COMPUTER SCIENCE TRIPOS Part IB

Monday 3 June 2013 1.30 to 4.30

COMPUTER SCIENCE Paper 3

*Answer **five** questions.*

*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

1 Algorithms II

The context for this question is the search for a minimum spanning tree (MST) for a weighted connected graph.

(a) Give a clear definition of the following MST technical expressions, describing also the type of X and Y: "X respects Y", "z is a safe edge". [3 marks]

(b) For each of the following statements, say whether it is true or false and then support your argument with a correctness proof or a small counterexample as appropriate. [*You should give a specific graph, preferably small, if you are offering a counterexample.*]

(i) In a graph where all edge weights are positive, if a subset of edges connects all vertices and has minimum total weight, then it is a tree. [2 marks]

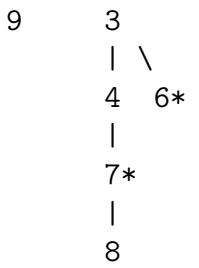
(ii) In a graph where edge weights may be positive or negative, if a subset of edges connects all vertices and has minimum total weight, then it is a tree. [5 marks]

(iii) Let T be a minimum spanning tree, C be a cut and e be the lightest edge crossing the cut. Assume $e \notin T$. Call f one of the edges of T that crosses the cut (one must exist because T spans all vertices). Then the set of edges $T \cup \{e\} \setminus \{f\}$ is also a tree. [10 marks]

2 Algorithms II

(*a*) Draw a clear sequence of commented snapshots showing what happens when applying the following sequence of operations to the two-tree Fibonacci heap pictured below, where asterisks denote marked nodes.

- insert key 5
- extract the minimum
- decrease key 8 by 2



[*You should draw one snapshot after each operation; and also, if it makes your explanation clearer, between 0 and 3 snapshots at key points during each operation.*]

[7 marks]

(*b*) Explain how "cascading cuts" work. [2 marks]

(*c*) Explain in clear detail why "cascading cuts" are necessary to achieve the Fibonacci heap's performance. What methods would be slower without cascading cuts, and by how much? [5 marks]

(*d*) "If, in a Fibonacci heap, we never call `decreaseKey()` or `delete()`, then at any time the degree (number of children) of any node is at most $\lceil \lg n \rceil$, where n is the total number of nodes in the heap at that time." True or false? Give a proof or a counterexample. [6 marks]

3 Programming in C and C++

(*a*) You have acquired a C program which declares an array `v`, populates it, and later writes it to a file in binary:

```
#define NITEMS 100
struct Elem { signed long val; char flags; } v[NITEMS];
...
fwrite(file, 1, sizeof(v), v);
```

When run on a legacy processor (which no longer exists) this produces a file containing 500 bytes, but when re-compiled and executed on three modern desktops using various compilers it produces files containing respectively 800, 1200 and 1600 bytes. On all implementations `char` is an 8-bit value.

(*i*) Explain what might be happening in the four versions in terms of compiler assumptions of alignment and size, in bits, of type `long`. Also give the values of `sizeof(struct Elem)`. [3 marks]

(*ii*) You now wish to read a file produced by the legacy processor into a version of the program running on your new desktop machine (one of the three above). Outline the changes, if any, you would need to make to the call to `fread`, mirroring the call to `fwrite` above, so that the resulting `v` may be successfully processed by the rest of the program. You may make any sensible assumptions about the legacy machine and your desktop machine provided you state them explicitly. Indicate how your program might be able to read both legacy- and new-format binary files. [10 marks]

(*b*) (*i*) Write a C++ class `T` which contains a `const` integer field `n`. `T` should also have constructor(s) which initialise `n` to an integer argument passed as a parameter or to zero if no argument is given; `T` should also have a destructor. The constructor(s) and destructor should print the value of the `n` field of the object being constructed or destructed. Indicate why, or why not, any of your fields or methods are qualified with `virtual`. [3 marks]

(*ii*) Explain how objects of class `T` are allocated and deallocated, for each of the three areas: heap, stack and static store, noting one case where appropriate use of `virtual` is essential. What, if any, overlap in programmer convenience is there between stack-allocated objects with destructors and `try-finally` in Java? [4 marks]

4 Compiler Construction

(*a*) Describe the costs and benefits of performing inline expansion of functions during compilation. [4 marks]

(*b*) Describe what is meant by *eliminating tail recursion*, when such an optimization can be applied and why it is a benefit. [4 marks]

