

## 8

## Algorithmen auf Datenstrukturen

## 8.1 Permutationen

## Beispiel: Fließbandarbeit

## Übungen

Ist die Reihenfolge der auszuführenden Arbeiten nicht fest vorgegeben und kann damit parallel erfolgen, so ist der Engpass derjenige Auftrag, der die längste Arbeitszeit erfordert. Mathematisch bedeutet dies

$$\max_{i=1,\dots,n} a_{ip_i} \rightarrow \text{Minimum}. \quad (8.1.2)$$

Das Problem ist mit dem gleichen Programm lösbar, wenn die Werte negativ eingegeben werden. Zu beachten ist, dass der Startwert Max nicht Null, sondern z. B. -9999 gesetzt wird.

Wir benutzen die gleichen Testdaten mit der Bedeutung, dass ein Positionswert den Arbeitsaufwand darstellt.

	A	B	C	D	E	F	G
1	-7	-5	-11	-13	-14	-9	
2	-3	-6	-15	-6	-3	-11	
3	-10	-6	-3	-1	-2	-7	
4	-4	-7	-5	-1	-6	-8	
5	-5	-7	-8	-3	-10	-6	
6	-4	-5	-3	-6	-12	-9	
7							
8	2	1	5	4	6	3	
9	-5	-3	-2	-1	-6	-3	-20

Abb. 8-1 Testdatenauswertung

Das Ergebnis liefert einen Maximalwert von 20, wobei der größte Aufwand der Arbeitswert  $a_{5,6}$  ist.

## 8.2 Nichtlineare Regression

Regressionsanalysen befassen sich mit dem funktionalen Zusammenhängen zwischen einer oder mehreren variablen Einflussgrößen und einer zufälligen Variablen als Zielgröße. Das Prinzip einer linearen oder nicht linearen Regressionsanalyse beruht darauf, dass eine Regressionsfunktion gefunden wird, die die Punktmenge mit möglichst geringen Abweichungen wiedergibt.

Einer linearen Regressionsanalyse im weitesten Sinne, liegt eine Regressionsfunktion vom Typ

$$f(x) = a \cdot x^2 + b \cdot x + c \quad (8.2.1)$$

zugrunde. Aber auch jede Linearkombination von Funktionen der Form

$$u(x) = a \cdot f(x) + b \cdot g(x) + c \cdot h(x) + \dots \quad (8.2.2)$$

Die Lösung zur Bestimmung der Parameter  $a$ ,  $b$ ,  $c$ , usw. läuft in der Regel immer auf die Bestimmung eines linearen Gleichungssystems hinaus.

Aber auch Regressionsfunktionen, die nicht linear von ihren Parametern abhängen, lassen sich oft durch Umwandlung in eine lineare Form überführen. So wird z. B. aus der Funktion

$$f(x) = a \cdot e^{b \cdot x}, \quad (8.2.3)$$

durch logarithmische Umstellung

$$\ln(f(x)) = \ln(a) + b \cdot x. \quad (8.2.4)$$

Nachdem im Buch das Beispiel einer linearen Regression beschrieben wurde, soll nachfolgend ein Beispiel aus der Vielzahl nicht linearer Regressionsanalysen gezeigt werden.

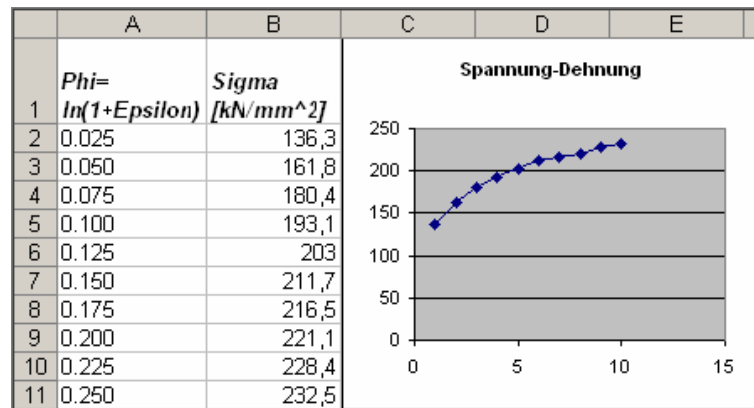
### Beispiel: Spannungs-Dehnungs-Feld

Bei einer Spannungs-Dehnungs-Messung treten in der Regel nichtlineare Zusammenhänge auf.

