

COMPUTER SCIENCE TRIPOS Part IB

Thursday 9 June 2005 1.30 to 4.30

Paper 6

*Answer **five** questions.*

*No more than **two** questions from any one section are to be answered.*

*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 Coversheets

Tags

SECTION A

1 Data Structures and Algorithms

A certain program has to maintain an array, `count`, of N counters which are all initialised to zero. The value of counter i can be incremented by one by the call: `increment(i)`, and this is the only way the program changes counter values. Two variables, `mincount` and `maxcount`, must always hold the smallest and largest of the counter values whenever the point of execution is not within the function `increment`. You may assume that `increment` is called about $1000N$ times when the program is run and that its argument is typically uniformly randomly distributed between 1 and N , but on some runs it cycles through the numbers 1 to N in order 1000 times.

- (a) Describe, in detail, an efficient data structure and algorithm to use when N is expected to be about 10. [5 marks]
- (b) Describe, in detail, an alternative data structure and algorithm to use when N is about a million. [10 marks]
- (c) Suppose your algorithm for (a) above were used when $N = 10^6$, estimate how much slower it would be compared with your algorithm for (b). [5 marks]

2 Computer Design

- (a) What is a *data cache* and why is it vital for high performance processors? [5 marks]
- (b) What is a *cache line* and how big is it likely to be? [3 marks]
- (c) What is a *page* and how big is it likely to be? [3 marks]
- (d) What is a *snoopy cache* and in what system is it likely to be used? [3 marks]
- (e) A computer system has the following memory parameters:

memory level	access time in clock cycles
first level cache	1
second level cache	3
main memory	100

If the probability of a hit in the first level cache is 80%, what hit rate do we need in the second level cache if the mean access time is to be two clock cycles? For this part, use the approximation that the cache line length is just one machine word. [6 marks]

3 Digital Communication I

- (a) Describe *on-off* flow control. In what circumstances is it appropriate? [4 marks]
- (b) Describe the operation of *window-based* flow control. [4 marks]
- (c) What happens if window-based flow control is used on a flow passing through a highly loaded resource (e.g. router) that is not participating in the flow control protocol? [4 marks]
- (d) How is this addressed in the Internet? [4 marks]
- (e) What are the advantages and disadvantages of having Internet routers participate in window-based flow control of every TCP connection? [4 marks]

4 Concurrent Systems and Applications

- (a) *Transactions* provide a means of grouping together a collection of separate operations to form a single logical operation. Explain the difference between implementations that enforce *strict isolation* and *non-strict isolation*. What are *dirty reads* and *cascading aborts*? [8 marks]
- (b) Optimistic Concurrency Control (OCC) is a mechanism used to enforce isolation of transactions.
- (i) Explain the OCC algorithm. [4 marks]
- (ii) Is deadlock possible? Are cascading aborts possible? [2 marks]
- (iii) Under what circumstances might this mechanism yield poor performance? [1 mark]
- (c) Explain how to use a *write-ahead log* to provide persistence by detailing what information must be written to the log and how it can be used to recover from a fail-stop catastrophe. What are *checkpoints* and why are they useful? [5 marks]

SECTION B

5 Compiler Construction

- (a) Give a program which gives different results according to whether dynamic or static binding is used. [3 marks]
- (b) Summarise two different methods of implementing functions which have free variables statically bound in an outer nested function, paying particular attention to (i) the lifetime of any storage used and (ii) any language restrictions which the implementation requires. Note also, for each case, what the effect is of assignment to such free variables, e.g. when calls to `g()` and `h()` are interleaved in

```
f(int x)
{  int a = x+7;
   int g() { a++; }
   int h(int y) { return a+y; }
   ...
}
```

[8 marks]

- (c) Give a program in which a function is called recursively without any value or function being defined recursively. [3 marks]
- (d) Give a program which produces three distinct results according to whether parameters are passed by value, by reference, or by value-result. [3 marks]
- (e) Explain the likely storage layouts for Java objects p and q respectively of classes P and Q defined by

```
class P          { int a,b; int f() { <body1> } }
class Q extends P { int c; int f() { <body2> }
                  int g() { <body3> } }
```

[3 marks]

6 Computer Graphics and Image Processing

- (a) In ray tracing, once we have determined where a ray strikes an object, the illumination at the intersection point can be calculated using the formula:

$$I = I_a k_a + \sum_i I_i k_d (\mathbf{L}_i \cdot \mathbf{N}) + \sum_i I_i k_s (\mathbf{R}_i \cdot \mathbf{V})^n.$$

Explain what real effect each of the three terms is trying to model, how accurately it models the real effect, and explain what each of the following symbols means, within the context of this formula:

$$I, I_a, i, I_i, k_a, k_d, k_s, \mathbf{L}_i, \mathbf{N}, \mathbf{R}_i, \mathbf{V}, n.$$

[12 marks]

- (b) Compare and contrast the ray tracing and z -buffer algorithms. [8 marks]

7 Comparative Programming Languages

- (a) Many computer scientists believe that languages with strong compile-time type checking are better than those that are typeless, dynamically typed, or are weakly type checked. Discuss the reasons for this view. [7 marks]
- (b) If strictly-checked data types are seen as good, discuss whether augmenting a language with many more primitive data types is better. Consider, in particular, the possibility of incorporating into a language such as Java many new numerical types such as packed decimal of various precisions, scaled arithmetic, and new types to hold values representing distance, mass and time. How would these additions affect the readability and reliability of programs? [7 marks]
- (c) Some languages allow different modes of calling of function arguments, such as call by value, call by reference and call by name. Discuss the advantages and disadvantages of incorporating the argument calling modes into the data types of functions. [6 marks]

8 Databases

(a) OLAP and OLTP.

