7 Sorting Networks

In this section we show how to construct networks that can efficiently sort numbers. Since permutation routing problems can be viewed as sorting problems, these networks allow also the efficient routing of arbitrary permutations.

Let $\bar{x} = (x_1, \ldots, x_n)$ be a sequence of natural numbers. To simplify the description in the following, we will always assume that n is a power of 2. The comparison and exchange of elements in this sequence is done with the help of *comparators*. Each comparator is represented by a pair $(i, j) \in \{1, \ldots, n\}^2$ with $i \neq j$. The effect of a comparator (i, j) is that if $x_i > x_j$, then their numbers are exchanged (if $x_i \leq x_j$, nothing happens). A *comparator network of depth* $d(n)$ consists of $d(n)$ levels of comparators. Within a level, every index $i \in \{1, \ldots, n\}$ is only allowed to be used once in a comparator. Hence, the maximum number of comparators in a level is $n/2$. See Figure 1 for an example of a comparator network.

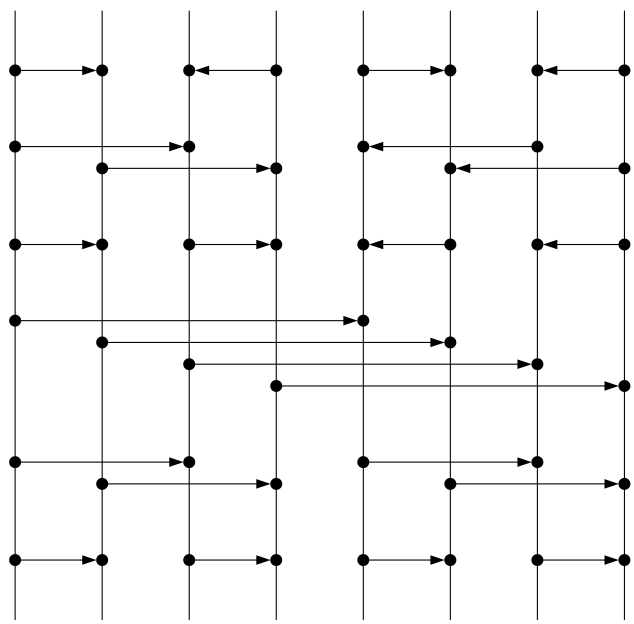

Figure 1: An example of a comparator network.

A comparator network is called a *sorting network* if for every sequence $(x_1, \ldots, x_n) \in \mathbb{N}$ the outcome of the comparisons is a sequence of the x_i sorted in increasing order. Before we show how to construct an efficient sorting network, we prove a fundamental principle.

7.1 The 0-1-principle

In this section we show that for comparator networks it suffices to prove that a sorting network $G(n)$ sorts all possible 0-1-sequences of length n (i.e., all sequences of n binary numbers) in order to be sure that it also sorts all other possible sequences of n numbers.

Theorem 7.1 *Let $G(n)$ be a comparator network. $G(n)$ sorts all sequences of natural numbers of length n if and only if it sorts all 0-1-sequences of length n.*

Proof. $\Rightarrow$: This is obvious.
$\Leftarrow$: We prove this in an indirect way.
Suppose that there is a comparator network $G(n)$ that sorts all 0-1-sequences, but does not sort some sequence $\bar{x} = (x_1, \ldots, x_n) \in \mathbb{N}^n$.