Gemessen wurden die in Abbildung 8-2 dargestellten Werte. Die Beziehung soll durch die Regressionsfunktion

$$f(x) = a_1 + (a_2 - a_1) \cdot e^{-a_3 \cdot x} \quad (8.2.5)$$

dargestellt werden.



**Abb. 8-2** Spannungs-Dehnungs-Messung

Auch hier gilt vom Ansatz her die Aufgabe, die Zielfunktion

$$g(a,b,c) = \sum (y_i - f(x_i))^2 \quad (8.2.6)$$

zu minimieren. Dies ist iterativ nur näherungsweise lösbar.

Ein typisches Verfahren dazu ist das Newtonsche Näherungsverfahren, auch als Newton-Raphson Methode bekannt. Es ist in der Numerischen Mathematik das Standardverfahren zur Lösung nichtlinearer Gleichungen und Gleichungssysteme.

Wenn man annimmt, die gesuchte Funktion ist nur von zwei Parametern abhängig ( $a_0, a_1$ ), dann kann man sich deren Verlauf als eine Fläche mit „Bergen“ und „Tälern“ vorstellen, in der man zu einer Talsohle gelang, wenn man schrittweise der Richtung des stärksten Abfalls folgt. Diese wird durch den negativen Gradienten von  $g$  angegeben, also durch den Vektor mit den partiellen Ableitungen nach den Parametern

$$-g'(a_1, a_2) = \left( -\frac{\delta g}{\delta a_1}; -\frac{\delta g}{\delta a_2} \right) \quad (8.2.7)$$

Es ergibt sich ein Iterationsverfahren für den gesuchten Vektor

$$\bar{a}_{neu} = \bar{a}_{alt} - Q(\bar{a}_{alt}) \cdot g'(\bar{a}_{alt}) \quad (8.2.8)$$

**Tab. 8-1** Nichtlineare Regression

Einlesen der Daten n, Max, t	
Einlesen der Daten x(i), y(i), i=1, ..., n	
Einlesen des Startvektors a(i), i=1, 2, 3	
Count = 1, 1, Max	
i = 1, 1, n	
$f(x_i) = a_1 + (a_2 - a_1) \cdot e^{-a_3 \cdot x_i}$	
$d_1(x_i) = 1 - e^{-a_3 \cdot x_i}$	
$d_2(x_i) = e^{-a_3 \cdot x_i}$	
$d_3(x_i) = -x_i \cdot (a_2 - a_1) \cdot e^{-a_3 \cdot x_i}$	
$\chi^2 = 0$	
i = 1, 1, n	
$\chi^2 = \chi^2 + (y(x_i) - f(x_i))^2$	
$\chi' = \sqrt{\frac{\chi^2}{n}}$	
Ist $\chi' \leq t$	
Ja	Nein

Ausgabe $\vec{a}$ und Abbruch der Count- Schleife	./.
$ \text{grad}  = 0$	
k = 1, 1, 3	
$d\chi(k) = 0$	
i = 1, 1, n	
$d\chi(k) = d\chi(k) - 2 \cdot (y(x_i) - f(x_i)) \cdot d(x_i, k)$	
$ \text{grad}  =  \text{grad}  + d\chi(k)^2$	
$\chi^2 = \chi^2 - n \cdot \text{Tol}^2$	
k = 1, 1, 3	
$a(k) = a(k) - \left( \chi^2 \cdot \frac{d\chi^2}{ \text{grad} } \right)$	

In der nachfolgenden Prozedur wird der Lösungsvektor bis zur angegebenen Genauigkeit bestimmt und anschließend werden die aus der Regressionsfunktion ergebenden Werte mit den Messwerten verglichen.

