ICS 491: Parallel Algorithms
Homework 1

Due: February 20, 2014

1 Prefix minima

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 correctness, time and work complexities of your algorithm.

2 Bubble sort network

Use the 0-1 principle to prove the correctness of the Bubble sort network.

3 ShearSort: sorting on square matrices

Consider the following algorithm for sorting items in a \( n \times n \) matrix:

- Repeat the following \( \log n \) times:
  1. Sort rows in alternating order: sort each odd-numbered row in increasing order and each even-numbered row in decreasing order.
  2. Sort each column in increasing order

(a) Assuming that the rows and columns are sorted using a sorting network, prove that the above algorithm sorts the matrix in a row-major order. (Hint: Use the 0-1 principle and consider how many rows remain to be sorted after each round)

(b) Analyze the running time of the algorithm.