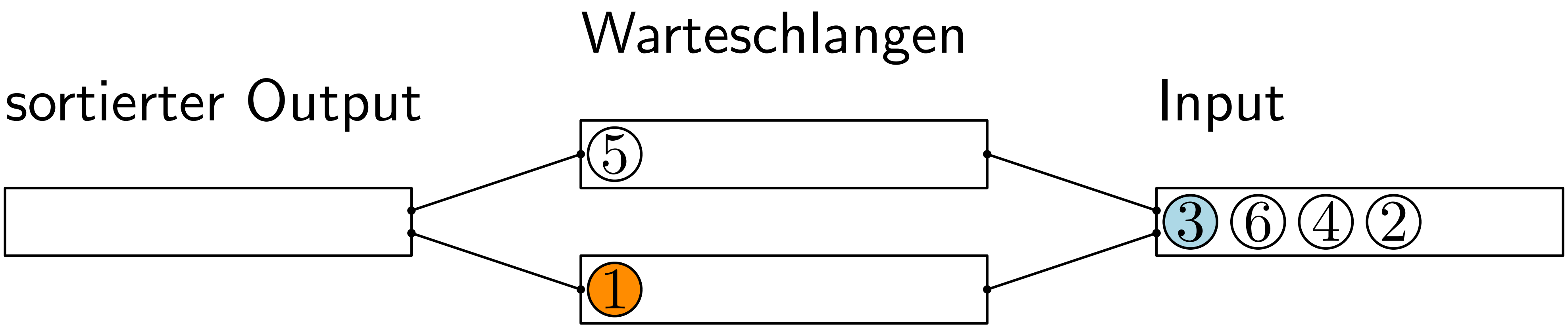
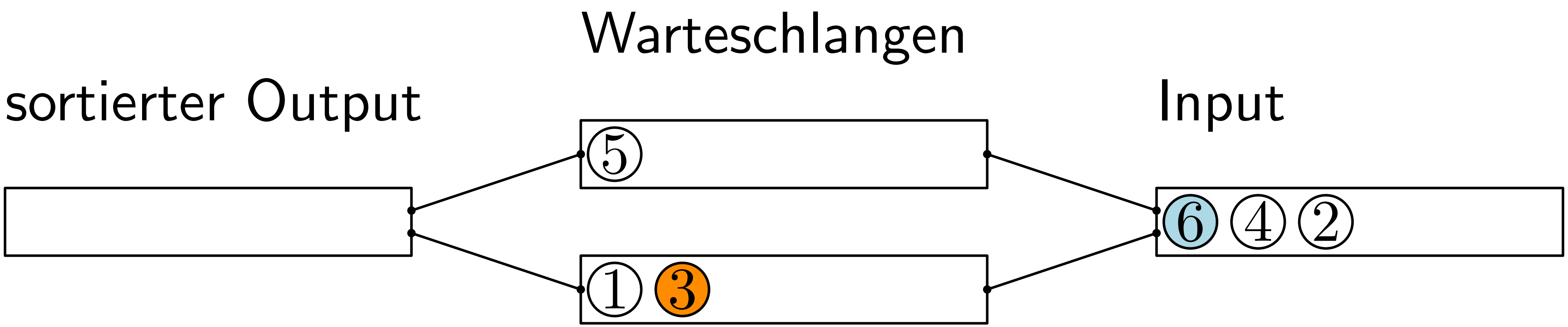
Sortieren mit Warteschlangen