(i) What is *on-line transaction processing* (OLTP)? [2 marks]

(ii) What is *on-line analytic processing* (OLAP)? [2 marks]

(iii) If you were designing a relational database system, how would your approach to schema design differ for OLTP and OLAP systems? [3 marks]

(iv) In OLAP, what is the meaning of the terms *drill down*, *roll up*, and *slice*? [3 marks]

(v) What is a *star schema*? [1 mark]

(b) Suppose we have the following relational schema,

```
Supplier(sid:integer, name:string, postcode:string)
Parts(pid:integer, name:string, description:string)
SuppliedBy(sid:integer, pid:integer, weight:integer)
```

where the underlined attributes represent the primary keys of the associated relation. The table *SuppliedBy* implements a relationship between suppliers and parts — indicating which parts are supplied by which supplier — using foreign keys pointing into the *Parts* and *Supplier* tables. The *weight* attribute is the parts weight in grams.

Write an SQL query that will return a list, without duplicates, of all postcodes associated with suppliers of parts less than one kilogram in weight. [5 marks]

(c) Define and explain the ACID properties of database transactions. [4 marks]

SECTION C

9 Logic and Proof

- (a) The propositional formula ϕ contains four propositional letters: P , Q , R and S . This formula evaluates to true in every case except when Q and R are false while S is true.

(i) What is the BDD for ϕ ? [2 marks]

(ii) What is the BDD for $\neg P \rightarrow \phi$? [3 marks]

(iii) What is the BDD for the formula $P \wedge S \rightarrow R$? [2 marks]

(iv) What is the BDD for the formula $(P \wedge S \rightarrow R) \wedge \phi$? [4 marks]

Use alphabetic ordering for all BDDs.

- (b) Use the DPLL procedure to determine whether or not the following set of clauses is satisfiable.

$\{P, Q, R\}$ $\{\neg P, Q, R\}$ $\{P, \neg Q, \neg R\}$ $\{\neg P, \neg Q, \neg R\}$ $\{\neg Q, R\}$ $\{\neg P, Q, \neg R\}$

[5 marks]

- (c) Prove the formula $\forall x[\neg P(x) \rightarrow Q(x)] \wedge \exists x \neg Q(x) \rightarrow \exists x P(x)$ using the tableau calculus (with Skolemization). [4 marks]

10 Foundations of Functional Programming

- (a) Give reduction rules associated with the standard combinators **S**, **K** and **I**. [4 marks]
- (b) Explain as simple a procedure as possible that converts lambda expressions into terms that use only these combinators. At this stage you are not expected to be concerned with efficiency, but you should make it clear why your combinator forms relate to the lambda expressions that you start with. [5 marks]
- (c) Illustrate the scheme you have given above by applying it to the lambda expression $\lambda x.\lambda y.((y\ x)\ y)$ and explain why in practical reduction to combinators at least additional symbols **B** and **C** are normally introduced. [5 marks]
- (d) Suppose that you have primitive functions available that can represent integers, booleans, arithmetic and an “if ... then” operation (and so on, as necessary). Show how to convert the following ML-like code first into raw lambda calculus and then into combinator form:

```
fun f n = if (iszero n) 1
          (double (f (subtract1from n)));
```

Note that the code has been presented in a way intended to show the functions used to perform arithmetic. You may assume the availability of a **Y** operator if that helps. [6 marks]

11 Semantics of Programming Languages

- (a) Discuss the operational semantics and type rules for subsumption and downcasting in a language with record subtyping. Identify clearly what is guaranteed by type-checking and run-time checks. [8 marks]
- (b) Explain, with examples, the subtype rule for function types $T_1 \rightarrow T_2$. [4 marks]
- (c) Explain, with examples, how a language with record subtyping, functions, and ML-style references can be used to express simple object idioms. [8 marks]

12 Complexity Theory

- (a) When defining space-complexity we use two-tape Turing Machines, with a read-only tape for input data and a read-write working tape. We count only the space used on the work tape. What difference would be made if we worked with standard one-tape Turing machines, loaded the input data onto the tape to start with and measured space in terms of total tape cells touched by the end of the computation? [4 marks]
- (b) Comment on the following: “The problem of finding a factor of a number N is NP, because if we have a factor P of N we can do a simple trial division and check it in time related to $\log(N)$. Thus finding factors of numbers of the form $2^p - 1$ (these are known as Mersenne Numbers) is a problem in the class NP”. [2 marks]
- (c) Define the class co-NP. State an example of a problem that lies in it. [2 marks]
- (d) What is a *witness function* for an NP problem? Why might some problem such as 3-SAT have many different witness functions associated with it? [2 marks]
- (e) Give and justify a relation between $NTIME(f(n))$ and $SPACE(f(n))$. [4 marks]
- (f) Matchings on bi-partite graphs can be found in polynomial time. The matching problem on tri-partite graphs is known to be NP-complete. Does this suggest that the corresponding problem with a graph whose nodes are partitioned into four sets (“quad-partite” matching) is liable to be exponential in complexity? Justify your answer. [4 marks]
- (g) Comment on the following proposition: “Determining which player can force a win from a given starting position in the game of Chess is an NP problem because given any sequence of moves it will be easy to verify that they are all legal moves and easy to see who wins at the end of them.” [2 marks]

END OF PAPER