Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Harald Räcke
Chris Pinkau

Winter Semester 2013/14
Problem Set 6
November 26th, 2013

Parallel Algorithms

Due date: December 3rd, 2013 **before** *class!*

Problem 1 (10 Points)

Given a set of positive integers $x_1, \dots, x_n$ stored in the first n cells of the global memory of an arbitrary CRCW PRAM, the *element distinctness problem* is to determine whether there exist $i \neq j$ such that $x_i = x_j$. Show how to solve this problem in $\mathcal{O}(1)$ time, using n processors.
Hint: Exploit the fact that the x_i's are positive integers.

Problem 2 (10 Points)

Consider the computation of x^n, where x is an input data stored in a location of the global memory.

1. Show that x^n can be computed in one step on the ideal PRAM model.
2. Show that $\Omega(\log n)$ steps are required by the standard arithmetic CREW PRAM. Assume that each step consists of a read instruction, or a write instruction, or an arithmetic operation from $\{+, -, \times, \div\}$.
Hint: Show that any function computed at the kth step is of degree $\leq 2^k$.

Problem 3 (10 Points)

The purpose of this exercise is to show that the Boolean OR function of n variables can be computed by an n-processor CREW PRAM in $\leq \log_{2.618} n + \mathcal{O}(1)$ steps, which is less than $\log_2 n$.
Let the input bits $x_1, \dots, x_n$ be stored in $M(1), \dots, M(n)$ of the global memory, and let $n = F_{2T+1}$, where F_i is the ith Fibonacci number with $F_0 = 0, F_1 = 1$, and $F_{m+2} = F_{m+1} + F_m$ for $m \geq 0$. Each processor P_i uses two variables y_i and t, initially set to 0. The algorithm executed by P_i is the following:

- **if** $i + F_{2t} \leq n$ **then** $y_i \leftarrow y_i + M(i + F_{2t})$
- **if** $(i > F_{2t+1} \text{and} y_i = 1)$ **then** $M(i - F_{2t+1}) \leftarrow 1$

1. Show that, just before step t, we have $y_i = x_i + x_{i+1} + \cdots + x_{i+F_{2t}-1}$ and $M(i) = x_i + x_{i+1} + \cdots + x_{i+F_{2t+1}-1}$ for $1 \leq i \leq n$.
2. Deduce that the algorithm uses at most $\log_{2.618} n + \mathcal{O}(1)$ steps.