Sortieren mit Warteschlangen

Sortieren mit Warteschlangen

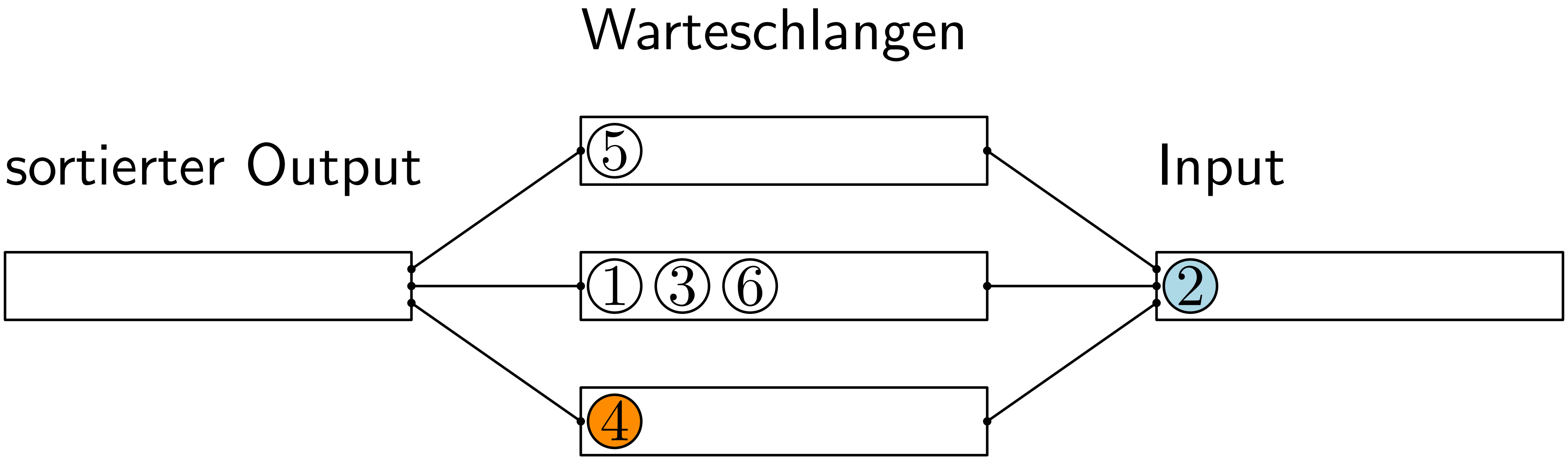


Sortieren mit Warteschlangen

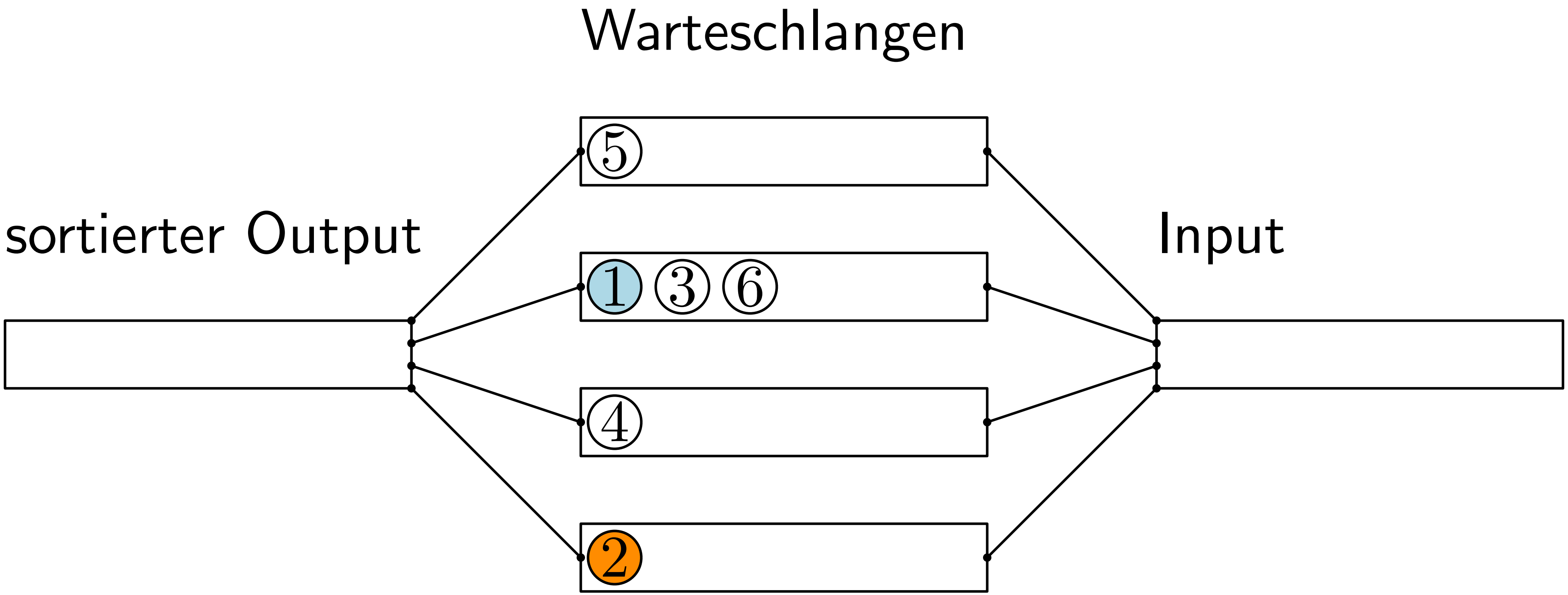


Sortieren mit Warteschlangen

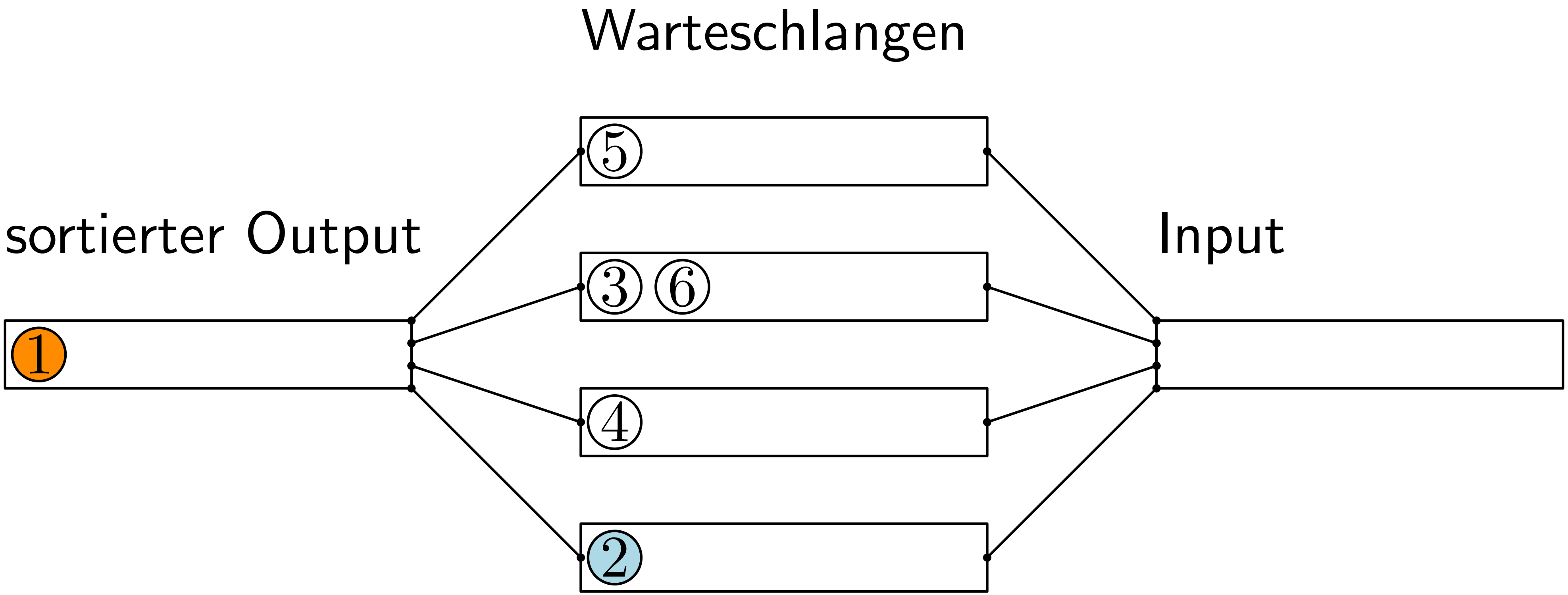
Sortieren mit Warteschlangen



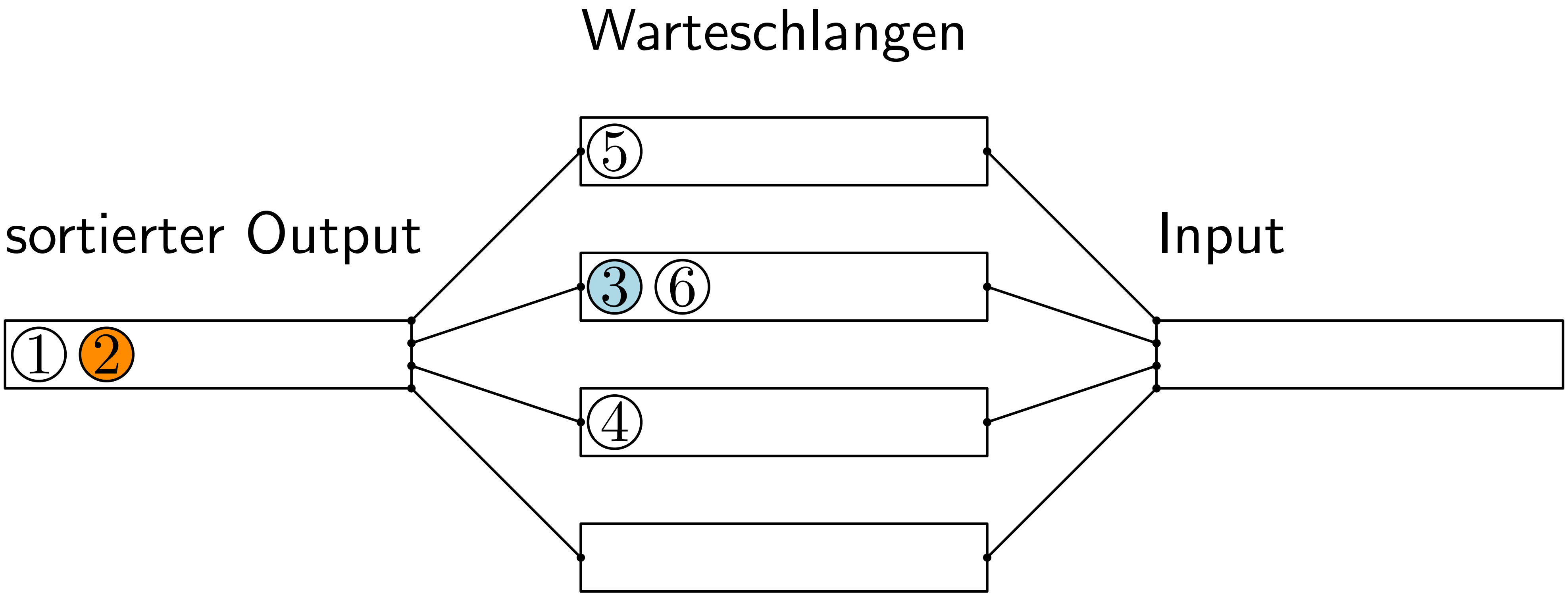
Sortieren mit Warteschlangen



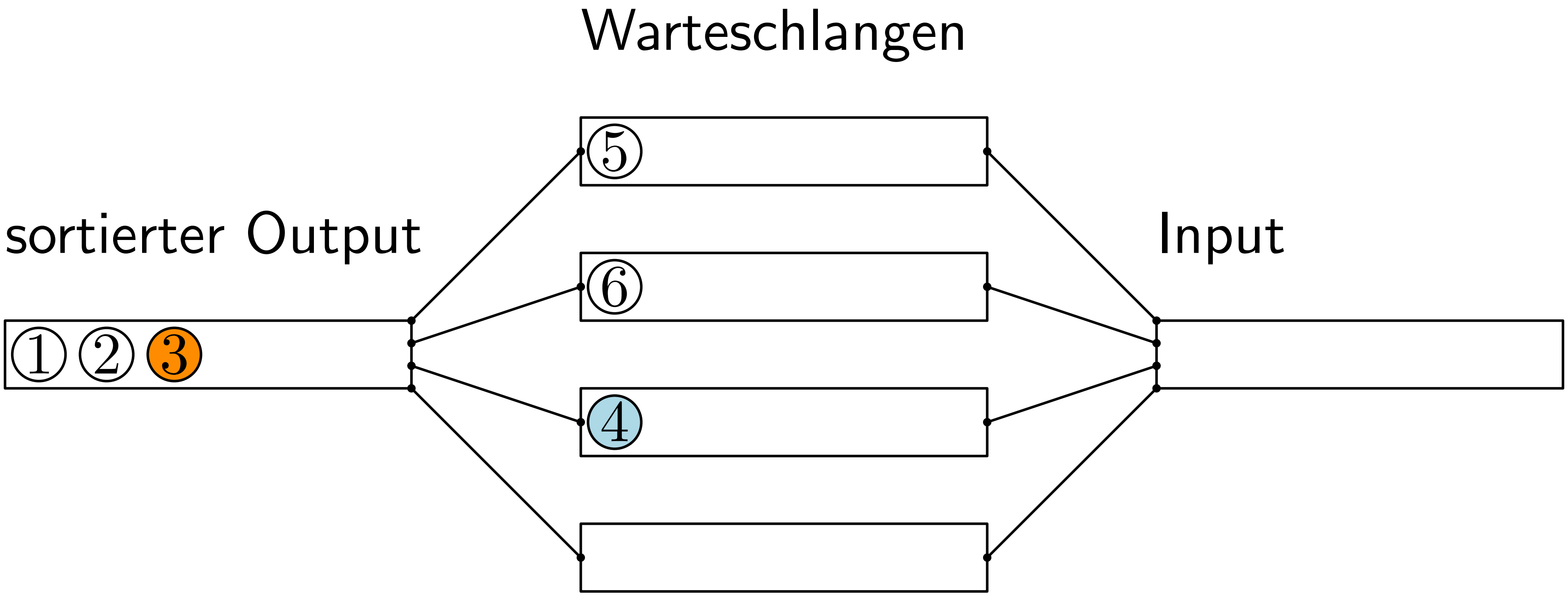
Sortieren mit Warteschlangen



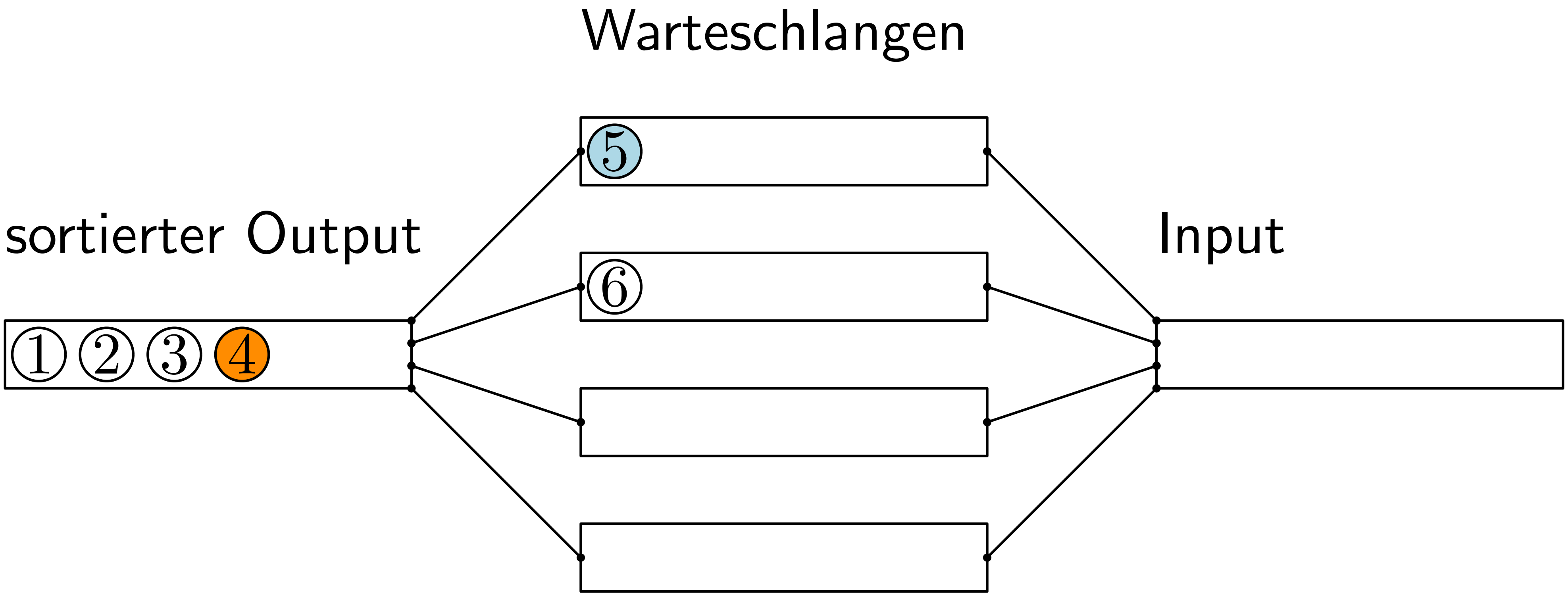
Sortieren mit Warteschlangen



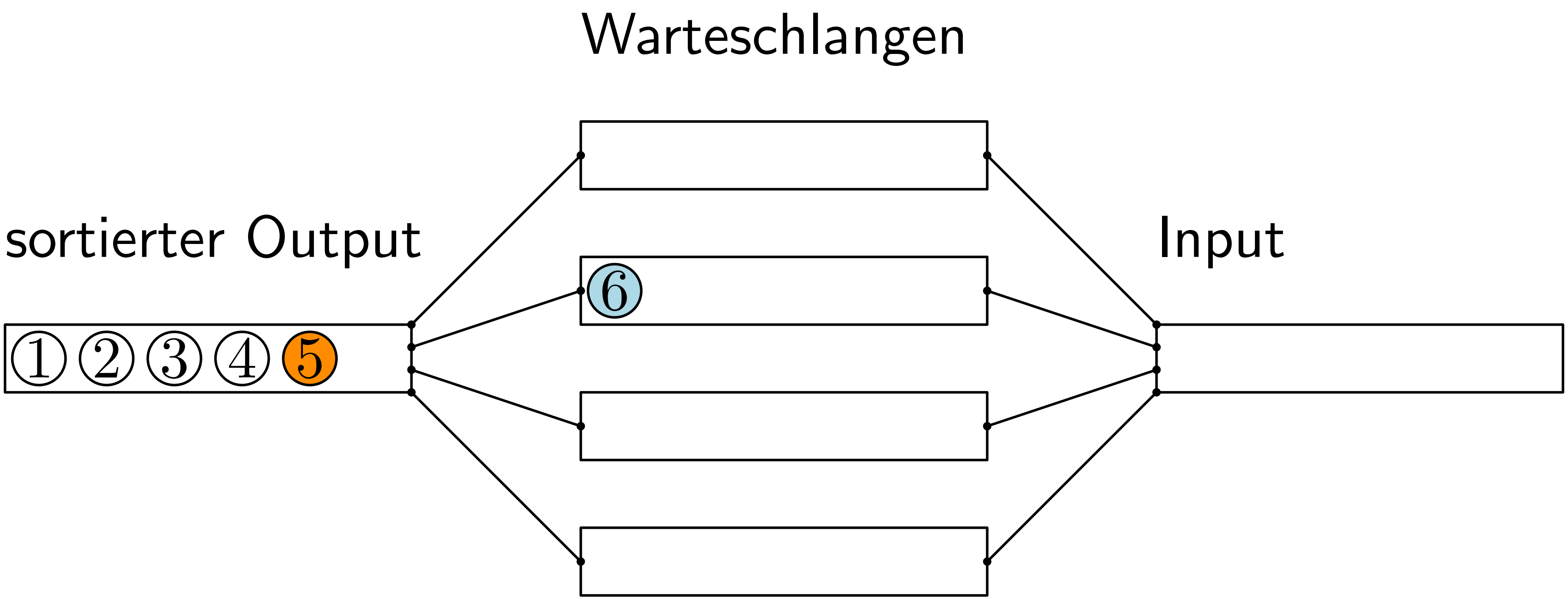
