

# Sortieralgorithmen

Problem:

- Es sind  $n$  Objekte zu sortieren.
- Wie hoch ist der Aufwand der Sortierung ?

## 1. Elementare Sortierverfahren

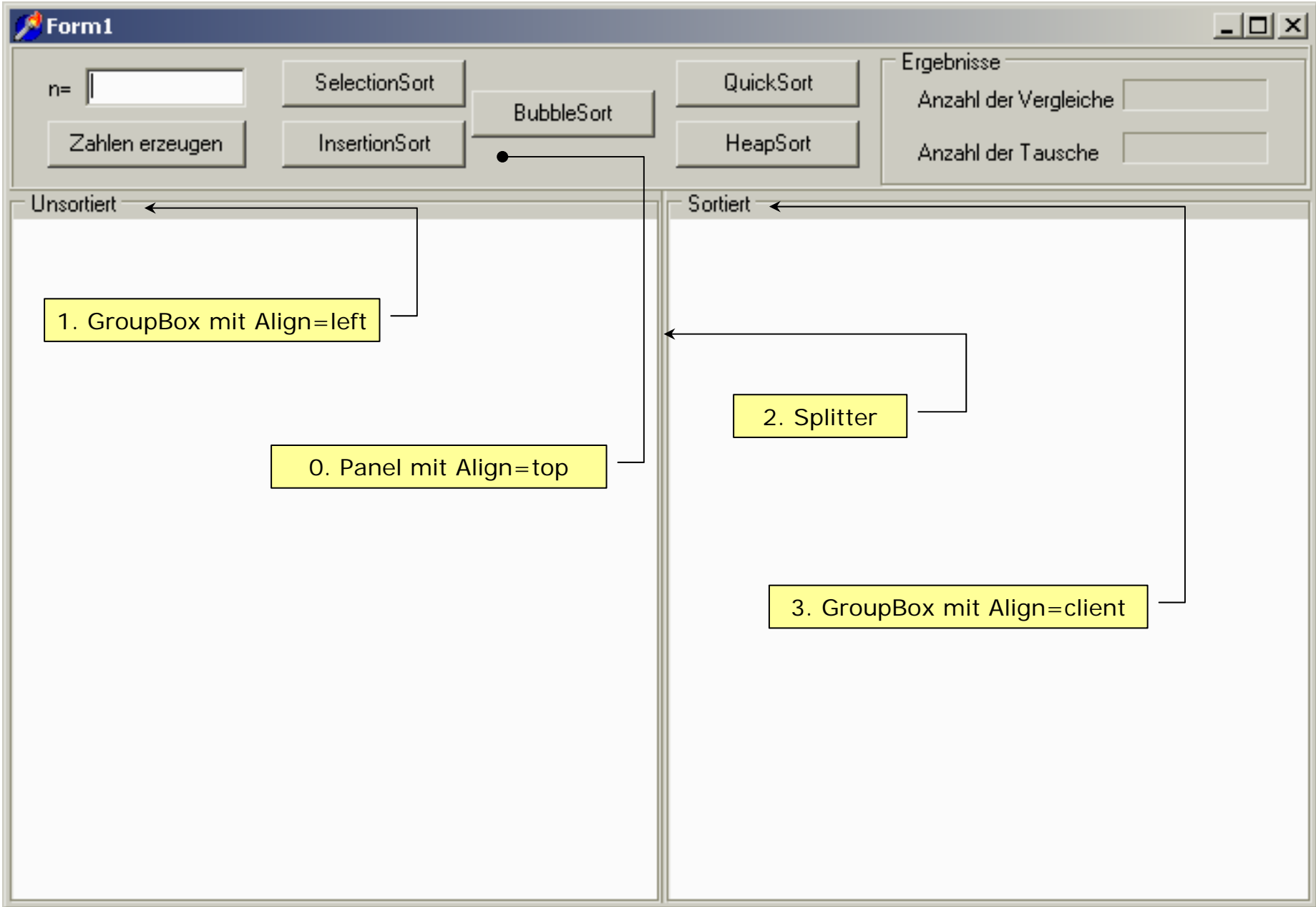
a) Sortieren durch Auswahl → SelectionSort

