UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

MASTER OF SCIENCE IN COMPUTER SCIENCE

DESIGN AND ANALYSIS OF PARALLEL ALGORITHMS

Wednesday 18 April 2001

9:30 to 11:00

Examiners:

I. Watson (External)

R.N. Ibbett (Chairman)

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

1. (a) Explain the problem solving strategy known as *divide-and-conquer*, discussing the issues which arise when it is applied to parallel algorithm design.

[*7 marks*]

(b) i. Describe a divide-and-conquer algorithm for the n processor hypercube architecture which solves the data re-arrangement problem of reversing a sequence of n items. The items are originally stored one per processor, with processor i storing the i^{th} item, for $0 \leq i \leq n-1$. The result is that processor i should store what was originally the $n-1-i^{th}$ item. You should aim for a $\Theta(log\ n)$ time algorithm. [*10 marks*]

ii. Calculate and explain the asymptotic run-time of your algorithm. [*3 marks*]

iii. Describe how your algorithm could be adapted to reverse sequences of n items on p processors for $n > p$, with adjacent blocks of items from the original and reversed sequences stored on adjacently indexed processors.

[*5 marks*]

2. (a) Explain the structure of the *odd-even transposition sort* algorithm for the case in which p processors in a linear array each store $\frac{n}{p}$ items. [*8 marks*]

(b) Sketch a diagram which illustrates the execution of the algorithm when run with four processors on the sixteen item input 3, 4, 7, 2, 5, 8, 9, 9, 4, 4, 1, 2, 1, 3, 7, 8. You should show the data stored by each processor after each step of the algorithm. Each step should include a very brief (a sentence at most) description of what is happening. [*5 marks*]

(c) Compute the asymptotic isoefficiency of the algorithm, clearly stating any assumptions you make. [*6 marks*]

(d) Consider implementing the same algorithm on the p processor PRAM. Discuss any changes that would be required, explain the weakest PRAM variant which could sensibly be used and describe any effect this change of architecture would have on the *asymptotic* analysis of the algorithm. [*6 marks*]

3. (a) Define and explain the concept of *cost-optimality* for PRAM algorithms. [*3 marks*]

(b) Describe an algorithm for the n processor CREW PRAM which, given an array of n integers, re-arranges the array so that those integers which are smaller than the average (of all the integers) precede those which are equal to or greater than the average. For example, for the input 4, 3, 6, 8, 2, 3, 1, 5, the result should be 3, 2, 3, 1, 4, 6, 8, 5 (because the average of the eight numbers is 4).

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 runs in $\Theta(log\ n)$ time provided that the operation being reduced or prefixed can itself be run in constant time. [*11 marks*]

(c) Explain briefly why an n processor algorithm which achieves the target run-time above cannot be asymptotically cost-optimal. [*3 marks*]

(d) Explain whether or not your algorithm from (b) can be adjusted to become cost-optimal by reducing the number of processors used without reducing the asymptotic run-time. [*8 marks*]