More Sorting Networks

CS255

Chris Pollett

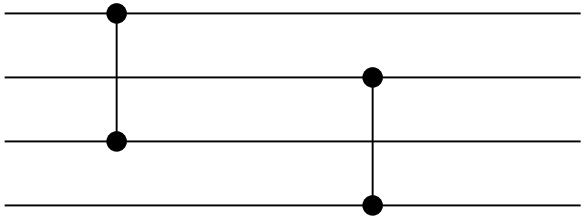
Feb. 13, 2006.

Outline

- Bitonic Sorting
- Merging Networks
- A Sorting Network

A Half-Cleaner

- Our bitonic sorter is composed of several stages, each of which is called a **half-cleaner**.
- Each half-cleaner is a comparison network of depth 1 in which input line i is compared with line $i+n/2$ for $i=1,2,\ldots, n/2$ (assume n is a power of 2).

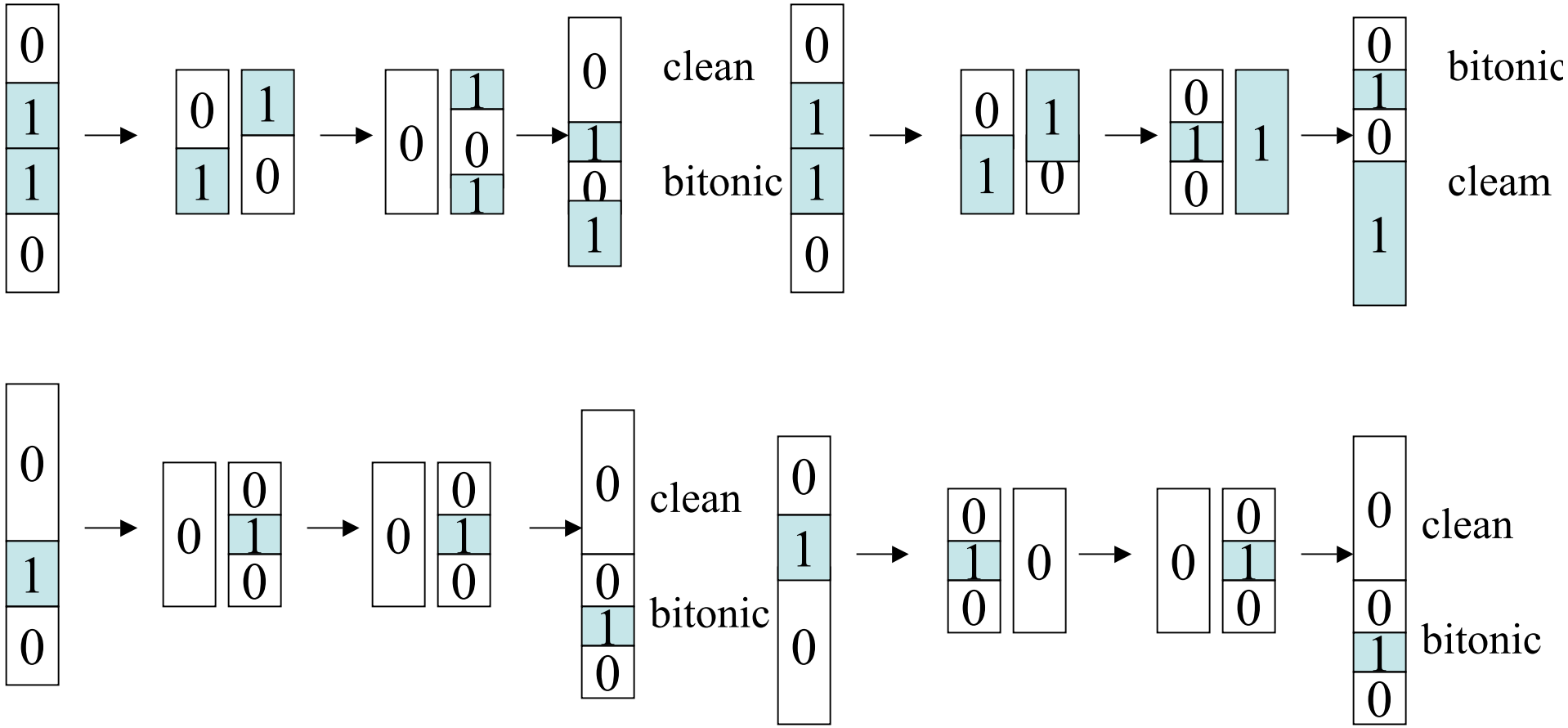


Lemma 27.3

Lemma If the input to a half-cleaner is a bitonic sequence of 0's and 1's then the output satisfies the following properties: both the top half and the bottom half are bitonic and every element in the top half is at least as small as every element of the bottom half, and at least one half is clean (all 1's or all 0's).

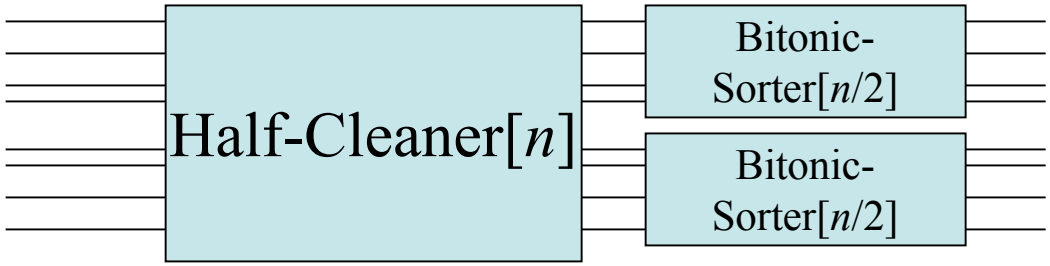