Let π denote the permutation of $\bar{x}$ at the end of the sorting network. Since $\bar{x}$ has not been sorted, there must be some smallest index i_0, $2 \leq i_0 \leq n$, so that $x_{\pi(i_0-1)} > x_{\pi(i_0)}$. We define now a 0-1-sequence $\bar{y}$ as follows:

$$y_j = \begin{cases} 0 & \text{if } x_j \leq x_{\pi(i_0)} \\ 1 & \text{if } x_j > x_{\pi(i_0)} \end{cases}$$

Does $G(n)$ sort this sequence? No, because we will show that for all levels $k \in \{1, \ldots, d(n)\}$ the permutation of $\bar{x}$ at the end of level k is equal to the permutation of $\bar{y}$ at the end of level k. Thus, since $\bar{x}$ is not sorted at the end, also $\bar{y}$ cannot be sorted at the send. To prove this property, we use an induction argument over all levels. At the beginning (which may be interpreted as the end of level 0), the permutation of $\bar{x}$ and $\bar{y}$ is certainly identical. Now assume that it is alreay known that up to some level $k \geq 0$ this property holds. Then we will show that it also holds for level $k+1$. Consider any comparator (ℓ, m) in this level, and let i and j denote the indexes of the elements in $\bar{x}$ and $\bar{y}$ that arrive at ℓ and m. According to the definition of $\bar{y}$ we have that if $x_i \leq x_j$ then $y_i \leq y_j$, and if $x_i > x_j$ then $y_i \geq y_j$. Hence, in both cases we can assume the comparator to work in an identical way. (If $y_i = y_j$, then we simply look at the operation performed for x_i and x_j in order to decide whether to exchange y_i with y_j or not.) This completes the proof of the theorem. □

Note that the 0-1-principle only holds for sorting networks based on compare-and-exchange operations. For other operations, this principle may not hold.

7.2 Construction of a sorting network

In this section we will show how to construct a sorting network called *Batcher's sorting network*. Due to the 0-1-principle, we will only deal with 0-1-sequences in the following.

Definition 7.2 *A 0-1-sequence $\bar{x}$ of length n is called* bitonic *if there is an index i so that $(x_1, \ldots, x_i)$ is monotonically increasing and $(x_{i+1}, \ldots, x_n)$ is monotonically decreasing, or vice versa. A comparator network is called a* bitonic sorting network *if it sorts all bitonic sequences.*

We start with a network B_n that, as we will show, sorts all bitonic 0-1-sequences $\bar{x}$ of length n. It's recursive definition can be found in Figure 2.

Lemma 7.3 *B_n is a bitonic sorting network of depth $\log n$ and consists of $\frac{1}{2}n \log n$ comparators.*

Proof. Let $\bar{x} = (x_1, \ldots, x_n)$ be the bitonic 0-1-sequence that is given to B_n, and let $(y_1, \ldots, y_n)$ be the permuted values of $\bar{x}$ after the first level of B_n. Furthermore, let $(z_1, \ldots, z_n)$ be the permuted values of $\bar{x}$ at the end of B_n. We will show by induction that for all $m = 2^k$ with $k \in \{1, \ldots, \log n\}$ the network B_m is a bitonic sorting network. For $m = 1$ this is certainly true. So consider now some B_m for which we know the hypothesis to hold for $B_{m/2}$. Suppose that the following two properties were true:

