

# Sortieralgorithmen

## Zur sortierendes Feld

Ausschneiden und entweder direkt benutzen oder auf der Rückseite eigene Zahlen draufschreiben.

1	2	3	4	4	5	6	7	9	9
---	---	---	---	---	---	---	---	---	---

## Sortiertes Feld

--	--	--	--	--	--	--	--	--	--

## Algorithmen

### Bubblesort – $O(n^2)$

Der Algorithmus vergleicht der Reihe nach zwei benachbarte Elemente und vertauscht sie, falls sie in der falschen Reihenfolge vorliegen. Dieser Vorgang wird solange wiederholt, bis keine Vertauschungen mehr nötig sind.

### Quicksort (Divide-and-Conquer) – instabil $O(n \log n)$

Die Liste wird „vorsortiert“, geteilt und dann jede Teilliste weitersortiert.

*Wähle ein Pivotelement und zerlege die Liste in zwei Teillisten. Suche ein Element von unten, das größer (oder gleichgroß) als das Pivotelement. Entsprechend suche von oben ein kleineres als das Pivotelement. Die beiden Elemente werden dann getauscht. Der Vorgang wird fortgesetzt, bis sich die untere und obere Suche treffen.*

### Insertionsort - $O(n)$ bis $O(n^2)$

Der Algorithmus entnimmt der unsortierten Eingabemenge ein beliebiges (z.B. das erste) Element und fügt es an richtiger Stelle in die (anfangs leere) Ausgabemenge ein.

### Selectionsort - $O(n^2)$

Das kleinste Element wird an die vorderste Stelle kopiert.  
(Alle Spielkarten sind bereits auf der Hand)

### Mergesort (Divide-and-Conquer) – stabil $O(n \log n)$

Im Vergleich zum Quicksort ist hier der verschmelzen (to merge) aufwendiger. Die Liste wird in kleinere zerlegt, sortiert und wieder zusammengefügt.

Kein Algorithmus ist schneller als  $O(n \log n)$ . Lineare sind nicht vergleichsbasiert.

### Bucket sort – $O(n+k)$

*Entspricht dem Sortieren von Post in die jeweiligen Postfächer*

(nicht zu verwechseln mit countsort bei dem aufgrund der Häufigkeit sortiert wird)