Proof Half-Cleaner[n] compares i with i+n/2 for each i=1,2, …, n/2. Without loss of generality, suppose the input is of the form $0^i1^j0^k$. The n/2 position can occur in the first block of 0's, in the block of 1's, or in the second block of 0's. The second case we further split in two. The next slide shows what to do in each case.

Cases for the Lemma



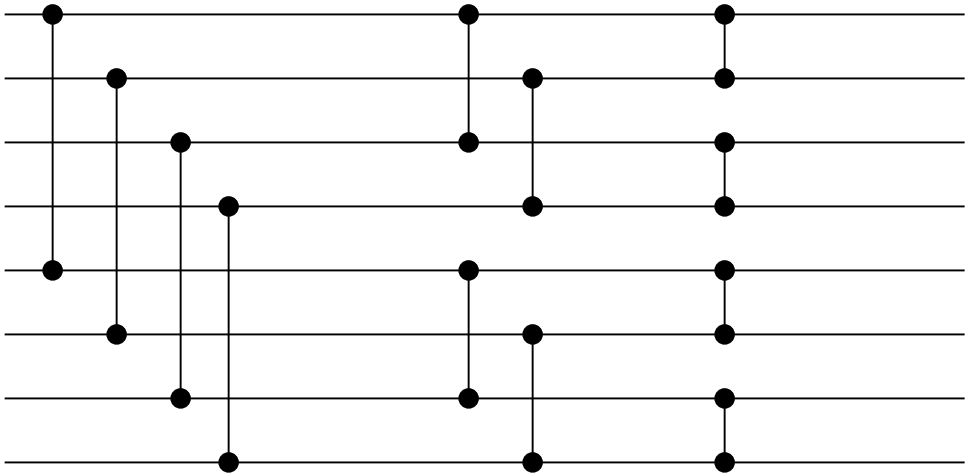
Bitonic Sorter

- By recursively combining half-cleaners, we can build a **bitonic sorter**.



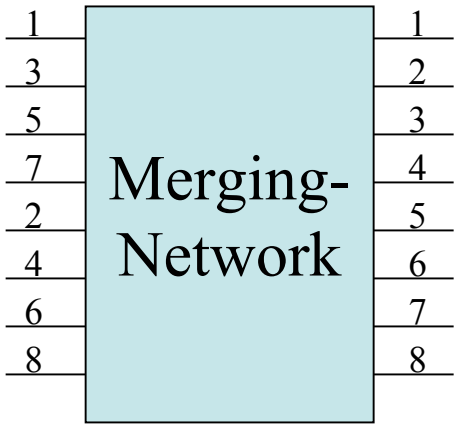
- By the previous lemma, Half-Cleaner[n] will produce two bitonic sequences such that the top-half is less than or equal to the values in the bottom half and one of these halves is clean. So this gives us both two smaller instances on which Bitonic-Sorter[n/2] can complete the sort. It also guarantees we can sort these two sides independently and still produce one sorted list.
- The depth of this network will be D(n) = 0 if n=1, and D(n/2)+1 if n=2^k and k>=1. So D(n) = lg n.