**Code 8-1** Nichtlineare Regression

Option Explicit

```

Sub eNlinRgress()
  Dim a(3), f(200) As Double
  Dim dchi2(3), absgrad, chi2, t As Double
  Dim w, xi, yi As Double
  Dim i, n, k, Count, Max As Integer

'Daten lesen
  n = Cells(13, 2)
  Dim x(n), y(n), d(n, 3) As Double

  t = Cells(14, 2)

```

```
Max = Cells(15, 2)

For i = 1 To n
    x(i) = Cells(i + 1, 1)
    y(i) = Cells(i + 1, 2)
Next i

a(1) = Cells(18, 2)
a(2) = Cells(19, 2)
a(3) = Cells(20, 2)

'Iterationsschleife
For Count = 1 To Max
    For i = 1 To n

        'Funktion und Ableitungen
        w = Exp(-x(i) * a(3))
        f(i) = a(1) + (a(2) - a(1)) * w
        d(i, 1) = 1 - w
        d(i, 2) = w
        d(i, 3) = -x(i) * (a(2) - a(1)) * w
    Next i

    'Berechnung von chi2(a)
    chi2 = 0
    For i = 1 To n
        chi2 = chi2 + (y(i) - f(i)) ^ 2
    Next i

    sqchi = Sqr(chi2 / n)
    Cells(Count, 4) = sqchi

    'Abbruchkriterium
    If sqchi <= t Then
        Cells(15, 3) = Count
        Cells(18, 3) = a(1)
        Cells(19, 3) = a(2)
        Cells(20, 3) = a(3)
        Count = Max
    End If

    'Berechnung von grad chi2(a) und abs(grad chi2(a))
    absgrad = 0
    For k = 1 To 3
        dchi2(k) = 0
        For i = 1 To n
```

```

        dchi2(k) = dchi2(k) - 2 * (y(i) - f(i)) * d(i,
            k)
    Next i
    absgrad = absgrad + dchi2(k) ^ 2
Next k

'Aktualisierung des Parametervektor
chi2 = chi2 - n * tol ^ 2
For k = 1 To 3
    a(k) = a(k) - (chi2 * dchi2da(k) / absgrad)
Next k
Next Count

'Berechnung der Funktionswerte
a1 = Cells(18, 3)
a2 = Cells(19, 3)
a3 = Cells(20, 3)
For i = 1 To n
    xi = Cells(i + 1, 1)
    yi = a1 + (a2 - a1) * Exp(-a3 * xi)
    Cells(i + 1, 3) = yi
Next i
End Sub

```

Nachfolgend sehen Sie das Ergebnis.

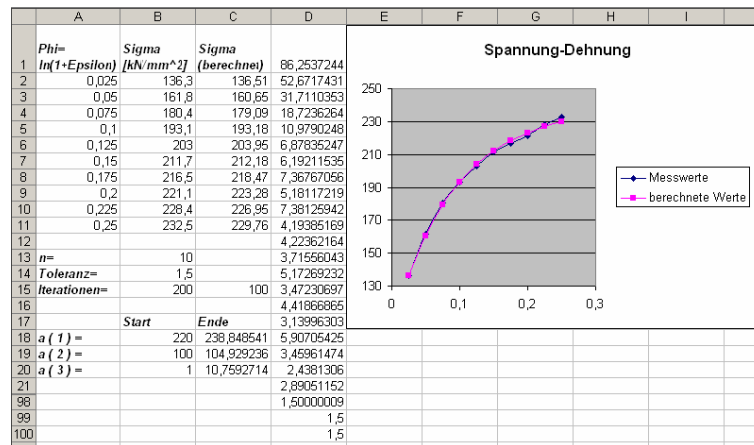


Abb. 8-3 Testdatenauswertung

### 8.3 Arrays und Datenfelder

Bereits im Buch wurde eine einfache Nutzwertanalyse beschrieben. Nachfolgend soll das Formblatt noch um Unterkriterien erweitert werden.

