ICS 691: Parallel Algorithms
Homework 1

Due: October 2, 2014

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

1 Recurrences (10 pts)

Solve the following recurrence for $T(N, N)$:

$$T(n, N) = \begin{cases} T(\frac{n}{2}, N) + \Theta(1) & \text{if } n \geq \frac{N}{\log N} \\ \Theta(1) & \text{otherwise} \end{cases}$$

Your solution should be better than $T(N, N) = O(\log N)$.

2 Prefix minima (20 pts)

Given an array $A = [a_1, a_2, \ldots, a_n]$, design a parallel algorithm that computes an array $B = [b_1, b_2, \ldots, b_n]$ such that each element $b_i = \min(a_1, a_2, \ldots, a_i)$, 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. Write down the pseudocode and prove the correctness, time and work complexities of your algorithm.

3 In-place Partition (30 pts)

Given an array A of elements and a bit vector B, the problem of *in-place partitioning* of A asks to rearrange elements of A, such that all elements $a_i \in A$ with corresponding bit vector values $B[i] = 0$ are placed in A before all elements $a_j \in A$ with corresponding bit vector values $B[j] = 1$.

(a) (15 pts) Design a parallel algorithm that solves the in-place partitioning problem. Write out the pseudocode for your algorithm. Your solution should exhibit the same time and work complexity as the prefix sums algorithm.

(b) (15 pts) Design a parallel algorithm that solves the segmented version of the in-place partitioning problem. That is, there is another bit vector C, that defines boundaries of segments within A and in-place partitioning must be performed within each segment. Write out the pseudocode for your algorithm. Your solution should exhibit the same time and work complexity as the segmented prefix sums algorithm.

4 Segmented prefix sums (40 pts)

(a) (15 pts) Write out the pseudo code for segmented prefix sums WITHOUT the reduction to the associative operator $\bullet$ described in class.

(b) (15 pts) Write out the pseudo code for segmented prefix sums using the reduction to the associative operator $\bullet$ described in class.

(c) (10 pts) Compare the two algorithms. Discuss in what ways they are similar and in what ways they differ. Which one was simpler to design?