

## BubbleSort

BubbleSort verfährt nach folgender Methode:

Man durchläuft die Liste  $a[1] \dots a[N]$  der Datensätze und betrachtet dabei je zwei benachbarte Elemente  $a[i]$  und  $a[i+1]$ . Ist  $a[i].key > a[i+1].key$ , so vertauscht man  $a[i]$  und  $a[i+1]$ . Das Durchlaufen des Arrays wird solange wiederholt, bis keine Elemente mehr vertauscht wurden, das Array also sortiert ist.

Das Verfahren hat seinen Namen, da größere Elemente wie Luftblasen nach oben/rechts aufsteigen.

Bsp.:

Feld B =                    15,2,43,17,4,8,47

Nach 1. Durchlauf:    2,15,17,4,8,43,47

Nach 2. Durchlauf:    2,15,4,8,17,43,47

Nach 3. Durchlauf:    2,4,8,15,17,43,47

Keine Vertauschungen mehr bei 4. Durchlauf → Feld Sortiert

Analyse:

- WorstCase: Feld ist nach absteigenden Schlüsseln sortiert!

→  $O(n^2)$

Warum: Ist das Feld absteigend sortiert, so muss das erste Element  $n$  Schritte bewegt werden, das zweite Element  $(n-1)$ , das  $n$ . Element 1 Schritt bewegt werden.

→ Es sind  $[n + (n-1) + (n-2) + \dots + 1] \cdot c = [c \cdot n \cdot (n-1)] / 2$  Schritte

notwendig. ( $c = \text{const.}$ )

→  $[c \cdot n \cdot (n-1)] / 2 \in O(n^2)$

- AverageCase  $O(n^2)$
- BestCase  $O(n)$
- Iterativer Algorithmus
- In-Place