Sortierverfahren

1. Einführung

Ein Problem beim Speichern großer Datenmengen besteht darin, ein gewünschtes Datum wieder zu finden. Zum schnellen Auffinden spielt hierbei eine wesentliche Rolle, wie die Daten abgelegt werden. Würde man z.B. die Telefonnummern eines Fernmeldebezirks einfach in der Reihenfolge des Einrichtens der Teilnehmeranschlüsse im Telefonbuch aufführen, hätte man seine liebe Not die Nummer eines Teilnehmers ausfindig zu machen. Im schlimmsten Fall müssten alle Teilnehmer durchsucht werden. Liegt jedoch z.B. eine alphabetische Ordnung der Namen vor, so wird man sehr viel schneller fündig, bzw. kann man rasch feststellen, ob ein Teilnehmer überhaupt aufgeführt ist.
Dem Sortieren von Daten kommt also eine große Bedeutung zu. Dies gilt auch für die Datenrecherche mit Unterstützung von Computern. Nicht umsonst gibt es über hundert Algorithmen zum Sortieren von Daten.
Für die Güte eines Sortieralgorithmus sind folgende Kriterien ausschlaggebend:

- Wie schnell sortiert der Algorithmus durchschnittlich die ihm übergebenen Daten?
- Wie schnell ist er im günstigsten bzw. im ungünstigsten Fall?
- Verhält sich der Algorithmus natürlich, d.h. bei gut sortierten Listen hat er nur wenig und bei unsortierten Listen hat er viel zu tun?

Grundsätzlich gibt es drei Methoden zum Sortieren:

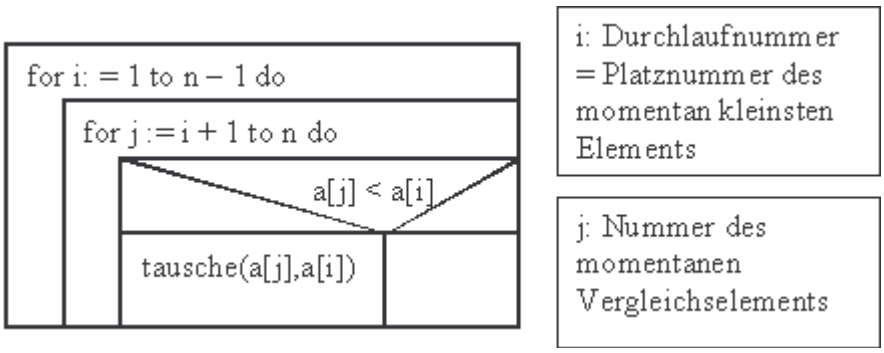
- Sortieren durch Auswahl
- Sortieren durch Einfügen
- Sortieren durch Austausch.

Einige einfache Verfahren sollen im Folgenden betrachtet werden. Die zu sortierenden Daten sollen der Übersichtlichkeit wegen ganze Zahlen sein. Das Sortieren dieser Zahlen erfolgt auf eindimensionalen Feldern.

2. Sortieren durch Auswahl – Selection Sort

Prinzip: Aus einer Menge wird das kleinste Element ausgewählt und mit dem ersten Element vertauscht. Diese Vorgänge werden wiederholt, wobei die Anzahl der Elemente bei jeder Wiederholung von vorn um eins verringert wird. Zuletzt werden nur noch zwei Elemente überprüft und gegebenenfalls vertauscht.

Struktogramm:



Aufgaben:

1. Testen Sie den Sortieralgorithmus mit Bleistift und Papier an folgendem Beispiel!
   6 4 10 8 8 2 1
2. Verfolgen Sie den Ablauf des Algorithmus mit Hilfe des Programms `sortiere.com`!
3. Veranschaulichen Sie sich den Sortiervorgang grafisch mit Hilfe des Programms `sortieren.exe`!
4. Vervollständigen Sie das bereits vorgefertigte Programm!

