

Aufgabe 20.

- (a) Der Algorithmus durchläuft das noch nicht sortierte Array und tauscht jeweils benachbarte Paare, sodass $a[i + 1] \geq a[i]$. Dieser Vorgang wird so lange wiederholt, bis keine Arrayeinträge mehr vertauscht worden sind und somit alle Einträge in aufsteigender Folge sortiert sind.
- (b) Da die `run`-Variable pro Durchlauf inkrementiert wird, erreicht sie irgendwann den Wert `a.length`. Da nun die innere Schleife nicht mehr durchlaufen wird, bleibt auch die `vertauscht`-Variable zwangsweise `false` und der Algorithmus terminiert.
- (c) Der worst-case ist $\mathcal{O}(n^2)$, da die äußere Schleife aufgrund der in (b) genannten Inkrementierung höchstens n -mal und die innere Schleife ebenso maximal n -mal durchlaufen wird. Dabei sei anzumerken, dass die innere Schleife pro Durchlauf der äußeren Schleife einmal weniger durchlaufen wird und somit die eigentliche Laufzeit $\frac{1}{2}n^2$ lautet, was in der \mathcal{O} -Notation auf die selbe Menge deutet.
Der Algorithmus unternimmt maximal $\frac{n \cdot (n-1)}{2}$ -Tauschoperationen.