TD n° 1. Sorting networks

Hugo Sacau

Q1. Let α be a sorting network.

"$\Rightarrow$" Suppose α sorts $\langle n, \dots, 1\rangle$. For all $i \in [\![1, n]\!]$, we define $f_i : j \mapsto \begin{cases} 1 & \text{if } i \leq j \\ 0 & \text{else.} \end{cases}$

We have that $\alpha(\langle n, \dots, 1\rangle) = \langle 1, \dots, n\rangle$

Then, for all $i \in [\![1, n]\!]$,

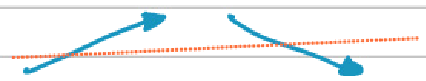
$$\alpha(f_i(\langle n, \dots, 1\rangle)) = \alpha(\langle \overbrace{1, \dots, 1}^{i}, 0, \dots, 0\rangle)$$
$$= f_i(\langle 1, \dots, n\rangle)$$
$$= \langle 0, \dots, 0, 1, \dots, 1\rangle.$$

"$\Leftarrow$" Suppose $a := \alpha(\langle n, \dots, 1\rangle)$ is unsorted: there exists $i \in [\![1, n-1]\!]$ such that $a_i < a_{i+1}$.

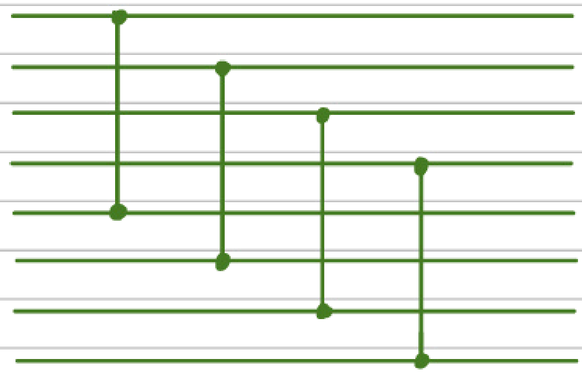
Define $b := \alpha(f_i(\langle n, \dots, 1\rangle))$. Thus $b_i = 1$ and $b_{i+1} = 0$ with the lemma. Therefore, $\alpha(f_i(\langle n, \dots, 1\rangle))$ is unsorted.

Q2. Yes it does!

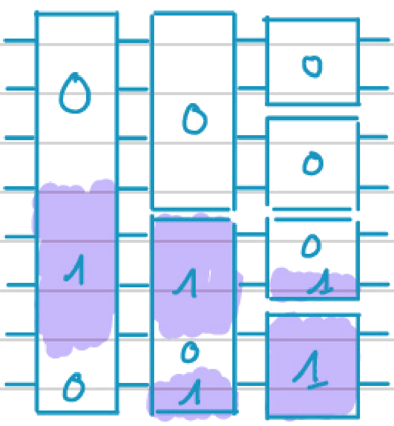
Intuition:



Example of a separator with 8 inputs



Q3 (a) We use the following network



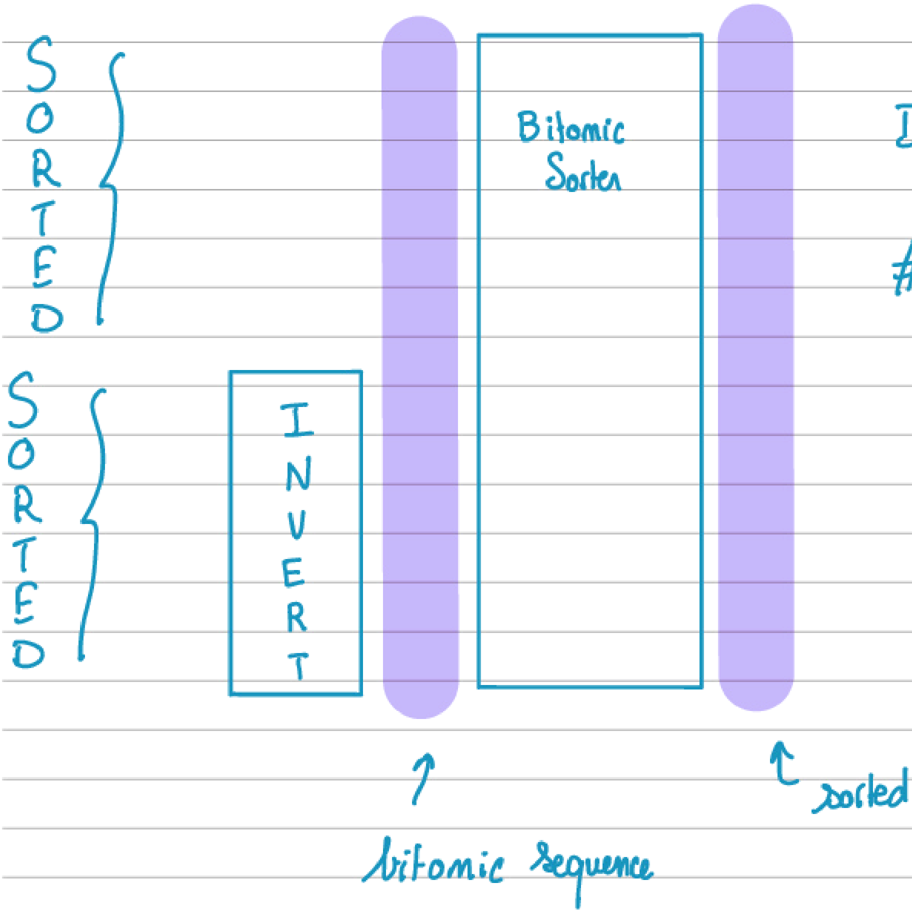
To see why it is correct, we show by induction that, for all binary bitonic sequences, they are correctly sorted.

Each step has the following property:

The current block is divided in two blocks such that one block is constant and the other is bitonic.

With $n = 2^m$ inputs, we have a network with depth $m = \log_2 n$ and $(n \cdot m)/2$ comparators.

Q.3(b)



Depth : $\underbrace{(\log n)}_{} \times \underbrace{(\log n)}_{\text{bitonic sorter}}$

\# operators : $\frac{n}{2}(\log n)(\log n)$

Q4(a)

$$\begin{bmatrix} 1 & 3 & 5 & 6 \\ 11 & 8 & 16 & 10 \\ 4 & 7 & 2 & 9 \\ 14 & 13 & 15 & 12 \end{bmatrix} \xrightarrow{\text{Step 1}} \begin{bmatrix} 1 & 5 & 3 & 6 \\ 11 & 16 & 8 & 10 \\ 4 & 2 & 7 & 9 \\ 14 & 15 & 13 & 12 \end{bmatrix}$$

$$\xrightarrow{\text{Step 2}} \begin{bmatrix} 1 & 2 & 3 & 6 \\ 5 & 4 & 8 & 7 \\ 11 & 14 & 9 & 10 \\ 16 & 15 & 13 & 12 \end{bmatrix} \xrightarrow{\text{Step 3}} \begin{bmatrix} 1 & 2 & 3 & 4 \\ 8 & 7 & 6 & 5 \\ 9 & 10 & 11 & 12 \\ 16 & 15 & 14 & 13 \end{bmatrix}$$

Q4(b) This can be done with an odd-even swap (like odd-even sort), thus we get a complexity of $n/2 - 1$ neighbour-swapping steps.