CST0
COMPUTER SCIENCE TRIPOS Part IA

Tuesday 15 June 2021 11:30 to 14:30 BST

COMPUTER SCIENCE Paper 1

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

*Submit each question answer in a **separate** PDF. As the file name, use your candidate number, paper and question number (e.g., `1234A-p1-q6.pdf`). Also write your candidate number, paper and question number at the start of each PDF.*

You must follow the official form and conduct instructions for this online examination

SECTION A

1 Foundations of Computer Science

Sequences (lazy lists) and trees are fundamental types in functional programming. Here are definitions of sequences and trees with integer elements:

```
type iseq = Nil
          | Cons of int * (unit -> iseq)

type itree = Leaf of int
           | Branch of itree * itree
```

(*a*) In an *ascending* sequence such as $1, 3, 3, 7, \ldots$ each element is at least as large as the previous elements.

Given two ascending sequences, write a function `merge2` that produces a sequence of the elements of both in ascending order. For example, passing $1, 3, 3, 7, \ldots$ and $2, 4, 5, 9, \ldots$ to `merge2` should produce the sequence $1, 2, 3, 3, 4, 5, 7, 9, \ldots$. [5 marks]

(*b*) Sequences are considered to be equal if corresponding elements are equal.

(*i*) Define a function `equal_seq` that compares two sequences for equality. [5 marks]

(*ii*) Define sequences `s1` and `s2` for which `equal_seq s1 s2` does not terminate. [3 marks]

(*c*) The *fringe* of a tree is the left-to-right sequence of the values at the leaves. For example, the fringe of `Branch (Leaf 3, Branch (Leaf 10, Leaf 4))` is the sequence $3, 10, 4$.

(*i*) Define a function `fringe` that computes the fringe of a tree. Your function should have the following type:

```
val fringe : itree -> iseq
```

[5 marks]

(*ii*) Using the functions you have defined above or otherwise, write a function `equal_fringes` that determines whether two trees have equal fringes. [2 marks]

2 Foundations of Computer Science

A $W \times H$ matrix can be represented in OCaml by a *flat list*: a list that concatenates the rows in order. For each of the following alternative ways to represent a 2D matrix in OCaml:

- State the type T of the representation;
- Give a function `create w m: int -> float list ->` T that constructs the matrix of type T equivalent to the input flat list `m` with row width `w`;
- Give a function `get r c m: int -> int ->` T `-> float` that gets the element of the matrix `m` at row `r` and column `c`.
- State the asymptotic complexity of the `get` function in terms of W and H

(a) A list of lists. [5 marks]

(b) An array of arrays. [6 marks]

(c) A functional array of functional arrays. [9 marks]

Your answers may use the `List` module and assume this functional array code:

```
type 'a tree = Lf | Br of 'a * 'a tree * 'a tree;;
exception Subscript;;

let rec update = function
  | Lf, k, w ->
    if k = 1 then
      Br (w, Lf, Lf)
    else
      raise Subscript
  | Br (v, t1, t2), k, w ->
    if k = 1 then
      Br (w, t1, t2)
    else if k mod 2 = 0 then
      Br (v, update (t1, k / 2, w), t2)
    else
      Br (v, t1, update (t2, k / 2, w));;

let rec sub = function
  | Lf, _ -> raise Subscript
  | Br (v, t1, t2), 1 -> v
  | Br (v, t1, t2), k when k mod 2 = 0 -> sub (t1, k / 2)
  | Br (v, t1, t2), k -> sub (t2, k / 2);;
```

SECTION B

3 Object-Oriented Programming

A programmer is using a cut-down version of Java that does not support static fields or inheritance (neither extends nor implements). Static methods and static inner classes are still supported.

(*a*) What are the implications of this for static methods in terms of:

(*i*) access to static fields; and [1 mark]

(*ii*) access modifiers. [1 mark]

(*b*) Describe how a programmer might use a shared instance of an environment object to emulate static fields. Consider:

(*i*) sharing state between all instances of a class; [4 marks]

(*ii*) access-modifiers; and [6 marks]

(*iii*) initialisation. [4 marks]

(*c*) What are the drawbacks and benefits of your scheme compared to static fields? [4 marks]

