Sorted Networks

Sergey Bereg
Sorting Networks

inputs

outputs

sorted
Comparator (2-sorter)

inputs

outputs

\( x \)

\( y \)

\( \min(x, y) \)

\( \max(x, y) \)
Comparator (2-sorter)

- **Inputs:** $x$, $y$
- **Outputs:** $\min(x, y)$, $\max(x, y)$
Depth

- input wire of a comparison network has depth 0
- A comparator with two input wire of depth $d_x$ and $d_y$ has depth $\max(d_x, d_y) + 1$
Insertion Sort Network

inputs

outputs

depth $2n - 3$
Comparison Network

sorted
Comparison Network

How can we verify if a network sorts all possible input sequences?

not sorted
Testing with 0/1

<table>
<thead>
<tr>
<th>inputs</th>
<th>outputs</th>
</tr>
</thead>
</table>

Try all possible 0/1 sequences.
Testing with 0/1

Try all possible 0/1 sequences.
Testing with 0/1

Try all possible 0/1 sequences.
Testing with 0/1

Try all possible 0/1 sequences.

inputs | outputs
---|---
000 | 000
001 | 001
010 | 001
Testing with 0/1

Try all possible 0/1 sequences.

<table>
<thead>
<tr>
<th>inputs</th>
<th>outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>001</td>
<td>001</td>
</tr>
<tr>
<td>010</td>
<td>001</td>
</tr>
<tr>
<td>011</td>
<td>011</td>
</tr>
</tbody>
</table>
Testing with 0/1

Try all possible 0/1 sequences.

<table>
<thead>
<tr>
<th>inputs</th>
<th>outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>001</td>
<td>001</td>
</tr>
<tr>
<td>010</td>
<td>001</td>
</tr>
<tr>
<td>011</td>
<td>011</td>
</tr>
<tr>
<td>100</td>
<td>001</td>
</tr>
</tbody>
</table>
Testing with 0/1

Try all possible 0/1 sequences.

<table>
<thead>
<tr>
<th>inputs</th>
<th>outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>001</td>
<td>001</td>
</tr>
<tr>
<td>010</td>
<td>001</td>
</tr>
<tr>
<td>011</td>
<td>011</td>
</tr>
<tr>
<td>100</td>
<td>001</td>
</tr>
<tr>
<td>101</td>
<td>011</td>
</tr>
</tbody>
</table>
Testing with 0/1

Try all possible 0/1 sequences.

<table>
<thead>
<tr>
<th>inputs</th>
<th>outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>001</td>
<td>001</td>
</tr>
<tr>
<td>010</td>
<td>001</td>
</tr>
<tr>
<td>011</td>
<td>011</td>
</tr>
<tr>
<td>100</td>
<td>001</td>
</tr>
<tr>
<td>101</td>
<td>011</td>
</tr>
<tr>
<td>110</td>
<td>101</td>
</tr>
</tbody>
</table>

not sorted!
Sorting Network?
Testing with 0/1

Try all possible 0/1 sequences.
Testing with 0/1

Try all possible 0/1 sequences.

<table>
<thead>
<tr>
<th>inputs</th>
<th>outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
</tr>
</tbody>
</table>
Testing with 0/1

Try all possible 0/1 sequences.

<table>
<thead>
<tr>
<th>inputs</th>
<th>outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>001</td>
<td>001</td>
</tr>
</tbody>
</table>
Testing with 0/1

Try all possible 0/1 sequences.

<table>
<thead>
<tr>
<th>inputs</th>
<th>outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>001</td>
<td>001</td>
</tr>
<tr>
<td>010</td>
<td>001</td>
</tr>
<tr>
<td>011</td>
<td>011</td>
</tr>
<tr>
<td>100</td>
<td>001</td>
</tr>
<tr>
<td>101</td>
<td>011</td>
</tr>
<tr>
<td>110</td>
<td>011</td>
</tr>
<tr>
<td>111</td>
<td>111</td>
</tr>
</tbody>
</table>

all sorted!
Zero-One Principle

If a comparison network sorts all possible sequences of 0’s and 1’s correctly, then it sorts all sequences of arbitrary numbers correctly.
Given