b) Sortieren durch Einfügen → InsertionSort

c) Sortieren durch Austausch → BubbleSort

## 2. QuickSort

## 3. HeapSort

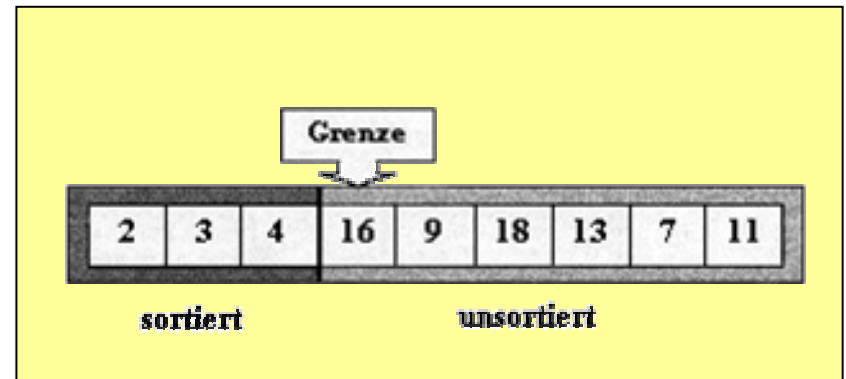


## Elementare Sortierverfahren

Bei den elementaren Sortierverfahren **SelectionSort**, **InsertionSort** und **BubbleSort** handelt es sich um interne und direkte Sortierverfahren, d.h. das Sortieren geschieht direkt in der Feldstruktur und intern im Speicher des Rechners. Die Daten müssen nicht in zusätzliche Strukturen übertragen oder auf einem Datenträger extern gespeichert werden.

Bei jedem der Sortierverfahren gibt es einen sortierten und einen unsortierten Teil, welche durch eine Grenze voneinander getrennt sind. Dabei ist der sortierte Teil anfangs entweder leer oder besteht aus einem Element.

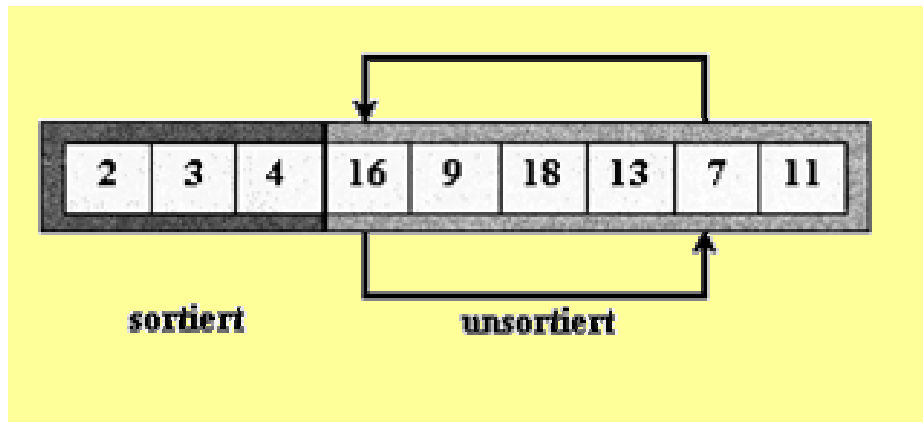
Im Laufe des Sortierens wandert die Grenze kontinuierlich, d.h. der sortierte Teil wird solange größer, bis es keinen unsortierten Teil mehr gibt. Dann ist der Sortiervorgang beendet.



## Sortieren durch Auswahl (SelectionSort)

### Verfahren:

- Der sortierte Teil ist zu Anfang leer.
- Bei jedem Durchgang wird aus dem unsortierten Teil das kleinste Element herausgesucht und mit dem ersten Element des unsortierten Teils vertauscht.
- Die Grenze wird um eine Position verschoben, so dass der sortierte Teil jetzt das zuvor als kleinstes Element eingestufte Element enthält.
- Dieser Vorgang wiederholt sich, bis die Feldstruktur nur noch aus dem sortierten Teil besteht.

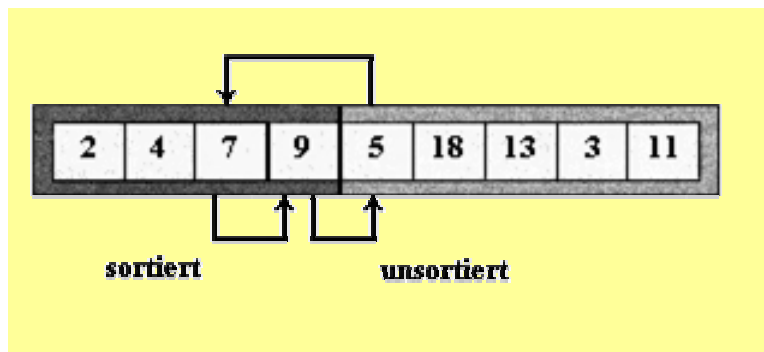


Entwickle für diesen Algorithmus ein Struktogramm und implementiere ihn in dein Programm.

