Principles of Distributed Computing
Exercise 8

1 Sorting Networks

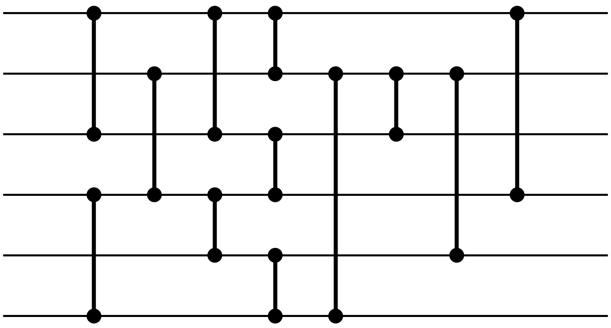

Figure 1: A Sorting Network?

For each of the following questions, prove or disprove the given claim.

a) The network of width 6 and 12 comparators in Figure 1 above is a sorting network, that is, it sorts each input sequence of numbers correctly.

b) Given any correct sorting network, adding another comparator at the end does **not** destroy the sorting property.

c) Given any correct sorting network, adding another comparator at the front does **not** destroy the sorting property.

d) Every correct sorting network needs to have at least one comparator between each two consecutive horizontal lines.

e) A network which contains all $\binom{n}{2}$ comparators between any two of the n horizontal lines, in whatever order they are placed, is a correct sorting network.

f) Given any correct sorting network, adding another comparator anywhere does not destroy the sorting property.

g) Given any correct sorting network, inverting it (i.e., feeding the input into the output wires and traversing the network "from right to left") results in another correct sorting network.

2 Alternative Proof for the 0-1 Sorting Lemma

Suppose that you are given an oblivious comparison-exchange network that transforms the input sequence $a = (a_1, a_2, ..., a_n)$ into the output sequence $b = (b_1, b_2, \cdots, b_n)$. In addition, suppose you are given a monotonically increasing function $f : \mathbb{N} \mapsto \mathbb{N}$. Note that a function f is called monotonically increasing if for any fixed $x, y \in \mathbb{N}$,

$$x \leq y \Rightarrow f(x) \leq f(y).$$

a) Prove that a single comparator with inputs $f(x), f(y) \in \mathbb{N}$ produces the outputs $f(\max(x, y))$ and $f(\min(x, y))$.

b) Prove that the oblivious comparison-exchange network transforms the input sequence $F(a) = (f(a_1), f(a_2), \cdots, f(a_n))$ into the output sequence $F(b) = (f(b_1), f(b_2), \cdots, f(b_n))$.

c) Use the previous question to prove the 0-1 Sorting Lemma: *If an oblivious comparison-exchange algorithm sorts all inputs of 0's and 1's, then it sorts arbitrary inputs.*

3 Recursive Sorting Networks

Suppose that you are given a black-box sorting network of width $n-1$ and that you must adapt it in order to build a sorting network of width n. You are only allowed to add comparators *after* the sorting network (see Figure 2). You can assume that comparators output the maximum value on the bottom wire (i.e., sorting in ascending order starting from the top wire).

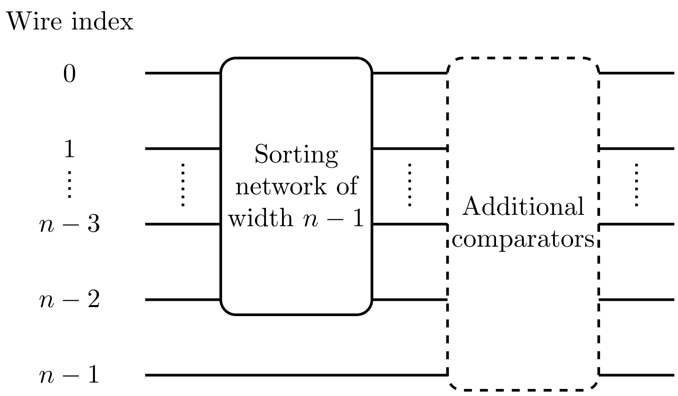

Figure 2: Recursive sorting network.

a) Find a solution with $n-1$ comparators. Is your solution unique?

b) Show that there is no solution with strictly less than $n-1$ comparators (hence proving that $n-1$ is optimal for building recursive sorting networks in this manner).

c) Suppose that you start with a single comparator (i.e., a sorting network of width 2) and that you recursively build sorting networks up to width n by adding comparators as above, but only using width 1 comparators (i.e., linking adjacent wires). What well-known sorting algorithm does the resulting network implement?

d) Same three questions, but now you can only add comparators *before* the black-box sorting network.