Sortieren mit Warteschlangen



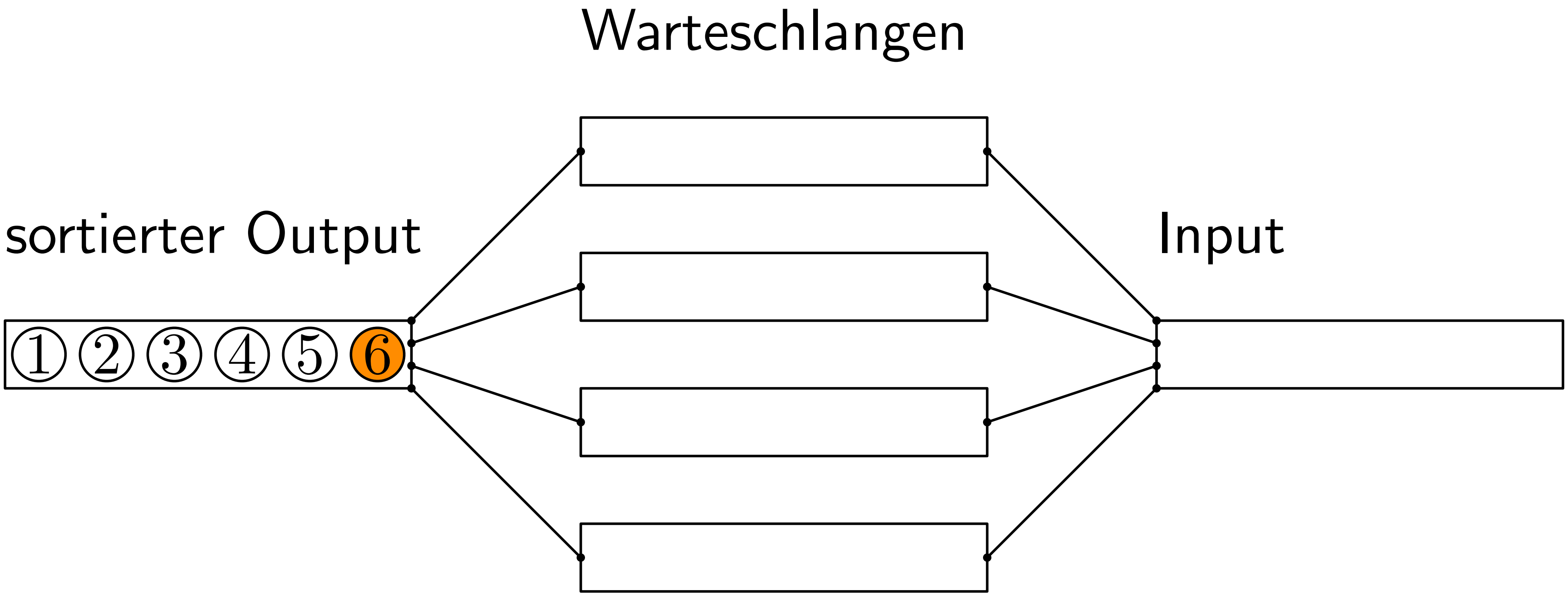
Sortieren mit Warteschlangen



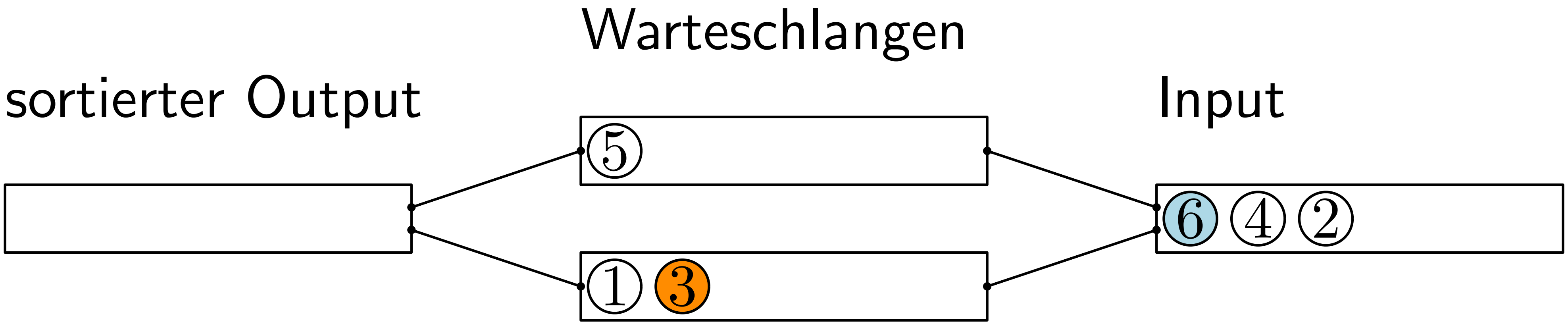
Sortieren mit Warteschlangen



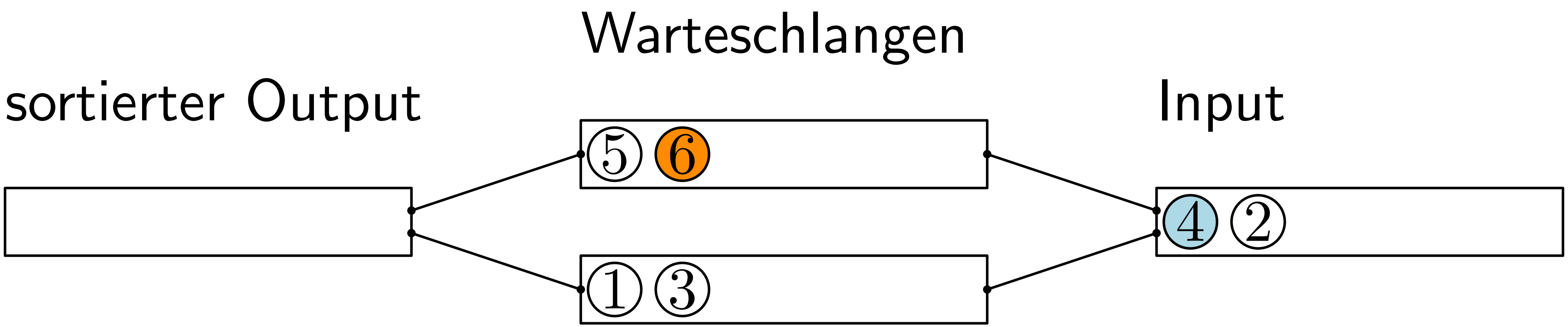
Sortieren mit Warteschlangen



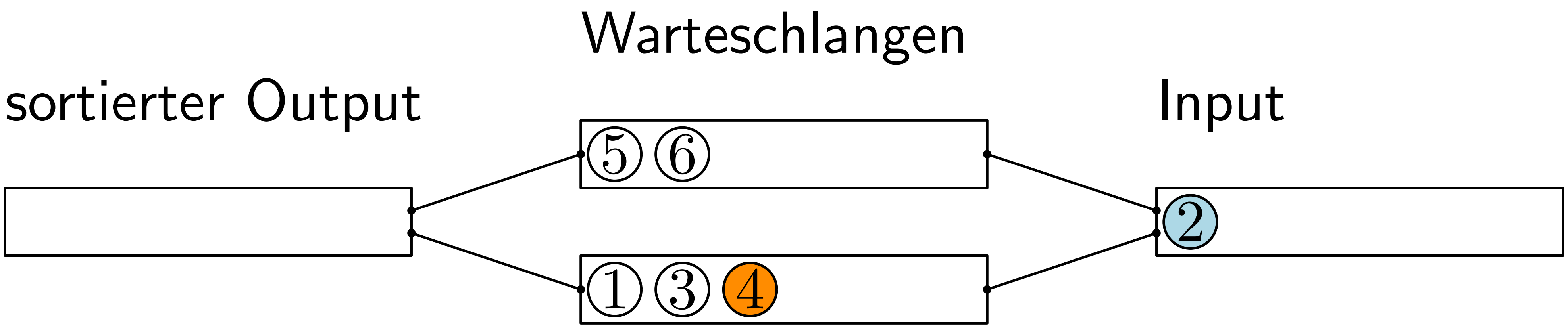
Sortieren mit Warteschlangen



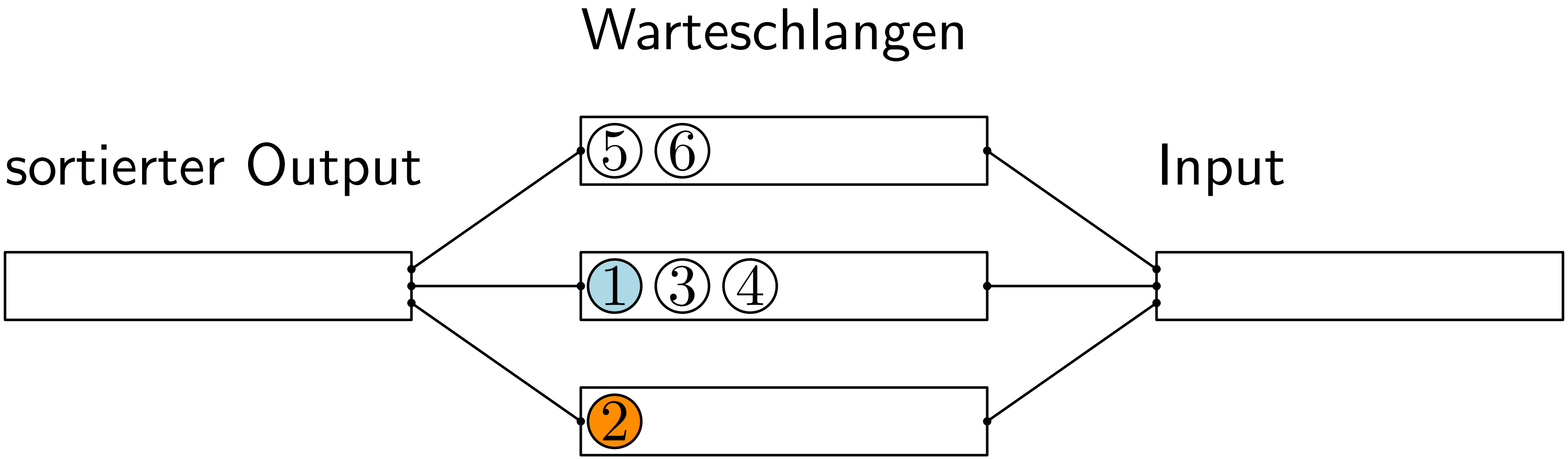
Sortieren mit Warteschlangen



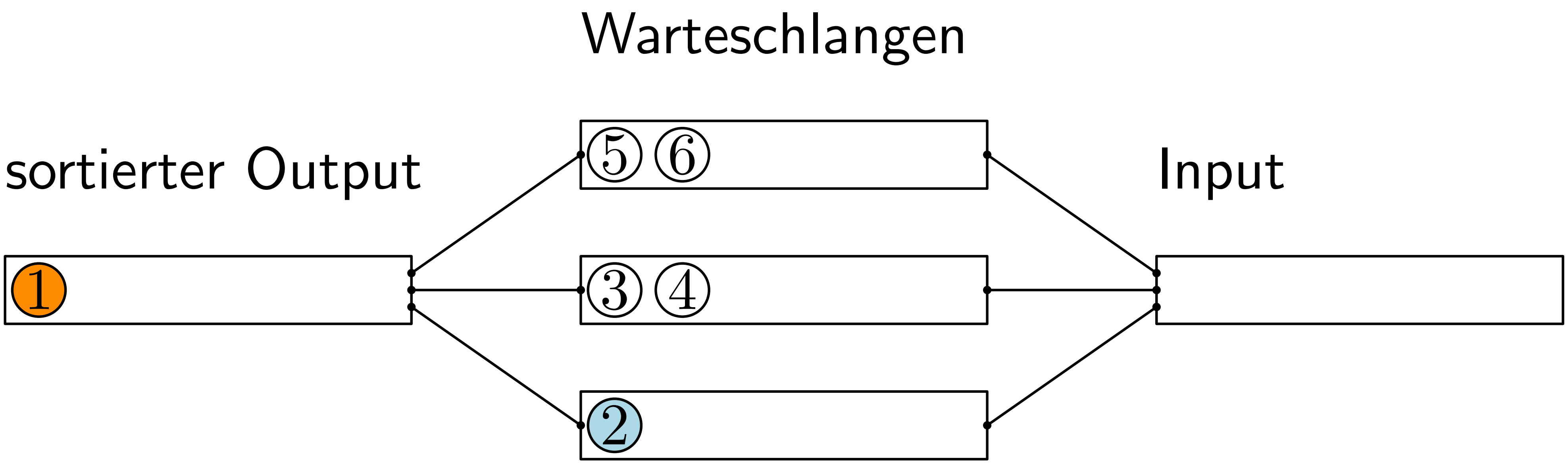
Sortieren mit Warteschlangen



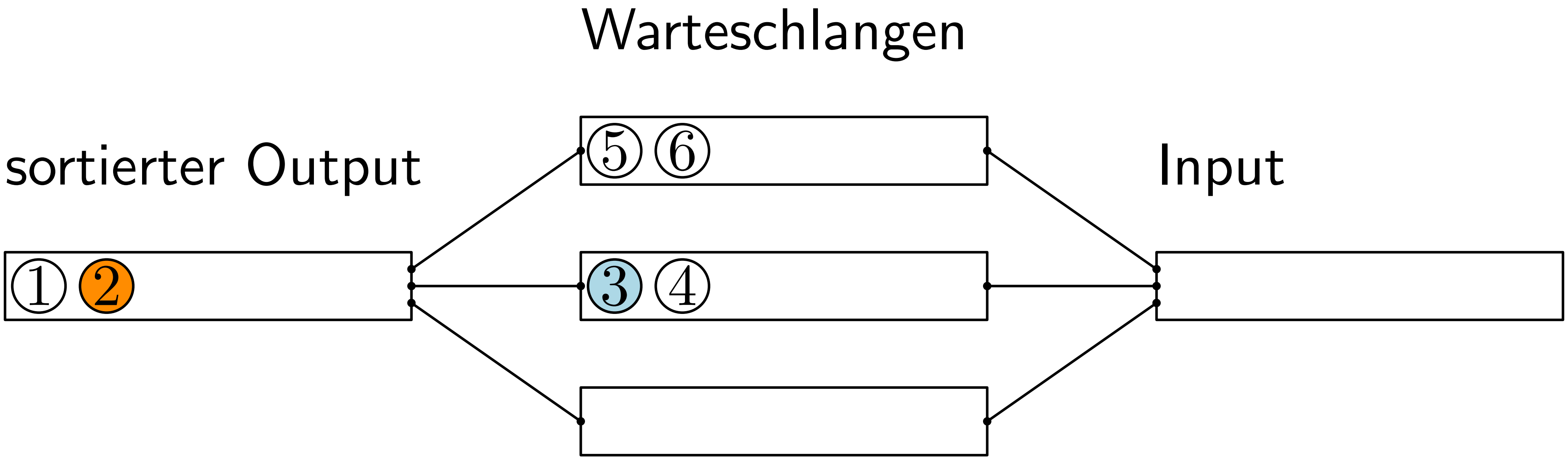
