University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
M. Fuchs, Z. Parsaeian

Theory of Distributed Systems
Sample Solution Exercise Sheet 11

Exercise 1: Sorting Network Short Questions

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

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

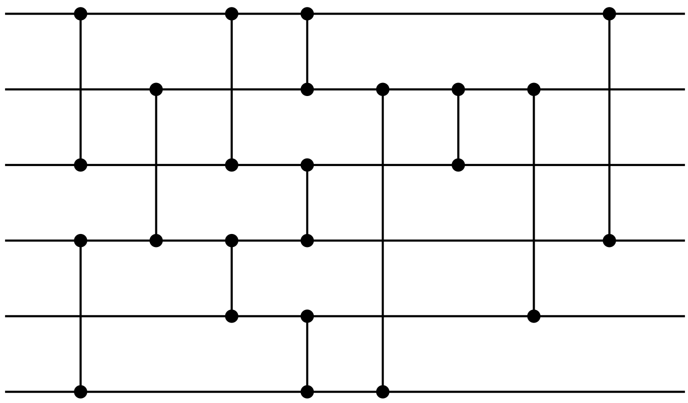


Figure 1: A sorting network of width 6 and 12 comparators.

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

c) 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.

d) There is a network which contains $\binom{n}{2}$ comparators that is a correct sorting a network with n wires.

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

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

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

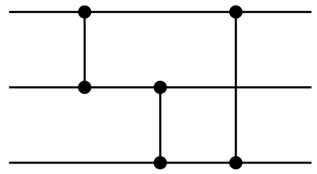
h) Given any correct sorting network, inverting it (i.e., feeding the input into the output wires and traversing the network "from right to left") results in another correct sorting network.

Sample Solution

a) Wrong, consider the input $I = (0, 0, 1, 0, 0, 0)$ (from top to bottom). The output is $(0, 0, 0, 1, 0, 0)$ and hence not sorted.

b) Correct. Assume otherwise: There is a correct sorting network S such that there is no comparator between horizontal lines i and $i + 1$. Consider the input sequence I consisting of 0's, then a 1 at the ith spot, a 0 at spot i+1, and the rest 1s. Now every comparator will leave I intact, because I is already sorted except for the horizontal lines i and $i + 1$. But since there is no horizontal line, the output to S is the same as the input, $O = I$, which is not sorted. Thus, no such S exists.

c) Wrong. Consider the following network and observe that $I = (1, 1, 0)$ is not sorted correctly.

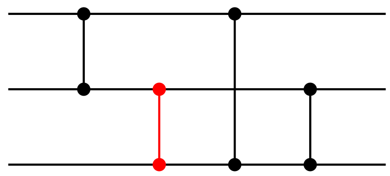


d) Correct. For example we add the comparators between each two wires, such that with $n - 1$ comparators, we got the highest value to its correct position. Now we recurse on the remaining $n - 1$ wires. The number of comparators needed is $(n - 1) + (n - 2) + ... + 2 + 1 = \frac{n \cdot (n-1)}{2} = \binom{n}{2}$.

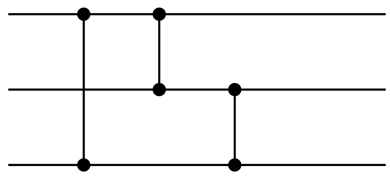
e) Correct. A comparator leaves a sorted sequence intact.

f) Correct. A proper sorting network needs to be able to sort any input sequence, and a comparator added to the front merely changes the input for the sorting network.

g) Wrong. There is a counterexample when adding the red comparator and observing $I = (1, 1, 0)$:



h) Wrong. There is a counterexample with $I = (1, 0, 0)$:



Exercise 2: Recursive Sorting Network

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).

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).

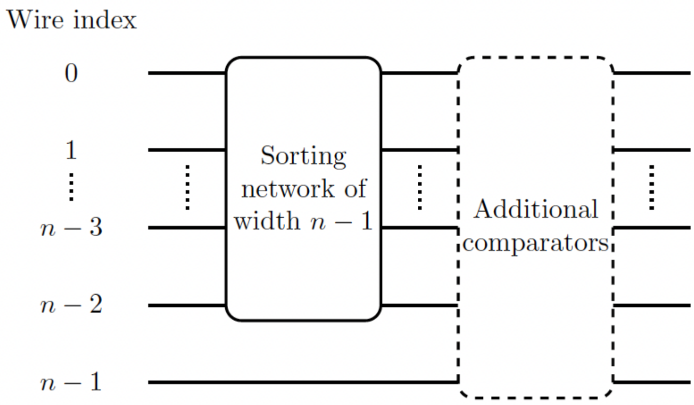


Figure 2: Recursive Sorting Network

c) What is the depth of your construction (ignoring the black-box network)? Show that, whenever n is a power of 2, any such network must have depth $\geq \log_2 n$.

d) Assuming $n = 2^k$, modify your construction (of task a)) to use at most $n-1$ comparators and at most $\log_2 n$ depth.

e) Does your previous constructions also work if you can only add comparators *before* the black-box sorting network?

