

### 4.3 Bubblesort

Das Sortieren durch Aufsteigen (engl. BubbleSort, "Blasensortierung") bezeichnet einen einfachen Sortieralgorithmus, der eine Reihe zufällig angeordneter Elemente der Größe nach ordnet.

Der Sortieralgorithmus fällt sämtliche Entscheidungen allein auf Basis des Größenvergleichs je zweier Elemente.

**Bubblesort** gehört zur Klasse der **In-place-Verfahren**, was bedeutet, daß der Algorithmus zum Sortieren keinen zusätzlichen Speicher außer den lokalen Laufvariablen benötigt.

#### Prinzip

*Bubblesort* 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. Mehrere Durchläufe sind in der Regel erforderlich.

Je nachdem, ob auf- oder absteigend sortiert wird, steigen die größeren oder kleineren Element wie Blasen immer weiter nach hinten, d.h. an das Ende der Reihe.

#### Pseudocode

```

prozedur bubbleSort( A : Liste sortierbarer Elemente )
  n := Laenge( A )
  wiederhole
    vertauscht := falsch
    fuer jedes i von 0 bis n - 2 wiederhole
      falls A[ i ] > A[ i + 1 ] dann
        // vertausche( A[ i ], A[ i + 1 ] )
        t := A[i]
        A[i] := A[i+1]
        A[i+1] := t
        vertauscht := wahr
      falls ende
    fuer ende
    n := n - 1
  solange vertauscht und n > 1
prozedur ende

```

Die äußerste Schleife durchläuft die zu sortierenden Daten, bis keine Vertauschungen mehr nötig sind.

#### Analyse

*Bubblesort* hat die Laufzeit  $\mathcal{O}(n^2)$  für Listen der Länge  $n$ . Im Falle einer umgekehrt sortierten Liste werden maximal  $n \cdot (n-1)/2$  viele Vertauschungen ausgeführt.

**Algorithmus Bubblesort in C - Notation:**

```

#include <stdio.h>
void bubblesort(int v[], int n) { /* Sortiert aufsteigend n Elemente des Arrays v */
    char b = 0;
    int i = 0;

    do { b = 0;          // Ruecksetzen Austauschmerkmal

        for (i = 0; i < n-1; i++) {

            // ggf. Korrektur der Ordnung zweier, aufeinanderfolgender Werte
            if (v[i] > v[i + 1]) {

                // Bubble-Schritt (Vertauschung)
                int t = v[i];
                v[i] = v[i + 1];
                v[i + 1] = t;
                b = 1;      // Austausch erfolgte
            }
        }
        n--;
    } while(b && n>1); // wurde ausgetauscht und noch
                    // mehr als 1 Wert in Liste ?
}

void main() { // Kleiner Test: Lasse es bubbeln !
    int i = 0;
    int values[] = {25, 13, 19, 31, 11, 99, 21, 17};
    int count = sizeof(values) / sizeof(*values);

    // Ausgabe der Werte (vorher)
    printf("Werte vor dem Sortieren:\n");
    for (i = 0; i < count; i++)
        printf("%d ", values[i]);
    printf("\n");
    // Sortiere alle (count!) Elemente von values
    bubblesort(values, count);

    // Ausgabe der Werte (nachher)
    printf("\n\nWerte nach dem Sortieren:\n");
    for (i = 0; i < count; i++)
        printf("%d ", values[i]);
    printf("\n");
    getchar();
}

/*
Werte vor dem Sortieren:
25 13 19 31 11 99 21 17

Werte nach dem Sortieren:
11 13 17 19 21 25 31 99
*/

```