Sortieren mit Warteschlangen



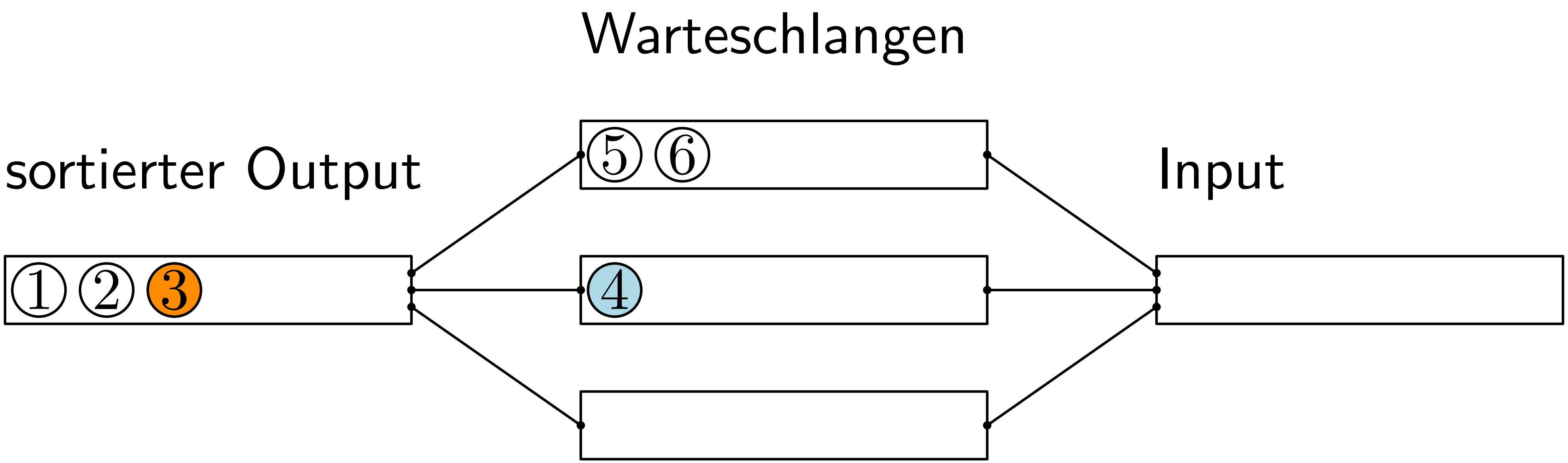
Sortieren mit Warteschlangen



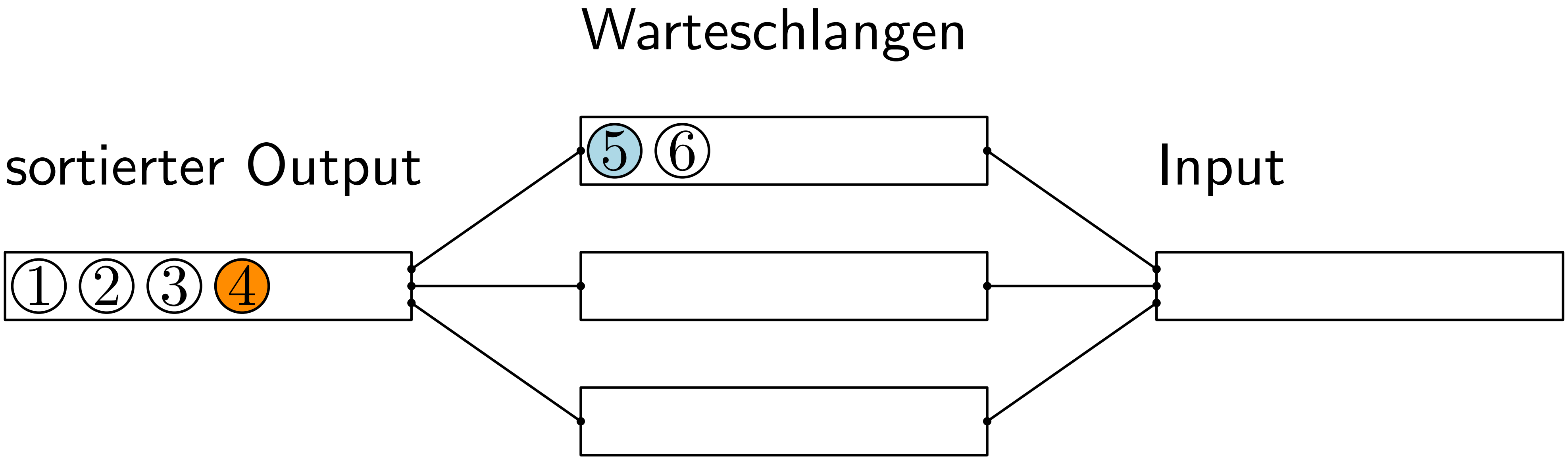
Sortieren mit Warteschlangen



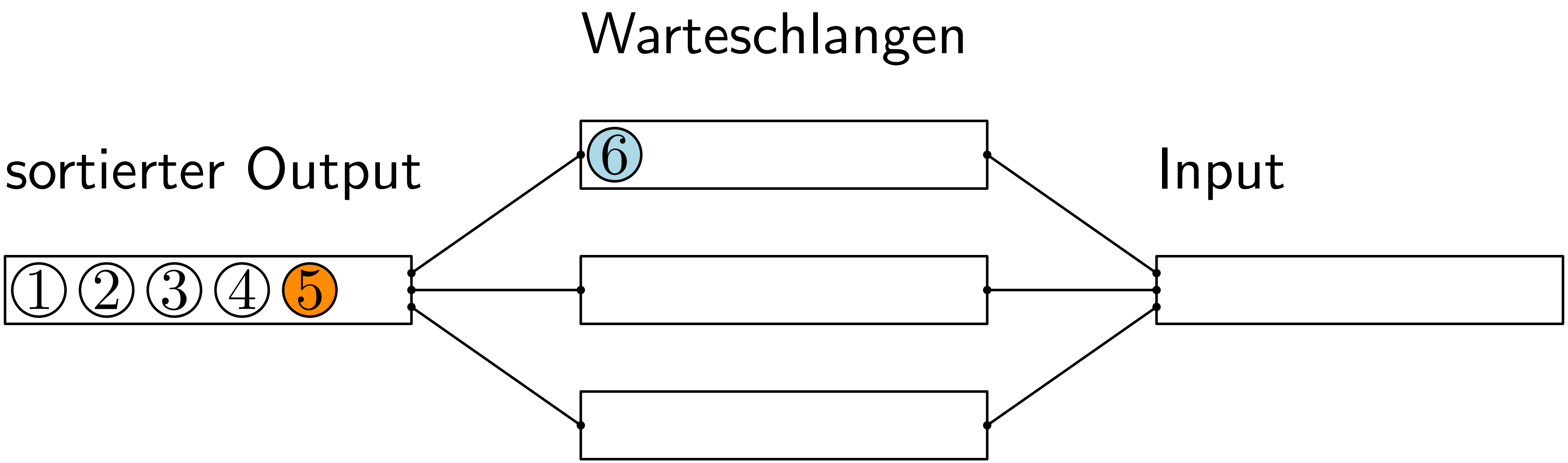
Sortieren mit Warteschlangen



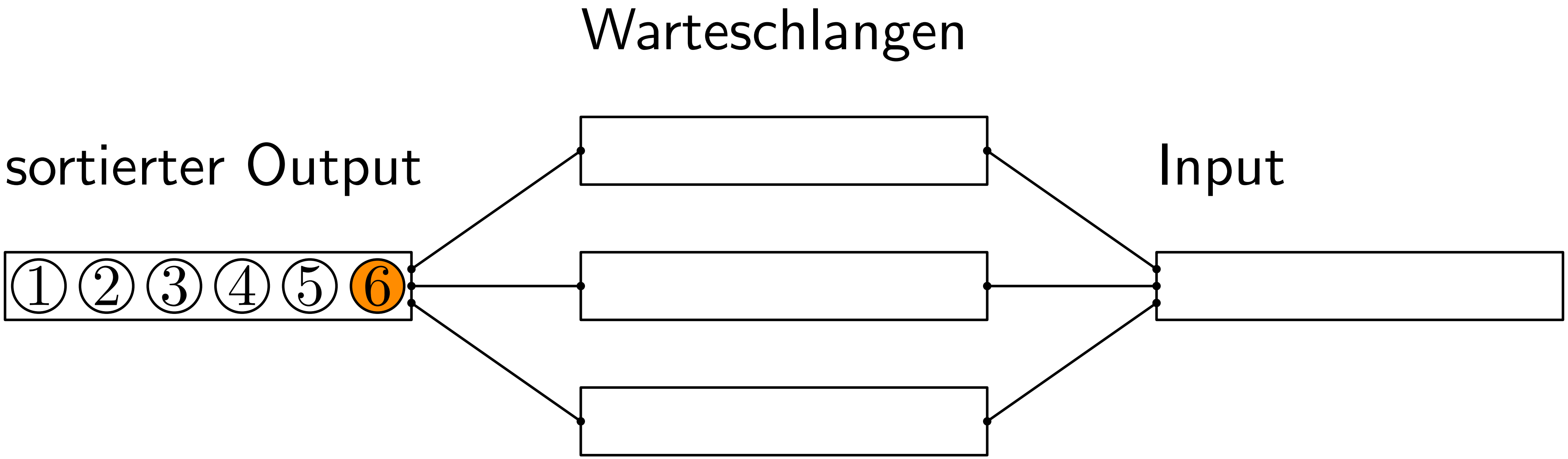
Sortieren mit Warteschlangen



Sortieren mit Warteschlangen



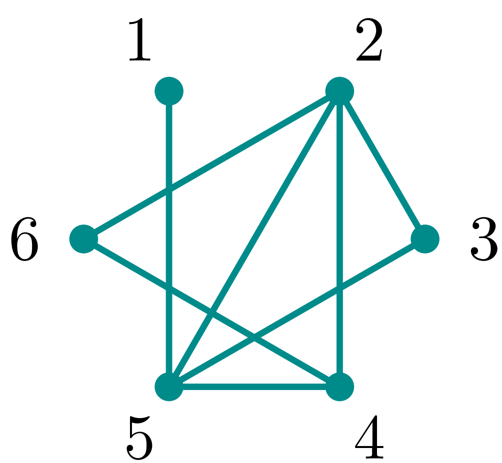
Sortieren mit Warteschlangen



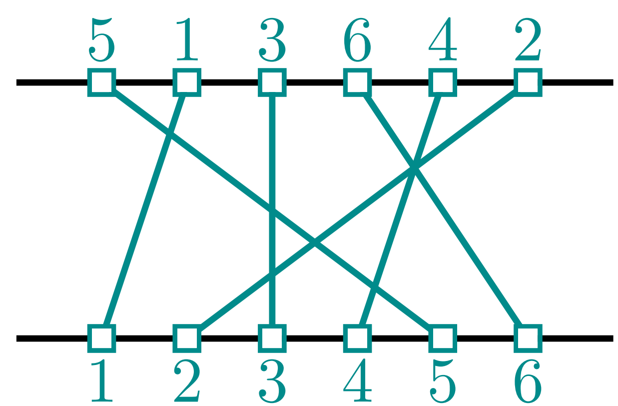
Sortieren mit Warteschlangen