4.1 Ergänzen Sie zunächst die Deklaration für das zu sortierende Zahlenfeld!

4.2 Schreiben Sie eine Prozedur, die 10 zufällige ganze Zahlen erzeugt und diese in ein Feld einliest!

4.3 Lassen Sie das unsortierte Feld ausgeben!

4.4 Schreiben Sie zwei Prozeduren `tausche` und `ausgabe` zum Vertauschen von zwei Zahlen bzw. zum Ausgeben des sortierten Feldes!

4.5 Erstellen Sie die Prozedur zum Sortieren des Feldes durch Auswahl! Nutzen Sie dazu das vorgegebene Struktogramm! Speichern Sie die Anzahl der notwendigen Tauschoperationen und die Anzahl der durchzuführenden Vergleiche in den Variablen `tausch` und `vergleich`!

4.6 Testen Sie das Programm mit mehreren Beispielen!

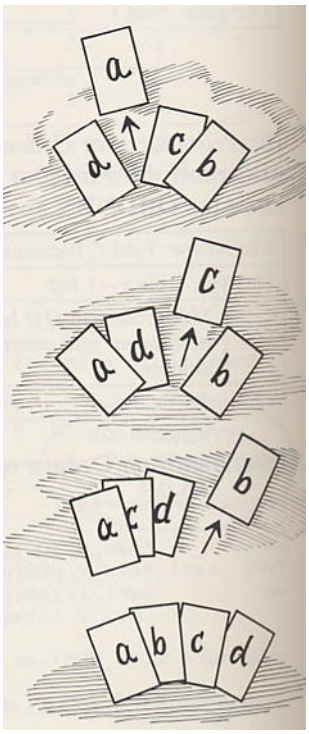
4.7 Ergänzen Sie das Programm dahingehend, dass es auch möglich ist die beiden Sonderfälle
- die 10 Zahlen sind bereits sortiert und
- die 10 Zahlen sind absteigend sortiert (also von 10 bis 1)

zu testen! Ermitteln Sie die maximale Anzahl der Tauschoperationen im *worst case*!

4.8 Ermitteln Sie die Anzahl der Vergleichsoperationen bei 10 Feldelementen! Bestätigen Sie die Anzahl der Vergleiche rechnerisch, indem Sie die Anzahl der Vergleiche für n Feldelemente berechnen (schwierig!)!

3. Sortieren durch Einfügen – Insertion Sort

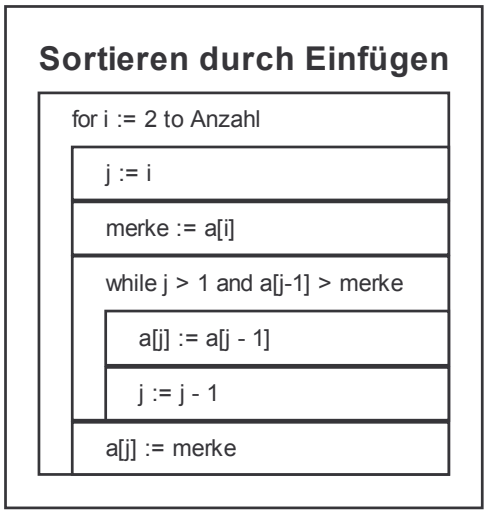
Dieses Verfahren wird oft beim Sortieren von Kartenstapeln, Namenslisten usw. angewendet. Der oberste Wert wird aus dem Stapel genommen und in den bereits sortierten Teil eingefügt (siehe Abbildung).