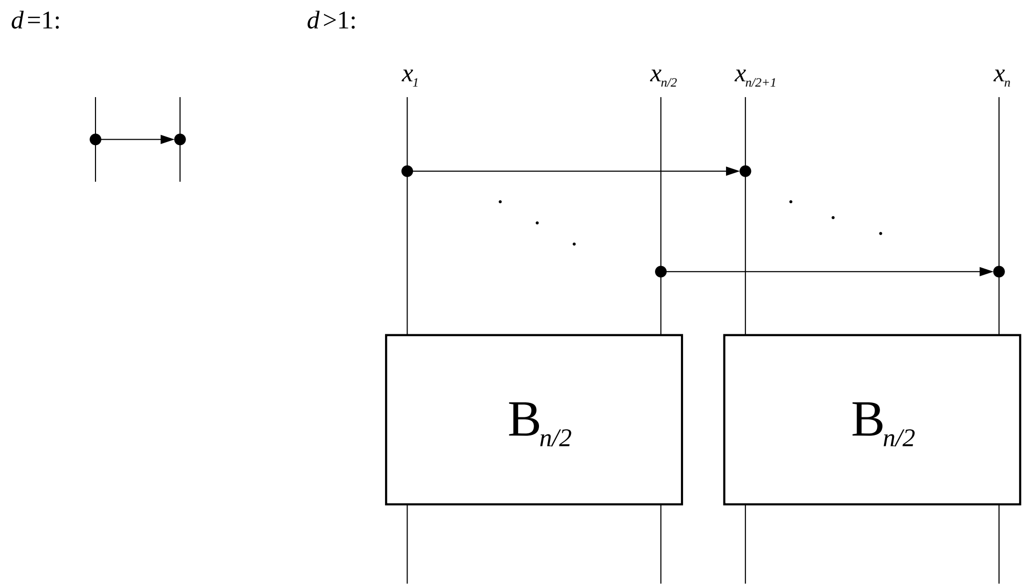

Figure 2: Recursive definition of the network B_n.

(i) $(y_1, \ldots, y_{m/2})$ and $(y_{m/2+1}, \ldots, y_m)$ are bitonic, and

(ii) $(y_1, \ldots, y_{m/2}) = (0, \ldots, 0)$ or $(y_{m/2+1}, \ldots, y_m) = (1, \ldots, 1)$.

Since by the induction hypothesis both $(z_1, \ldots, z_{m/2})$ and $(z_{m/2+1}, \ldots, z_m)$ are sorted, we obtain from (ii) that also $\bar{z}$ must be sorted. Hence, it remains to show properties (i) and (ii). We will only concentrate on bitonic sequences that are first increasing and than decreasing. For the other bitonic sequences, similar arguments can be used.

1. $\bar{x} = (\underbrace{0, \ldots, 0}_{\ell}, \underbrace{1, \ldots, 1}_{k} \,|\, \underbrace{1, \ldots, 1}_{j}, \underbrace{0, \ldots, 0}_{i})$ with $\ell + k = j + i = m/2$.

   (a) $k \leq i$: Then we have $(y_1, \ldots, y_{m/2}) = (0, \ldots, 0)$ and $(y_{m/2+1}, \ldots, y_m) = (\underbrace{1, \ldots, 1}_{j}, 0, \ldots, 0, \underbrace{1, \ldots, 1}_{k})$.

   (b) $k > i$: Then we have $(y_{m/2+1}, \ldots, y_m) = (1, \ldots, 1)$ and $(y_{m/2+1}, \ldots, y_m) = (\underbrace{0, \ldots, 0}_{\ell}, 1, \ldots, 1, \underbrace{0, \ldots, 0}_{i})$.

2. $\bar{x} = (\underbrace{0, \ldots, 0}_{m/2} \,|\, \underbrace{0, \ldots, 0, 1, \ldots, 1, 0, \ldots, 0}_{m/2})$: Then $\bar{y} = \bar{x}$.

3. $\bar{x} = (\underbrace{0, \ldots, 0, 1, \ldots, 1, 0, \ldots, 0}_{m/2} \,|\, \underbrace{0, \ldots, 0}_{n/2})$: Then $\bar{z} = (x_{m/2+1}, \ldots, x_m, x_1, \ldots, x_{m/2})$.

Hence, in all cases (i) and (ii) are fulfilled.

Let $d(n)$ be the depth of B_n. Then we have

$$\left.\begin{array}{ll} n=2: & d(2)=1 \\ n>2: & d(n)=t(n/2)+1 \end{array}\right\} \quad \Rightarrow \quad d(n)=\log n$$

Since there are $n/2$ comparators in each level, B_n altogether contains $\frac{1}{2}n\log n$ comparators. $\square$

The bitonic sorting network can now be used as shown in Figure 3 to construct a sorting network S_n, called Batcher's sorting network. $\bar{S}_{n/2}$ in this figure means the mirror image of the network $S_{n/2}$.