(*c*) Consider the following ML-like program where the function `f` returns a function:

```
let val a = 99 in
let fun  f b = let g c = a + b + c in g end
let val f1 = f 17 in
let val f2 = f 33 in
let val v = (f1 a) + (f2 a) in
...
...
```

Describe carefully how this program fragment could be compiled. Explain how the expression

```
(f1 a) + (f2 a)
```

would be evaluated by your compiled code. [12 marks]

5 Compiler Construction

(*a*) When is it useful to eliminate left-recursion from a grammar and why? [2 marks]

(*b*) Write a recursive descent parser for the language generated by the following grammar.

```
E ::= E + F
    |  E - F
    |  F

F ::= NUM
    | ID
    | ( E )
```

[6 marks]

(*c*) This section deals with how object-oriented classes are typically implemented by a compiler when only simple inheritance (each class has exactly one parent) is supported.

(*i*) Describe in detail how objects are represented in memory and how this representation captures inheritance of attributes. [6 marks]

(*ii*) Discuss how virtual methods can be implemented. [6 marks]

6 Concepts in Programming Languages

(*a*) Give one difference and one similarity between the programming languages:

(*i*) Algol and Pascal [2 marks]

(*ii*) SIMULA and Smalltalk [2 marks]

(*b*) Define the parameter-passing mechanisms: by value, by reference, by value/result, and by name. [4 marks]

(*c*) What is the type of the expression

```
fn f => fn g => fn x => f( g( f x ) )
```

given by the SML interpreter? Explain how this is inferred. [6 marks]

(*d*) Give an example in the SML Modules language of a signature with an opaque type specification together with a structure that matches it. Explain your answer. [4 marks]

(*e*) List two language innovations of the programming language Scala. Justify your answer. [2 marks]

7 Further Java

A Java developer implements a class loader as follows:

```
public class NetworkClassLoader extends ClassLoader {

 private String server;
 private int port;

 NetworkClassLoader(String server, int port) { ... }

 @Override
 public Class<?> findClass(String name)
            throws ClassNotFoundException {
  //TODO: download class file and place contents in byte array b.
  return defineClass(name, b, 0, b.length);
 }
}
```

(*a*) Describe what a Java class loader is, when it is used, and why a developer might need to implement their own class loader. [3 marks]

(*b*) Write a Java program which accepts two arguments on the command line: a network port number and the full path to a Java class file in the file system. When executed, your program should wait indefinitely on the specified port for connections from clients. Whenever a client connects, your program should send the contents of the Java class file to the client. [8 marks]

(*c*) Complete the method `findClass` by downloading the class file from the program you wrote in part (*b*). Any error states should generate a `ClassNotFoundException`. [8 marks]

(*d*) Outline a security vulnerability which might arise when using your implementation of `NetworkClassLoader` from part (*c*) together with the server you wrote for part (*b*). [1 mark]

8 Prolog

(*a*) The propositional logic formula $A \wedge B$ can be represented by the Prolog term `and(lit(A),lit(B))`.

Describe a scheme based on this example for representing an arbitrary propositional logic formula in Prolog. Demonstrate your scheme by encoding the formula $\neg(\neg P \wedge (Q \vee \neg(R \wedge S)))$. [4 marks]

(*b*) A formula is in Conjunctive Normal Form (CNF) if it is expressed as a conjunction ($\wedge$-ing) of clauses, where each clause is a disjunction ($\vee$-ing) of literals.

Write a Prolog program for converting a propositional logic formula into CNF by implementing the following procedure:

(*i*) Push negations inwards until each applies only to a literal using *De Morgan's laws*: $\neg(A \vee B) \simeq \neg A \wedge \neg B$ and $\neg(A \wedge B) \simeq \neg A \vee \neg B$ [5 marks]

(*ii*) Remove double negations of literals: $\neg\neg A \simeq A$ [1 mark]

(*iii*) Distribute one disjunction from the formula over a conjunction or fail if no such disjunction exists: $A \vee (B \wedge C) \simeq (A \vee B) \wedge (A \vee C)$ [6 marks]

(*iv*) Repeatedly apply the distribution step until no more distribution can be done [4 marks]

Ensure that your predicates behave appropriately with backtracking, avoid over-use of cut, and are commented appropriately. Minor syntactic errors will not be penalised.

9 Software Engineering

Google have spent several years developing self-driving cars which rely on a range of sensors and stored data and are now claimed to have covered hundreds of thousands of miles with a good safety record.

A Google executive has now asked to meet the Transport Secretary, who has in turn asked your company for advice. What sort of safety case should the government demand from vendors and service providers if autonomous vehicles are to be allowed on Britain's roads? [20 marks]

END OF PAPER