Homework #7: Parallel Algorithms
Due Date: Tuesday, April 14, 1998 (5:00pm)

Collaboration Policy

You may collaborate with others on this entire homework. However, please re-read the policy guidelines from Handout #1.

Reading

You should read the introduction of Ch. 30 of CLR, along with sections 30.1, 30.2, and 30.4.

Practice

Recall, these exercises are purely for your own practice. You need only turn in the official problems.

- Do CLR 30.1-8.
- Do CLR 30.1-4.
- Do CLR 30.2-3.
- Do CLR 30.2-4.
- Do CLR 30.3-2.

Problems

1. Do CLR 30-2. (35 points)
   We wish to find the maximum of n numbers on a CRCW PRAM with $p = n$ processors.

   a. Show that the problem of finding the maximum of $m \leq p/2$ numbers can be reduced to the problem of finding the maximum of at most m^2/p numbers in $O(1)$ time on a p-processor CRCW PRAM. (*Hint:* Keep in mind that we know how to find the maximum of k numbers in constant time with k^2 processors. How can you assign groups of processors to groups of input numbers to utilize this fact?)
   b. If we start with $m = \lfloor p/2 \rfloor$ numbers, how many numbers remain after k iterations of the algorithm in part (a).
   c. Show that the problem of finding the maximum of n numbers can be solved in $O(\lg \lg n)$ time on a CRCW PRAM with $p = n$ processors.

2. (35 points)
Given an n-element list, where some elements of the list are marked as being blue, your goal is to create a list consisting of exactly the blue elements. Give an algorithm on an EREW PRAM, using $P(n) = n$ processors, and $T(n) = O(\lg n)$ running time. In your write-up, you should *reasonably justify* the correctness of your algorithm, the running time, and also that it truly obeys EREW. (partial credit will be given for algorithms which require a more powerful model of concurrency.)

3. CLR 30.1-5 and CLR 30.1-6 (30 points)
Show how to modify the Euler tour technique, to give EREW algorithms which compute the following information for an arbitrary n-node binary tree, using $O(\lg n)$ time and $O(n)$ processors. (you should give a separate algorithm for each of the following)

   a. (6 points) The inorder numbering of all nodes
   b. (6 points) The preorder numbering of all nodes
   c. (6 points) The postorder numbering of all nodes
   d. (12 points) The size of the subtree rooted at each node
   (*Hint:* Take the difference of two values in a running sum along an Euler tour. Alternatively, since you already know how to compute the answers to parts a, b, and c, can you use this information?)

Extra Credit

(CLR 30.2-6)
Prove that for any constant $\epsilon > 0$, there is an $O(1)$-time CRCW algorithm using $O(n^{1+\epsilon})$ processors to find the maximum element of an n-element array. (*Hint:* Think about your solution to Problem 1 above. Can you modify the analysis?)