mpi INFORMATIK

Universität
des
Saarlandes
FR 6.2 Informatik

Dr. Ulrich Meyer **WS 2003/4**

Exercises for Parallel and Distributed Algorithms

1. Assignment Due 05. Nov. 2003

Exercise 1 (2 + 2 *Points*)
Show that any algorithm for the "telephone sum" problem of n numbers requires

a) $n-1$ phone calls

b) $\Omega(\log n)$ parallel phases/time.

Recall that each person can only talk with one other person at a time.

Exercise 2 (2 *Points*)
Show that the following problem can be solved in constant time on an EREW PRAM with n PUs: Given a sorted boolean sequence (e.g. $(0,0,0,1,1,1,1,1)$) of length n. Determine the number of zeros in the sequence.

Exercise 3 ($2+2^*$ *Points*)

a) Show that an EREW PRAM with n PUs can compute the numbers $\lfloor\sqrt{n}\rfloor$ and $\lceil\sqrt{n}\rceil$ in constant time even if the single PUs are only capable to perform the numeric operations $+$, $-$, $*$, and div (integer division).

b) * What about computing $\lfloor n^{5/17}\rfloor$?

We assume that the number n is initially known to each PU.

Exercise 4 (3 *Points*)
(Segmented Prefix-Sums) Imagine you want to solve several prefix-sum problems whose input elements are from the same universe U, use the same associative operator $\circ$ but are otherwise independent. The input for all problems is given in an array $A[1..n]$, starting with the input of the first problem, then for the second one, and so on. A second array $B[1..n]$ denotes the starting points of the subproblems: for $i=1,\ldots,n$: $B[i]=1 \Leftrightarrow A[i]$ is the first input element of this subproblem. Show that the parallel solution of all subproblems can be reduced to a single prefix-sum problem of size n. More specifically, compute $C[1..n]$ where $C[i]=A[j_i]\circ A[j_i+1]\circ\cdots\circ A[i]$ for $i=1,\ldots,n$ where $j_i=\max\{j\le i : B[j]=1\}$ (we always have $B[1]=1$).
Hint: Define an appropriate associative operator on pairs (x,b) where $x\in U$ and $b\in\{0,1\}$.

Exercise 5 (3 *Points*)
(Segmented Broadcast) Show that the following problem can be solved in $O(\log n)$ time on a PRAM with n PUs: Given an array $A[1..n]$ in which some cells can be empty (formally, being empty can be specified using a second boolean array $B[1..n]$ similarly to exercise 4), store in each empty cell $A[i]$ of A the element of the nearest nonempty cell left to $A[i]$ (leave the cell empty in case there are no nonempty cells to the left).