ICS 643: Advanced Parallel Algorithms
Homework 1

Due: Wednesday, October 12, 2016, at the start of lecture

Instructions: You may discuss the problems with other students in the class, but you must write up the solutions on your own and give credit to the students with whom you discussed each problem.

1 Prefix minima (15 pts)

Given an array $A = [a_0, a_1, \ldots, a_{n-1}]$, design a EREW PRAM parallel algorithm that computes an array $B = [b_0, b_1, \ldots, b_{n-1}]$ such that each element $b_i = \min(a_0, a_1, \ldots, a_i)$, i.e. is the minimum among the first $i$ elements of $A$. Your algorithm should run in $O(\log n)$ time and $O(n)$ work. You should present your algorithm as pseudocode. Prove the correctness, time and work complexities of your algorithm.

2 Partition (35 pts)

Given an array $A$ of $n$ numbers and a bit vector $B$ of $n$ bits, the problem of partitioning asks to rearrange elements of $A$, such that all elements $a_i \in A$ whose corresponding bit value $B[i]$ is equal to 0 come before all elements $a_j \in A$ whose corresponding bit value $B[j]$ is equal to 1. For example, $A = [4, 3, 1, 5, 10]$ and $B = [1, 0, 1, 0, 0]$ should result in $A' = [3, 5, 10, 4, 1]$. Design a EREW PRAM parallel algorithm that solves the partitioning problem. Write out the pseudocode for your algorithm. Your solution should exhibit the same time and work complexity as the prefix sums algorithm.

3 Parallel Merging (50 pts)

Let $A$ and $B$ be two sorted arrays, each containing $n$ numbers.

(a) (5 pts) Write down pseudocode for a sequential algorithm to merge the two arrays into another array $C$ of size $2n$. Your algorithm should run in time $T_1(n) = O(n)$ time.
(b) (15 pts) Give an $O(\log n)$-time sequential algorithm to find the median of all $2n$ elements in arrays $A$ and $B$. Write out the pseudocode for your algorithm and prove its running time.

(c) (15 pts) Design a parallel algorithm for 2 processors that merges the arrays $A$ and $B$ into another array $C$ and runs in time $T_2(n) = O\left(\frac{T_1(n)}{2} + \log n\right)$. Analyze the running time of your algorithm. (*Hint: use the median computation from part (b) to make sure that each processor compares at most $n$ elements.*)

(d) (15 pts) Generalize your solution in part (c) to a $P$-processor parallel algorithm. Analyze the running time of your new algorithm. What is the maximum value of $P$ for which your algorithm remains work-optimal?