sortierter Output | Warteschlangen | Input

①②③④⑤⑥ ——— ??? ——— ⑤①③⑥④②



Inversionsgraph



Matching Repräsentation

Eingabe : Permutationsgraph $G = (V, E)$.
Ausgabe : Knotenfärbung COLOR und Clique C.

Bestimme Permutation π mit $G \cong G[\pi]$;
für $j \leftarrow 1$ **bis** n **tue**
    $i \leftarrow$ kleinster Index einer zulässigen Warteschlange;
    $\mathrm{COLOR}(\pi_j) \leftarrow i$; d.h. $\mathrm{LAST}(i) < \pi_j$
    $\mathrm{PRED}(\pi_j) \leftarrow \mathrm{LAST}(i-1)$;
    $\mathrm{LAST}(i) \leftarrow \pi_j$; $\mathrm{LAST}(0) \leftarrow \mathrm{null}$
Ende
$C \leftarrow \emptyset$; $k \leftarrow \max\{i\colon\ \mathrm{LAST}(i)$ existiert$\}$; $j \leftarrow \mathrm{LAST}(k)$;
solange $j \neq \mathrm{null}$ **tue**
    $C \leftarrow C + j$;
    $j \leftarrow \mathrm{PRED}(j)$;
Ende
Gebe COLOR und C **aus**;

Algorithmus 9 : Bestimmung von $\chi(G)$ und $\omega(G)$

Eigenschaft V: G ist ein Vergleichbarkeitsgraph.

Eigenschaft $\overline{V}$: $\overline{G}$ ist ein Vergleichbarkeitsgraph.

Eigenschaft C: G ist chordal.