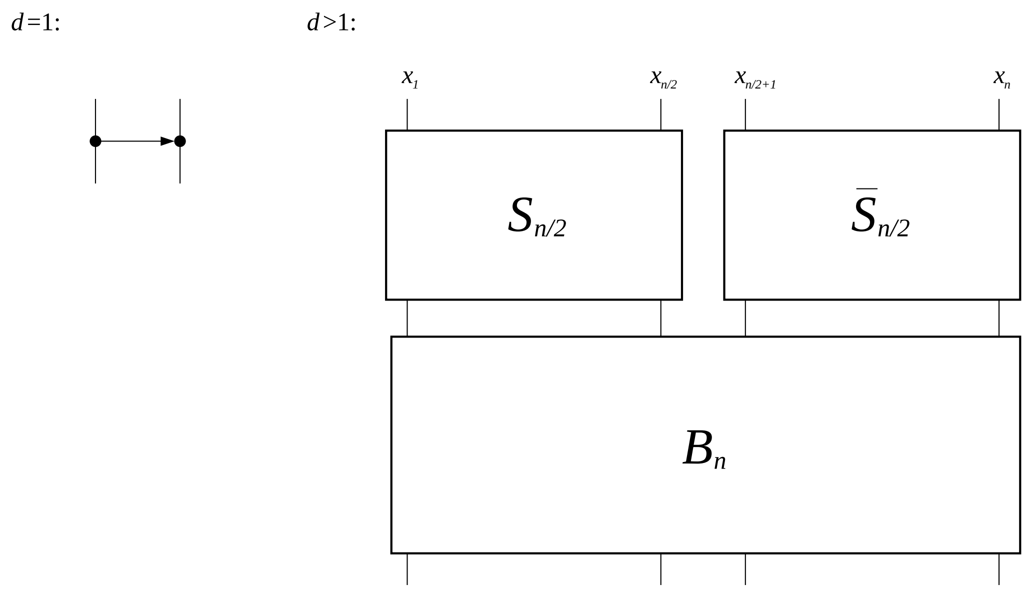

Figure 3: Recursive definition of the network S_n.

Theorem 7.4 *S_n is a sorting network of depth $\frac{1}{2}\log n(\log n+1)$ and consists of $\frac{1}{4}n\log n(\log n+1)$ comparators.*

Proof. If $S_{n/2}$ sorts all 0-1-sequences of length $n/2$ in increasing order and $\bar{S}_{n/2}$ sorts all 0-1-sequences of length $n/2$ in decreasing order, then the concatenation of these ouput sequences results in a bitonic sequence that is sorted by B_n according to Lemma 7.3. Thus, it follows easily by induction that S_n sorts any 0-1-sequence of length n.

Let $d(n)$ be the depth of S_n. Then we get that

$$\left.\begin{array}{ll} n=2: & d(2)=1 \\ n>2: & d(n)=d(n/2)+\log n \end{array}\right\} \quad \Rightarrow \quad d(n)=\frac{1}{2}\log n(\log n+1)$$

Since the number of comparators is equal to $d(n)\cdot n/2$, the theorem follows. $\square$

Many more fundamental statements about sorting networks can be found in Knuth's book. The first that were able to construct a sorting network of depth $O(\log n)$ were Ajtai, Komlós, and

Szeméredi. Due to these authors, the network is known as AKS network. An understandable description of this network has been done by Paterson. Since the constant hidden in the $O(\log n)$ is very large, for all practical purposes Batcher's sorting network is much better than the AKS network.

7.3 References

- M. Ajtai, J. Komlós, and E. Szeméredi. Sorting in $c \cdot \log n$ parallel steps. *Combinatorica*, 3:1–19, 1983.
- K.E. Batcher. Sorting networks and their applications. In *AFIPS Conf. Proc. 32*, pp. 307–314, 1968.
- D.E. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching.* Addison-Wesley, Reading, Massachusetts, 1973.
- M.S. Paterson. Improved sorting networks with $O(\log n)$ depth. *Algorithmica*, 5:75–92, 1990.