edge? [2 marks]

(*ii*) Explain whether it is necessary to perform the extra relaxation if no changes were made in one of the first $|V| - 1$ scans. [2 marks]

(*iii*) What optimisation might you suggest if no changes were made in one of the first $|V| - 1$ scans? [2 marks]

10 Algorithms 2

(a) Give a clear and concise description of an algorithm to determine one topological sort of a directed acyclic graph represented using adjacency lists. Include pseudocode. [6 marks]

(b) Derive the asymptotic running time of your algorithm from Part (a). [4 marks]

(c) Devise an algorithm to find the transitive closure of a directed acyclic graph that exploits the nature of this type of graph. [6 marks]

(d) Derive the asymptotic running time of your algorithm from Part (c). [4 marks]

END OF PAPER