Eigenschaft $\overline{C}$: $\overline{G}$ ist chordal.

V	$\overline{V}$	C	$\overline{C}$	Graphenklasse	
✓				Vergleichbarkeitsgraphen	Kap.4
		✓		chordale Graphen	Kap.3
	✓	✓		Intervallgraphen	Kap.7
		✓	✓	Split-Graphen	Kap.5
✓	✓			Permutationsgraphen	Kap.6
✓		✓		cycle-free partial orders	???

Satz 7.1.
Für ungerichtete Graphen $G = (V, E)$ sind äquivalent:

(i) G ist ein Intervallgraph.

(i.v) G ist chordal und $\overline{G}$ ist Vergleichbarkeitsgraph.

(ii) $C_4 \not\subseteq_{\mathrm{ind}} G$ und $\overline{G}$ ist Vergleichbarkeitsgraph.

(iii) Es existiert eine Ordnung $A_1, \ldots, A_x$ der inklusionsmaximalen Cliquen von G sodass

$\forall v \in V$ gilt $\{i \mid v \in A_i\}$ ist Intervall von $\{1, \ldots, x\}$.

Eingabe : ungerichteter Graph $G = (V, E)$.
Ausgabe : JA wenn G Intervallgraph, NEIN sonst.

Bestimme σ_1 mit LexBFS;
Bestimme σ_2 mit LexBFS mit tie breaker σ_1;
Bestimme σ_3 mit LexBFS mit tie breaker σ_2;
Bestimme σ_4 mit LexBFS mit tie breaker σ_3;
Gebe aus ob σ_4 Intervalle induziert;

Algorithmus 10 : Erkennung von Intervallgraphen