Prinzip: Man wählt ein Element (nacheinander vom zweiten bis zum letzten) und fügt es jeweils an die richtige Stelle der davor stehenden geordneten Teilmenge ein durch:

- Speichern des Elementes in einer Hilfsvariablen
- Verschieben aller größeren Elemente nach oben
- Wiedereinfügen des Elementes an der freien Stelle

Struktogramm:



Aufgaben:

1. Testen Sie den Sortieralgorithmus mit Bleistift und Papier an folgendem Beispiel!
   6 4 10 8 8 2 1
2. Verfolgen Sie den Ablauf des Algorithmus mit Hilfe des Programms `sortiere.com`!
3. Veranschaulichen Sie sich den Sortiervorgang grafisch mit Hilfe des Programms `sortieren.exe`!
4. Ergänzen Sie in Ihrem Programm das Sortieren durch direktes Einfügen!

4.1 Erstellen Sie die Prozedur zum Sortieren des Feldes durch Einfügen! Nutzen Sie dazu das vorgegebene Struktogramm! Speichern Sie die Anzahl der durchzuführenden Vergleiche in der Variablen `vergleich`!

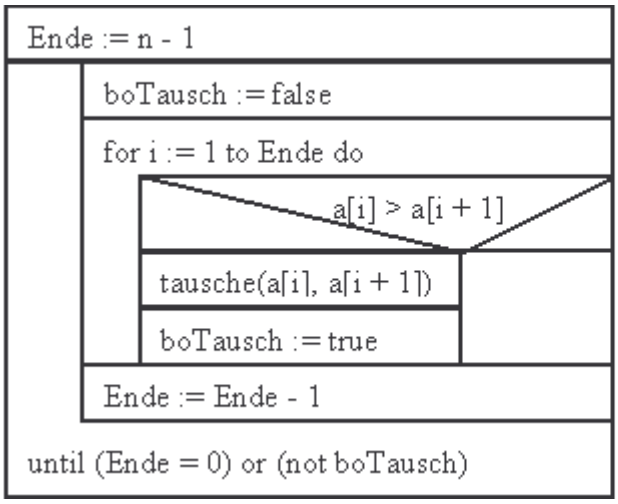
4.2 Testen Sie das Programm mit mehreren Beispielen!

4.3 Vergleichen Sie die Anzahl der durchzuführenden Vergleiche mit denen beim Sortieren durch Auswahl! Was stellen Sie fest?

4. Sortieren durch Austausch – Bubble Sort

Prinzip: Dieses Verfahren ist ähnlich dem Auswahlverfahren. Es werden jedoch nebeneinander stehende Elemente verglichen. Ist das nachfolgende Element kleiner, so erfolgt ein Austausch der betrachteten Vergleichs-Elemente. Danach rückt man eine Position weiter und vergleicht diese Elemente. Die größte Zahl "steigt" bei jedem Durchlauf nach rechts ähnlich den Luftblasen in einem Glas Wasser. Daher spricht man auch von *Bubblesort*.

Struktogramm:



Ende: Rechtes Ende des noch unsortierten Listenteils

boTausch: Stellt fest ob während eines Durchlaufs ein Austausch stattfand

i: Position des aktuellen Elements während eines Durchlaufs

Aufgaben:

1. Testen Sie den Sortieralgorithmus mit Bleistift und Papier an folgendem Beispiel!
   6 4 10 8 8 2 1
2. Verfolgen Sie den Ablauf des Algorithmus mit Hilfe des Programms `sortiere.com`!
3. Veranschaulichen Sie sich den Sortiervorgang grafisch mit Hilfe des Programms `sortieren.exe`!
4. Ergänzen Sie in Ihrem Programm das Sortieren durch Austausch!

4.1 Erstellen Sie die Prozedur zum Sortieren des Feldes durch Austauschen! Nutzen Sie dazu das vorgegebene Struktogramm! Speichern Sie die Anzahl der notwendigen Tauschoperationen und die Anzahl der durchzuführenden Vergleiche in den Variablen `tausch` und `vergleich`!

4.2 Testen Sie das Programm mit mehreren Beispielen!

4.3 Vergleichen Sie die Anzahlen der durchzuführenden Vergleiche und Tauschoperationen mit denen der anderen Sortierverfahren! Was stellen Sie fest?