Sample Solution

a) The solutions are not unique, see the following two possibilities:

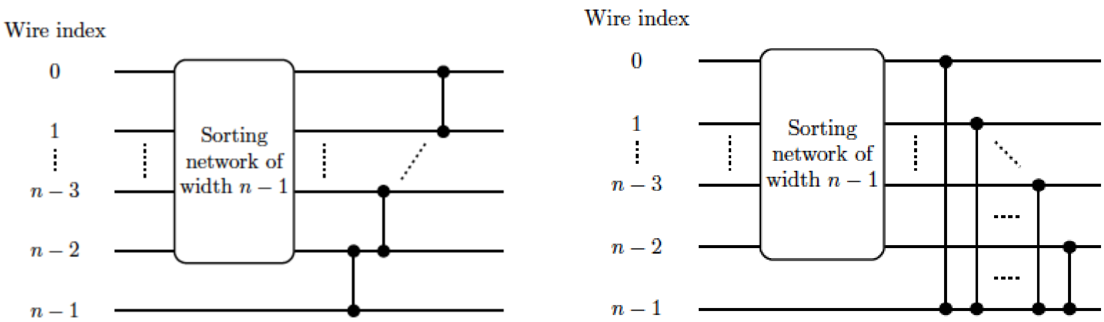


Figure 3: possible solutions

b) By contradiction, assume that there exists a solution with $k < n-1$ comparators. This means, that one of the n wires is not connected to any of the others (as one needs at least $n-1$ comparators to connect each wire with a comparator). Let's call this wire X. If X is the new wire, it is clear that we cannot sort sequences where the input on X is not the maximum of all values. If X is one of the 'old' wires, we consider an input sequence where the value of our new wire needs to be outputted at wire X (to meet the sorting property). Thus, we need at least one comparator at X to achieve that. Contradiction.

c) The depth of both solutions in a) is $n-1$. We will now give an argument why depth of $\log_2 n$ is needed. Initial, there is only 1 interesting wire (we say a wire is interesting when it could contain the new input, i.e., it is wire $n-1$ or there is a path over comparators from wire $n-1$ to it), and this is the new one. As all the other ones are already in sorted manner. If there now is a comperator connecting the new wire to some other wire, there are at most 2 interesting wires (cause one of these 2 connected wires may has the new value). If there are (parallel) comparators between these 2 wires and some arbirtrary other wires, there are at most 4 interesting wires... Thus, to make sure that all n wires are interesting (and thus could see the new input) we need depth of at least $\log_2 n$.

d) This is possible, the construction is as follows. In the first step you put a comparator between wire $n-1$ and $n/2-1$. Now you recurse on the (equally sized) blocks of wires $0, ..., n/2-1$ and $n/2, ..., n-1$.
To see why this works we use induction. It's easy to see that the construction works with 2 wires.

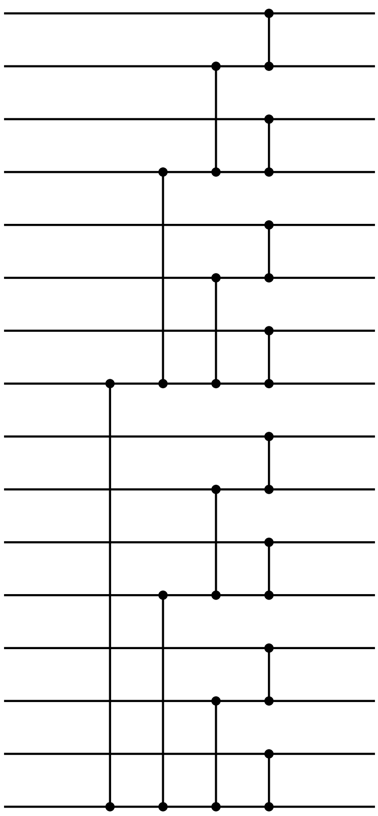


Figure 4: Example with 16 wires

For the induction step assume that our scheme works for $n/2$ wires that are presorted except of the last one. Having n wires, the new value from wire $n-1$ is either in the output of wires from $0, ..., n/2-1$ or from $n/2, ..., n-1$. Assume we are in the first case, then our first comparator is guaranteed to switch the values of wires $n-1$ and $n/2-1$. Now the first block consists of $n/2$ wires sorted except of the last one and the second block also consists of $n/2$ wires sorted except of the last one. Thus, these blocks are sorted correctly by our induction hypothesis. If we are in the case that our new value should be in the block $n/2, ..., n-1$, the first comparator has no effect, and the remaining block $n/2, ..., n-1$ is again correctly sorted by the induction hypothesis.
It remains to argue that this construction uses exactly $n-1$ comparators. We again use induction. For 2 wires the statement is obvious. For n wires we need 2 times the number of comparators for $n/2$ wires plus the additional one (that connects both blocks). Overall, this sums up to $2 \cdot (n/2-1)+1 = n-2+1 = n-1$ many wires.

e) The construction does not work as it is. The previous constructions will work when you reverse the order of the comparators (make the rightmost one the leftmost one and so on). The idea here is that we want the largest value on wire $n-1$ as the remaining will be anyway sorted correctly afterwards. That is indeed what our reverse construction achieves. This can be shown by induction, similar to what we have done in c).