Example Bitonic-Sorter[8]



Merging Networks

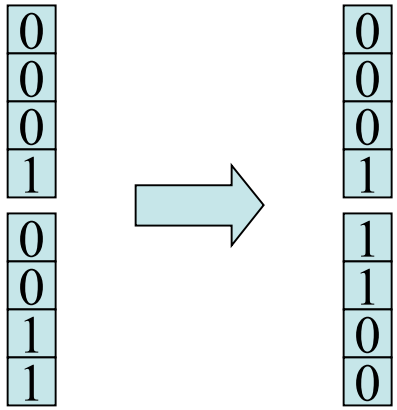
- A sorter will be constructed from merging networks.
- These are networks which can take two sorted inputs and merge the result into one sorted output.



- One can prove the correctness of merging networks also using only zero-one inputs.

More on Merging Networks

- Given two sorted zero-one input sequences, consider the effect of reversing the second sequence:

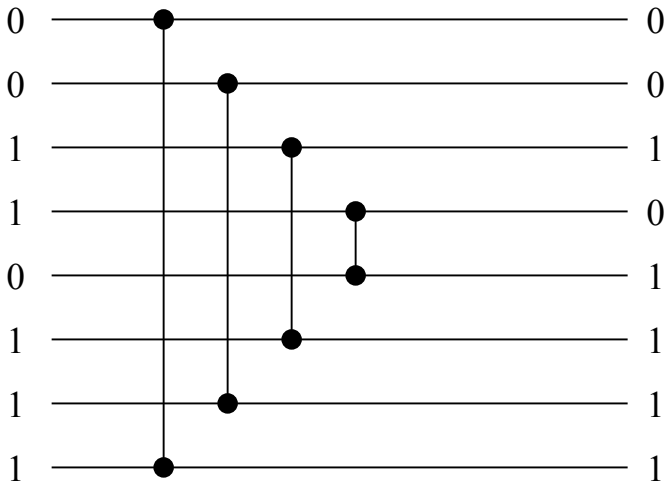


- It is bitonic. Then we can use our bitonic sorter on the result.

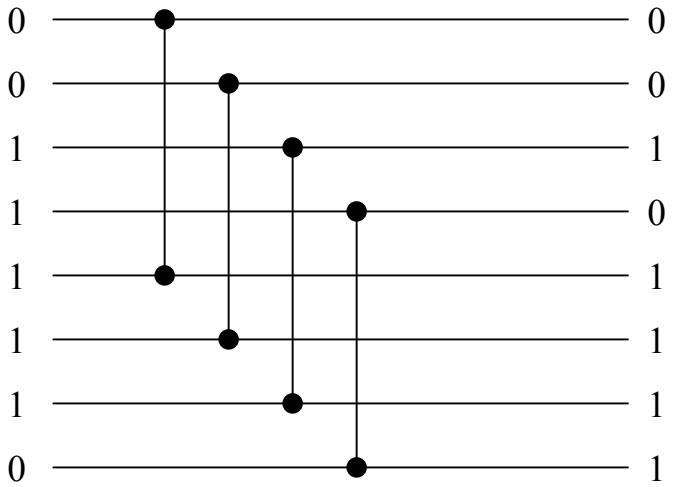
Yet More on Merging Networks

- Even better, we can modify our initial half-cleaner of our bitonic-sorter so that it acts on the second half upside down:

A modified half-cleaner like below on two sorted inputs:

