

Umgangssprachliche Formulierungen und Pseudocodes

Umgangssprachlich:

Bubblesort:

Fang mit den ersten beiden Elementen an. Vergleiche diese miteinander und tausche, wenn das linke größer ist als das rechte. Mache jeweils eine Position weiter rechts weiter, bis du beim letzten zu vergleichenden Element angekommen bist (das ist anfangs das ganz rechte). (Bei einem Durchlauf wandert das größte Element wie eine Blase durch).

Wiederhole den beschriebenen Vorgang und zwar pro Durchlauf einmal weniger und so lange, bis es mal keinen Tausch gab oder man bei 0 zu vergleichenden Elementen angelangt ist.

Insertionsort:

Anfangs hat man eine sortierte Liste der Länge eins. Nimm jeweils das erste nicht sortierte Element, durchlaufe die sortierte Liste beginnend von rechts nach links, vergleiche es mit dem jeweils betrachteten Element und verschiebe dies um eine Position nach rechts, solange du noch nicht durch die ganze Liste durch bist und das jeweils betrachtete Element größer ist. Füge nun das einzufügende Element ein. Wiederhole diesen Vorgang bis zum n-ten Element.

Mergesort:

Man halbiert eine Liste so lange rekursiv, bis alle Teillisten die Länge 1 haben. Diese Teillisten fügt man nun wieder zusammen, indem beim Abarbeiten zweier Teillisten (jeweils von links nach rechts) jeweils das kleinere Element der beiden Listen in einer neuen Liste speichert. Auf diese Weise sortiert man die Teillisten, bis man eine sortierte Gesamtliste hat.

Selectionsort:

Suche die gesamte Liste vom anfänglich ersten Element bis zum Ende durch, finde die Position des Minimums (indem du die jeweils aktuell gefundene Minimumposition überschreibst) und tausche es auf die vorderste nicht-sortierte Position. Führe diesen Vorgang jeweils ab einer Position weiter durch, bis die gesamte Liste sortiert ist.

Quicksort:

Bestimme mit irgendeinem Algorithmus ein Pivotelement, und sortiere die die Liste in zwei Teillisten so um, dass links nur Elemente kleiner als das Pivotelement und rechts nur Elemente größer als das Pivotelement vorkommen. Sortiere die Teillisten jeweils rekursiv, solange die Teillistenlänge größer 1 ist.

Pseudocodes:**Bubblesort:**

```

Methode bubbleSort für eine Liste
  setze Zähler auf Listenlänge-1
  wiederhole
    setze vertauschtMerker auf falsch
    wiederhole für jedes i von 1 bis Zähler
      wenn i-tes Element größer als i+1-tes
        vertausche die beiden Elemente
        vertauschtMerker auf wahr setzen
    *wenn
    *wdh mit fester Anzahl
    Wert von Zähler minus eins (letztes ist fest)
  *wiederhole bis Zähler=0 || vertauschtMerker==falsch
Ende Methode

```

Insertionsort:

```

Methode insertionSort für eine Liste
  Wiederhole für Stellen 2 bis n
    einzusortierendenWert an der Stelle rausnehmen
    eins links davon starten mit Listenposition
    Solange einzusort.Wert < aktuellerListenwert und
    man noch nicht am Anfang ist
      aktuellenListenwert eins nach rechts schieben
      eins weiter links vergleichen
    *ende solange
    einzusortierendenWert einfügen
  *ende Wdh.
Ende Methode

```

Mergesort:

```

mergesort(liste)
  falls (listengröße > 1) dann
    halbiere liste in lListe, rListe
    mache mergesort von lListe
    mache mergesort von rListe
    merge von lListe und rListe zurückgeben
  *ende falls
*ende mergesort
merge(lListe, rListe)
  solange (lListe und rListe nicht leer)
    falls (1.El lListe <= 1.El rListe)
      1.El von lListe raus und hinten in Liste rein
    sonst
      1.El von rListe raus und hinten in Liste rein
  *solange
  solange (lListe nicht leer)
    1.El von lListe raus und hinten in Liste rein
  *solange
*merge

```

Selectionsort:

```

Methode SelectionSort für eine Liste
  linkerZeiger am Anfang ganz links
  wiederhole solange linkerZeiger < EndeListe
    setze Minipos auf linkerZeiger
    für jedes i ab nach linkerZeiger bis EndeListe
      falls Element i kleiner Element Minpos
        setze Minpos auf i
    Ende falls
  Ende für
  Tausche Element Minpos mit Element linkerZeiger
  linkerZeiger eins nach rechts setzen
Ende wiederhole
Ende Methode

```

Quicksort:

```

Sortiere eine Liste mit gegebenem Unter-, und Obergrenzenindex
  wähle als Pivotelement Element in Mitte der Liste
  setze den linken Marker auf den Untergrenzenindex
  setze den rechten Marker auf den Obergrenzenindex
  solange linker Marker noch nicht auf rechter Marker mache:
    wandere durch linken Listenteil solange Ele kleiner PivEl
    wandere durch rechten Listenteil solange Ele größer PivEl
    wenn linker Marker kleinergleich rechter Marker
      tausche die aktuellen Elemente
      wandere jeweils eins weiter
  *wenn
  *solange
  wenn die linke Teilliste länger als 1 ist,
    mache Quicksort von linker Teilliste
    (also von Untergrenze bis zum rechten Zeiger)
  *wenn
  wenn die rechte Teilliste länger als 1 ist,
    mache Quicksort von rechter Teilliste
    (also vom linken Zeiger bis zur Obergrenze)
  *wenn
*sortiere

```