For a monotonically increasing function $f$,

$$a_0 \rightarrow b_0 \quad f(a_0) \rightarrow f(b_0)$$

$$a_1 \rightarrow b_1 \quad f(a_1) \rightarrow f(b_1)$$
Proof: Lemma

\[
\begin{align*}
& a_0 & \Rightarrow & \min(a_0, a_1) \\
& a_1 & \Rightarrow & \max(a_0, a_1)
\end{align*}
\]
Proof: Lemma

\[ f(a_0) \rightarrow \min(f(a_0), f(a_1)) \]

\[ f(a_1) \rightarrow \max(f(a_0), f(a_1)) \]
Proof: Lemma

\( f \) is monotonically increasing:

\[ x \leq y \implies f(x) \leq f(y) \]

\[ f(a_0) \quad \text{min}(f(a_0), f(a_1)) \]

\[ f(a_1) \quad \text{max}(f(a_0), f(a_1)) \]
Proof: Lemma

$f$ is monotonically increasing:
\[ x \leq y \implies f(x) \leq f(y) \]

\[ f(a_0) \quad \xrightarrow{\quad} \quad f(\min(a_0, a_1)) \]

\[ f(a_1) \quad \xrightarrow{\quad} \quad f(\max(a_0, a_1)) \]
Proof: Lemma

\( f \) is monotonically increasing:

\[ x \leq y \implies f(x) \leq f(y) \]

\[ f(a_0) \quad \quad \quad f(b_0) \]

\[ f(a_1) \quad \quad \quad f(b_1) \]
Generalization

Given
For a monotonically increasing function $f$, 

**Generalization**

(by induction)
Proof: Zero-One Principle

Suppose

a) the network sorts all sequences of 0’s and 1’s,
b) there exists a sequence $a_0, a_1, \ldots, a_{n-1}$ that it doesn’t sort, i.e.,

$\exists a_i, a_j$ such that $a_i < a_j$

but $a_j$ is placed before $a_i$ in the output.

Define

$$f(x) = \begin{cases} 
0 & \text{if } x \leq a_i \\
1 & \text{otherwise}
\end{cases}$$
Proof: Zero-One Principle
Proof: Zero-One Principle

\[
\begin{align*}
&\quad \text{Sorting Network} \\
&f(a_0) \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad f(a_0') \\
f(a_1) \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad f(a_1') \\
\vdots \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \vdots \\
f(a_{n-1}) \quad \quad \quad \quad \quad \quad \quad f(a'_{n-1})
\end{align*}
\]
Proof: Zero-One Principle

\[ f(a_0) \]
\[ f(a_1) \]
\[ \vdots \]
\[ f(a_{n-1}) \]

\[ f(a'_0) \]
\[ f(a'_1) \]
\[ \vdots \]
\[ f(a'_{n-1}) \]

\[ 1 \]
\[ 0 \]

\{ contradiction! \}
First, we design a network that can sort only bitonic sequences.

Take a sequence that monotonically increases and then monotonically decreases and shift it circularly (shift can be 0). The result is a bitonic sequence.

Example: \((1, 4, 6, 8, 3, 2), (6, 9, 4, 2, 3, 5), \) and \((9, 8, 3, 2, 4, 6)\) are bitonic.

Is the sequence \((1, 4, 3, 8, 3, 2)\) bitonic?
First, we design a network that can sort only bitonic sequences.

Take a sequence that monotonically increases and then monotonically decreases and shift it circularly (shift can be 0). The result is a bitonic sequence.

Example: (1,4,6,8,3,2),(6,9,4,2,3,5), and (9,8,3,2,4,6) are bitonic.

(1,4,3,8,3,2) is not bitonic.
By 0/1 principle, it suffices to sort only bitonic 0/1 sequences.

*Half-cleaner* splits bitonic sequence into two bitonic sequences such that one of them is *clean*, i.e. consists of all ‘0’s or all 1’s.

Proof. Suppose that the input is 00...011...100..0 (the case 11..100..011..1 is symmetric).
What is the depth of $\text{HALF-CLEANER}[n]$?
The depth of $\text{HALF-CLEANER}[n]$ is 1.
Bitonic Sorter

Construct bitonic sorter recursively.
Bitonic Sorter

Construct bitonic sorter recursively.

What is the depth of Bitonic-Sorter\([n]\)?
Bitonic Sorter

Construct bitonic sorter recursively.

Let $D(n)$ be the depth of Bitonic-Sorter[$n$].

Recurrence $D(n) = D(n/2) + 1$.

Solution $D(n) = \lg n$. 
Use divide-and-conquer:

- divide the sequence into two,
- sort them at the same time, and
- merge the sorted sequences into one sorted sequence.

So, we just need to create Merger[\(n\)].
Given two sorted sequences $X$ and $Y$, concatenate $X$ and reverse $Y$; then $XY^R$ is bitonic sequence.

Example: $X = 00000111$, $Y = 0001111$, and $XY^R = 000011111110000$.

Idea: “reverse” wires of second half and use bitonic sorter. Better idea: instead of reversing wires, we modify the first half-cleaner of bitonic sorter.

The depth of $\text{MERGER}[n]$ is $\lg n$. 
Sorting Networks

What is the depth of the sorting network?

Recurrence $D(n) = D(n/2) + \log n$. 
What is the depth of the sorting network?

Recurrence $D(n) = D(n/2) + \lg n$.

Solution $D(n) = \lg^2 n$. 
Sorting Network for $n = 4$
Sorting Network for $n = 8$

Sorter[8]:

![Sorting Network Diagram](image-url)