UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

MASTER OF SCIENCE IN COMPUTER SCIENCE

DESIGN AND ANALYSIS OF PARALLEL ALGORITHMS

Thursday 20 April 2000

2:00 to 3:30

Examiners:

I. Watson (External)

R.N. Ibbett (Chairman)

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

1. (a) Define the logical behaviour of the computation usually known as 'prefix' or 'scan', both in general terms (i.e. independent of any particular operator) and with a concrete example.

(5 marks)

   (b) i. Explain how the computation of a prefix with an associative operator can be viewed as an example of 'divide-and-conquer'.

(5 marks)

   ii. Describe an implementation of the associative prefix of n items, initially distributed one per processor on an n processor binary hypercube. The implementation should run in $O(log\ n)$ steps. Explain the conditions under which the actual run-time will be $O(log\ n)$.

(10 marks)

   (c) Explain why an $O(log\ n)$ time algorithm as described above is not cost-optimal and outline a context in which it can nevertheless play a role in another asymptotically cost-optimal n processor computation.

(5 marks)

2. (a) Describe the problem solving strategy known as 'dynamic programming' and discuss its similarities with and differences from 'divide-and-conquer'.

(7 marks)

(b) Recall that the matrix parenthesization problem requires us to decide how to group a sequence of matrix multiplications in order to minimise the total number of operations performed.

i. Explain how matrix parenthesization can be solved by dynamic programming and include a sketch of the dependencies in a problem of size four (i.e. in which four matrices are being multiplied together).

(7 marks)

ii. Suggest a simple but efficient scheme for mapping the operations required by your algorithm, generalised to the case involving n matrices, onto an $n - 1$ processor machine in a manner which is independent of the underlying architecture. Explain which communication primitives would be invoked by the algorithm.

(6 marks)

iii. Derive the asymptotic run time of the resulting system as an expression parameterized by the architecture-specific costs of the communication primitives.

(5 marks)

3. (a) Explain, with the help of an example, why cost-optimality is a useful property for parallel algorithms.

(6 marks)

(b) Describe an algorithm for the n processor CREW PRAM which, given a sequence of n integers stored in an array, finds the length of the longest ascending portion of the sequence. For example, given the sequence 3,4,5,2,4,6,9,1 the algorithm would produce the value 4, corresponding to the sub-sequence 2,4,6,9.

You should aim for a $\Theta(log\ n)$ run-time algorithm.

You can assume that the usual reduction, prefix and segmented prefix operations are available and each run in $\Theta(log\ n)$ time provided that the operation being reduced or prefixed can itself be run in constant time.

(11 marks)

(c) Explain whether or not your algorithm is cost-optimal.

(3 marks)

(d) Discuss the relevance of Brent's theorem to the task of scaling your algorithm down to work on fewer processors.

(5 marks)