

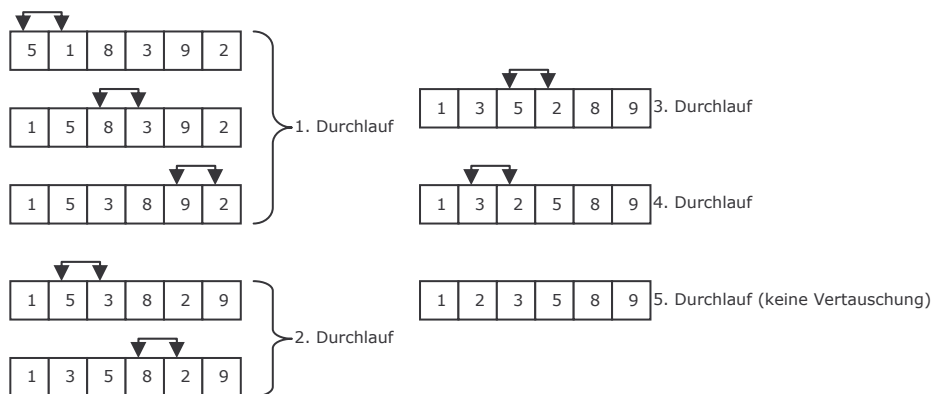


# Übersichtsblatt: Bubble Sort

## Idee:

Der Bubble Sort ist ein Sortieralgorithmus, der auf folgendem System basiert. Man geht ein übergebenes Array immer wieder durch und vertauscht benachbarte Elemente, so lange bis alle Elemente in der Sortierreihenfolge stehen, sprich bis es keine Vertauschungen mehr gibt.

## Beispiel:



## Pseudocode:

Eingabe: Feld A mit n Komponenten  
Ergebnis: Sortiertes Feld A' mit  $A'[0] \leq A'[1] \leq \dots \leq A'[n-1]$

```
int bubbleSort(A, n) {
    do {
        count = 0;          /*Bleibt bestehen wenn nichts Vertauscht wird!*/
        for(i=0; i < n-1; i++) { /*Gehe alle Elemente durch!*/
            if(A[i] > A[i+1]) { /*Ist das Linke größer als das Rechte?*/
                swap(A[i], A[i+1]); /*Wenn ja, dann vertusche Sie!*/
                count = 1; /*Sagt uns, dass es eine Vertauschung gab!*/
            }
        }
    } while(count==1); /*Solange es Eine Vertauschung gab!*/
    return A;
}
```

## Analyse:

Auf Grund der verschachtelten Schleife liegt die Laufzeit des Bubble Sort im Bereich  $O(n^2)$ . Dennoch hat er gegenüber anderen einfachen Sortierverfahren einen erheblichen Vorteil:

Die Daten die vertauscht werden sollen, liegen im Speicher eng beieinander, was es sehr wahrscheinlich macht, dass die benötigten Informationen bereits im CPU-Cache vorhanden sind.

Für vorsortierte Daten kommt der Bubble Sort daher in der Regel mit sehr wenigen Vertauschungen aus.