4 Object-Oriented Programming

A program which decrypts files under the Swap Encryption Scheme by swapping pairs of characters is given below. Some code has been omitted and you do not need to understand the operation of the algorithm.

```
1   class Swapper extends Reader {
2       private final PushbackReader pushBack;
3       Swapper(PushbackReader p) { pushBack = p; }
4
5       @Override
6       public int read(char[] cbuf, int off, int len) {
7           int r = wrap.read(cbuf, off, len);
8           if (r % 2 == 1) { pushBack.unread(cbuf, off + --r, 1); }
9           for (int i = 0; i < r; i += 2) { swap(cbuf, i, i + 1); }
10          return r;
11      }
12  }
13
14  class Decryptor {
15      static List<String> read(String fileName) {
16          try (BufferedReader r = new BufferedReader(new Swapper(
17            new PushbackReader(new FileReader(fileName))))) {
18              return readLines(r);
19          }
20      }
21  }
```

(*a*) The four principles of object-oriented programming are encapsulation, abstraction, inheritance of code and polymorphism. Explain how the program above makes use of each of them, with reference to specific lines in the code.
[2 marks each]

(*b*) The program attempts to use `Swapper` as part of the Decorator pattern. What changes would you make to improve the design? [2 marks]

(*c*) Explain how this program demonstrates the open-closed principle. [2 marks]

(*d*) How would you change this implementation to allow users to specify an arbitrary operation to apply to pairs of characters (rather than just swapping them)?
[4 marks]

(*e*) Explain why this design does not satisfy the open/closed principle with respect to adding support for decrypting images. What are the implications of this for object-oriented program design? [4 marks]

SECTION C

5 Introduction to Probability

(a) A travel agency is surveying their customer satisfaction by randomly polling 300 of their customers. From experience, 80% of their customers are typically happy with their service. Let H be the number of happy customers in the current poll.

(i) Randomly polling 300 different customers, specify a suitable distribution for H, including its parameters, expected value and variance. [1 mark]

(ii) State a suitable approximation of H and specify its distribution including its parameters, and compute the expected value and variance. [2 marks]

(iii) Using the approximation from Part (a)(ii), what is the probability that more than 220 and fewer that 260 customers are happy in the current poll? [4 marks]

(iv) Now, let X be the proportion of customers that are happy in the current poll. Following your approximation from Part (a)(ii), give the distribution for X, including its parameters, expected value and variance. [3 marks]

(b) Let X and Y have a joint density function

$$f(x,y) = \begin{cases} cx & \text{if } 0 < y < x < 1, \\ 0 & \text{otherwise.} \end{cases}$$

(i) Find the value of the constant $c > 0$. [4 marks]

(ii) Find the marginal density functions of X and Y. [4 marks]

(iii) Are X and Y independent? Justify your answer. [2 marks]

6 Introduction to Probability

(a) A korfball player is practicing shots and has a 90% chance of scoring. Assume that their shots are independent of one another.

(i) Let S be the number of successful shots made in 200 attempts. Specify a suitable distribution for S including its parameters, and compute the expected value and variance. What is the probability mass function of S? [3 marks]

(ii) Following the experiment in Part (a)(i), let M be the number of shots made before the first miss. Specify a suitable distribution for M including its parameters, and compute the expected value and variance. What is the probability of $M > 100$? [4 marks]

(iii) Use a suitable distribution to approximate the probability that there are at most 3 misses in the first 200 shots. Note: you do not need to compute the final numerical value. [3 marks]

(b) Consider an urn containing balls labelled $0, 1, 2, \ldots, n-1$ and the experiment of drawing n of these balls uniformly and without replacement. Let X_i denote the label of the ball drawn in the i-th step, $1 \leq i \leq n$.

(i) For any $1 \leq i \leq n$, what is $\mathbf{E}[X_i]$ and $\mathbf{V}[X_i]$? Justify your answer. [2 marks]

(ii) Compute $\mathbf{Cov}[X_1, X_2]$. [4 marks]

(iii) Suppose now that n is an unknown parameter and you observe the absolute difference between the labels of the first two balls, that is, $Z := |X_1 - X_2|$. Can you find an unbiased estimator of n based on Z? Justify your answer. [4 marks]

SECTION D

7 Algorithms

Consider alternative algorithms for sorting an array of n items.

(a) The *BST-sort* algorithm looks at each element of the array in turn, starting at position 0, and inserts it into a BST (pass 1). Having processed all elements, it repeatedly extracts the minimum from the BST, refilling the array from position 0 onwards (pass 2).

(i) Derive, with justification, the computational complexity of each of the two passes of BST-sort. [2 marks]

(ii) Describe a way of asymptotically speeding up pass 2 without changing the data structure, yielding *enhanced BST-sort*, and give the new computational complexity of pass 2 and of the overall algorithm. [2 marks]

(iii) Compare enhanced BST-sort against heapsort, mergesort and quicksort with respect to asymptotic worst-case time and space complexity, saying when (if ever) it would be preferable to any of them. [3 marks]

(b) The *enhanced 2-3-4-sort* algorithm is obtained by replacing the BST with a 2-3-4 tree in enhanced BST-sort.

