

# Homework 2

CSc 8530 Parallel Algorithms  
Spring 2019

Due: 11:59pm, March 9, 2019

1. **(15 pts) Write pseudocode** for a non-recursive prefix-sums algorithm that is similar to the one studied in class but that does not use the auxiliary variables  $B$  and  $C$ . The input array  $A$  should hold the prefix sums when the algorithm terminates.
2. **(15 pts)** We are given an array of colors  $A = [a_1, a_2, \dots, a_n]$  drawn from  $k$  colors  $\{c_1, c_2, \dots, c_k\}$ , where  $k$  is a constant. We wish to compute  $k$  indices  $i_1, i_2, \dots, i_k$ , for each element  $a_i$ , such that  $i_j$  is the index of the closest element to the *right* of  $a_i$  whose color is  $c_j$ . If no such element exists, then set  $i_j = 0$ . **Write pseudocode** for solving this problem in  $O(\log(n))$  using a total of  $O(n)$  operations.
3. **(15 pts)** Suppose that we have an algorithm  $A$  to solve a given problem  $P$  of size  $n$  in  $O(\log(n))$  time on the PRAM model using  $O(n \log(n))$  operations. On the other hand, an algorithm  $B$  exists that reduces the size of  $P$  by a constant fraction in  $O\left(\frac{\log(n)}{\log \log(n)}\right)$  time using  $O(n)$  operations without altering the solution. Derive an  $O(\log(n))$  time algorithm to solve  $P$  using  $O(n)$  operations.