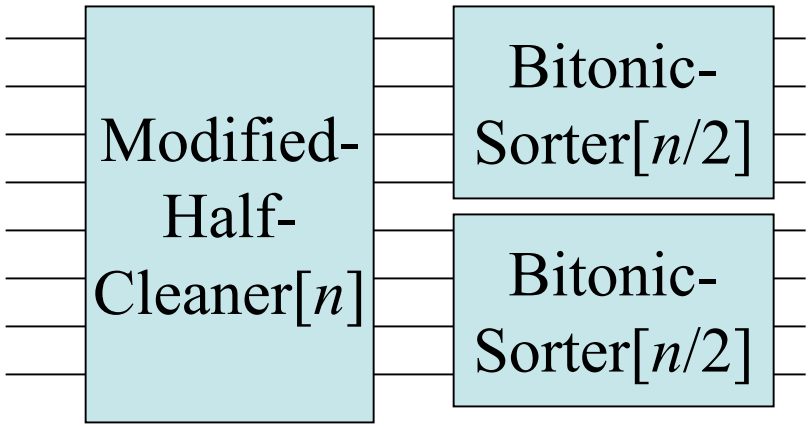


Has the same effect as a half-cleaner on the second half-reversed bitonic input:



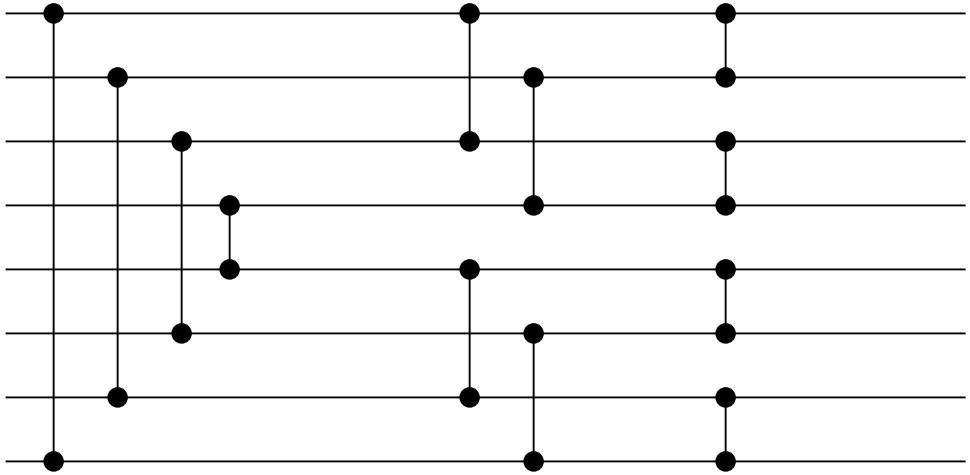
Even More on Merging Networks

- To be precise, we define a Modified-Half-Cleaner[n] to be the network with comparators between line i and line n-i+1.
- Our Merge-Network[n] will then look like:



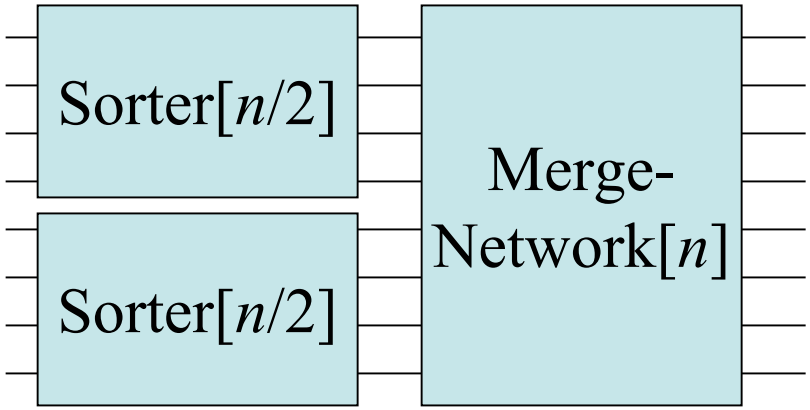
- It will have the same depth (lg n) as our bitonic sorter.

Example Merge-Network[8]



Sorter[n]

- Using our merge networks, we are now in a position to describe our complete sorting network, Sorter[n]:



- It has depth $D(n) = 0$ if $n=1$, $D(n/2) + \lg n$ if $n=2^k$ amd $k>=1$. So $D(n) = \Theta(\lg^2 n)$.