(i) Perform pass 1 of enhanced 2-3-4 sort on the array {6,9,3,1,4,3,6,7,5,0,2}, redrawing the tree at each insertion. [*Hint:* Remember to split 4-nodes on the way down when inserting, and to put $\leq$ keys in the left child and $>$ in the right.] [5 marks]

(ii) How much space will enhanced 2-3-4-sort require to sort an array of n items, if each item is m bits long? Give exact upper and lower bounds in terms of n and m rather than an asymptotic estimate. [3 marks]

(iii) Repeat question (a)(i) for enhanced 2-3-4-sort. [2 marks]

(iv) Repeat question (a)(iii) for enhanced 2-3-4-sort. [3 marks]

8 Algorithms

An e-commerce website sells an average of m items per second, from a catalogue of n digital items. Among other details the website keeps track, for every sale, of a timestamp, the item's code and the sale price. It maintains three bestseller lists, refreshed every s seconds:

- the k best selling items ever, ranked by how many units were sold;
- the k best selling items of the past 30 days, ranked by how many units were sold;
- the k highest-revenue items of this calendar year, ranked by total revenue since 1st of January of this year.

(a) Each item will be held in a record. Describe all the data structures that must refer to these records to implement the required functionality. Describe all the fields that the record must have to implement the required functionality, and how each of these fields has to be updated and when. [5 marks]

(b) The obvious baseline solution is to re-sort the n items and to take the top k every time the bestseller lists must be produced. Assuming the number of items and the given average rates stay constant, what is its asymptotic worst-case time cost per unit time? [1 mark]

(c) Describe three alternative strategies, each better than the baseline, to implement the required functionality. Use the heaps or search trees of your choice, explaining precisely what you would store in each data structure to implement the required functionality. Describe, in each case, how to initialize the data structures, how to update the data structures after each sale, how to recompute the three bestseller lists every s seconds, together with the worst-case asymptotic time cost of each operation as a function of m, n, k, s (cost per unit time for the second and third operations). [6 marks]

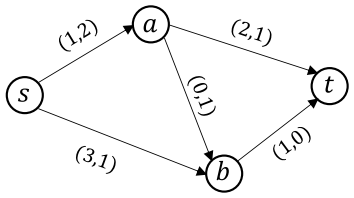
(d) Recommend the most appropriate strategy for $m = 10^4, n = 10^9, k = 10^2, s = 10^0$, with justification. [2 marks]

(e) Repeat for $n = 10^{14}$ and the other parameters unchanged. [3 marks]

(f) Repeat for $m = 10^{-4}, n = 10^2, k = 10^1, s = 10^5$, reasoning about the structural difference between this and the websites of cases (d) and (e). [3 marks]

9 Algorithms

Consider a directed graph in which each edge is labelled by a pair of non-negative costs, for example a distance and a travel time. A path between a pair of nodes is called 'Pareto efficient' (after the economist Vilfredo Pareto) if there is no other path for which *both* costs are lower.



(*a*) In the graph shown here, find all Pareto efficient paths from s to t, and state their costs. [1 mark]

(*b*) Show that, if $v_0 \to v_1 \to \cdots \to v_k$ is a Pareto efficient path from v_0 to v_k, then $v_0 \to \cdots \to v_{k-1}$ is a Pareto efficient path from v_0 to v_{k-1}. [3 marks]

(*c*) Let $v_0 \to \cdots v_k$ be a Pareto efficient path from v_0 to v_k, and let its costs be (c_a, c_b). Show that there is a Pareto efficient path from v_0 to v_k with costs (c_a, c_b) that has $\leq V - 1$ edges, where V is the number of vertices in the graph. [3 marks]

(*d*) We are given a start vertex s. Give an algorithm to compute *all* costs achievable by Pareto efficient paths from s to every other vertex. [6 marks]

(*e*) Prove that your algorithm is correct. [7 marks]

10 Algorithms

This question is concerned with connected undirected graphs in which each edge has a weight, and with spanning trees in such graphs.

(*a*) Explain what is meant by the *translation* strategy, and outline briefly the steps of a translation-based proof of correctness. [3 marks]

(*b*) Give an algorithm for finding a *maximum* spanning tree, that runs in $O(E + V \log V)$ time. Explain why your algorithm's running time is as required. [8 marks]

(*c*) Prove rigorously that your algorithm is correct. [9 marks]

[*Note:* You may refer to algorithms from lecture notes without quoting the code. You may use results from lecture notes without proof, but you must state them clearly.]

END OF PAPER