## Sortieren durch Einfügen (Insertion Sort)

### Verfahren:

- Der sortierte Teil besteht zu Anfang aus einem Element.
- Bei jedem Durchgang wird aus dem unsortierten Teil das erste Element genommen und an der passenden Stelle im bereits sortierten Teil eingefügt. *(Der sortierte Teil wird von hinten nach vorne durchlaufen; jedes verglichene Element wird um eine Position nach hinten verlagert, bis das neue Element an der richtigen Stelle steht.)*
- Die Grenze wird um eine Position verschoben.
- Dieser Vorgang wiederholt sich, bis die Feldstruktur nur noch aus dem sortierten Teil besteht.

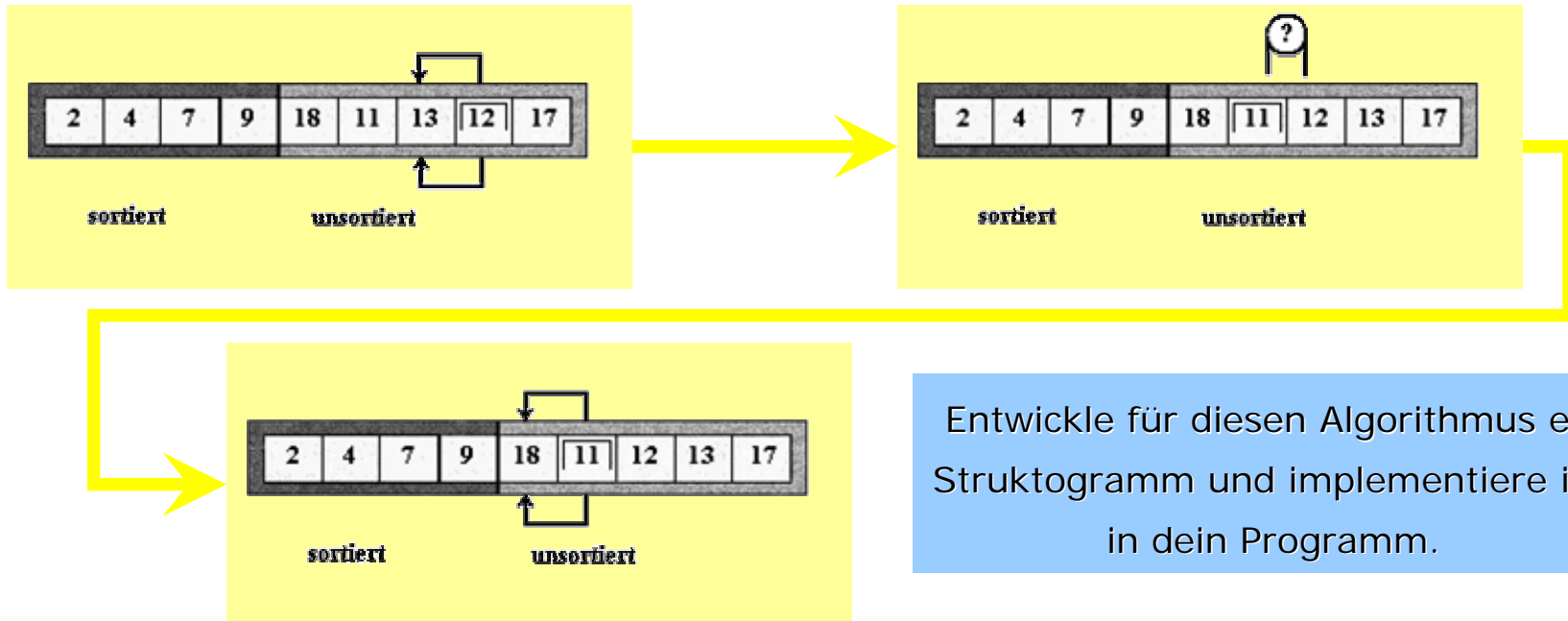


Entwickle für diesen Algorithmus ein Struktogramm und implementiere ihn in dein Programm.

# Sortieren durch Austausch (Bubble Sort)

## Verfahren:

- Der sortierte Teil besteht zu Anfang leer.
- Bei jedem Durchgang wird aus dem unsortierten Teil das letzte Element gewählt und solange mit dem jeweils nächsten Element vertauscht, bis es auf ein kleineres Element trifft.
- Das bis dahin gewanderte Element bleibt nun an der Position, und das neue, kleinere Element wird gewählt.
- Dieser Vorgang wiederholt sich solange, bis der unsortierte Teil einmal durchlaufen ist.
- Die Grenze wird um eine Position verschoben.
- Der Vorgang wiederholt sich, bis die Feldstruktur nur noch aus dem sortierten Teil besteht.



Entwickle für diesen Algorithmus ein Struktogramm und implementiere ihn in dein Programm.