#### Beispiel: Nutzwertanalyse

In dem nachfolgenden Formblatt können zu jedem Kriterium vier verschiedene Unterkriterien eingegeben werden. Diese Unterkriterien werden in die Felder A 1.1 ... C 3.3 eingetragen. Nach Eingabe der Bewertungen ermittelt die Auswertung den maximalen Nutzen.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Lösungen	1. Kriterium			2. Kriterium			3. Kriterium		3		Nutzen
2		A 1.1	3		A 2.1	3		A 3.1	4			
3	A	A 1.2	4	4	A 2.2	2	3	A 3.2	5	5		32
4		A 1.3	3		A 2.3	2		A 3.3	4			
5		A 1.4	4		A 2.4	3		A 3.4	4			
6		B 1.1	3		B 2.1	3		B 3.1	2			
7	B	B 1.2	5	5	B 2.2	3	4	B 3.2	2	3		31
8		B 1.3	4		B 2.3	3		B 3.3	3			
9		B 1.4	4		B 2.4	4		B 3.4	3			
10		C 1.1	4		C 2.1	2		C 3.1	1			
11	C	C 1.2	3	5	C 2.2	2	4	C 3.2	2	2		28
12		C 1.3	4		C 2.3	4		C 3.3	2			
13		C 1.4	5		C 2.4	3		C 3.4	1			
14												

Abb. 8-4 Nutzwertanalyse mit Testdaten

#### Code 8-2 Erweiterte Nutzwertanalyse

```
Option Explicit

Sub Nutzwert_Formblatt()

    ThisWorkbook.Worksheets("Nutzwert").Cells.Clear

    Range("A1").Select
    ActiveCell.FormulaR1C1 = "Lösungen"
    Range("B1").Select
    ActiveCell.FormulaR1C1 = "1. Kriterium"
    Range("E1").Select
    ActiveCell.FormulaR1C1 = "2. Kriterium"
    Range("H1").Select
    ActiveCell.FormulaR1C1 = "3. Kriterium"
    Range("L1").Select
    ActiveCell.FormulaR1C1 = "Nutzen"
```

```
Range("A2").Select
ActiveCell.FormulaR1C1 = "A"
Range("A6").Select
ActiveCell.FormulaR1C1 = "B"
Range("A10").Select
ActiveCell.FormulaR1C1 = "C"

Range("A2:A5").Select
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .MergeCells = True
End With
Range("A6:A9").Select
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .MergeCells = True
End With
Range("A10:A13").Select
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .MergeCells = True
End With

Cells(2, 2) = "A 1.1"
Cells(3, 2) = "A 1.2"
Cells(4, 2) = "A 1.3"
Cells(5, 2) = "A 1.4"
Cells(6, 2) = "B 1.1"
Cells(7, 2) = "B 1.2"
Cells(8, 2) = "B 1.3"
Cells(9, 2) = "B 1.4"
Cells(10, 2) = "C 1.1"
Cells(11, 2) = "C 1.2"
Cells(12, 2) = "C 1.3"
Cells(13, 2) = "C 1.4"

Cells(2, 5) = "A 2.1"
Cells(3, 5) = "A 2.2"
Cells(4, 5) = "A 2.3"
Cells(5, 5) = "A 2.4"
Cells(6, 5) = "B 2.1"
Cells(7, 5) = "B 2.2"
Cells(8, 5) = "B 2.3"
```

```
Cells(9, 5) = "B 2.4"  
Cells(10, 5) = "C 2.1"  
Cells(11, 5) = "C 2.2"  
Cells(12, 5) = "C 2.3"  
Cells(13, 5) = "C 2.4"  
  
Cells(2, 8) = "A 3.1"  
Cells(3, 8) = "A 3.2"  
Cells(4, 8) = "A 3.3"  
Cells(5, 8) = "A 3.4"  
Cells(6, 8) = "B 3.1"  
Cells(7, 8) = "B 3.2"  
Cells(8, 8) = "B 3.3"  
Cells(9, 8) = "B 3.4"  
Cells(10, 8) = "C 3.1"  
Cells(11, 8) = "C 3.2"  
Cells(12, 8) = "C 3.3"  
Cells(13, 8) = "C 3.4"  
  
Range("D2:D5").Select  
With Selection  
    .HorizontalAlignment = xlCenter  
    .VerticalAlignment = xlCenter  
    .MergeCells = True  
End With  
Range("D6:D9").Select  
With Selection  
    .HorizontalAlignment = xlCenter  
    .VerticalAlignment = xlCenter  
    .MergeCells = True  
End With  
Range("D10:D13").Select  
With Selection  
    .HorizontalAlignment = xlCenter  
    .VerticalAlignment = xlCenter  
    .MergeCells = True  
End With  
  
Range("G2:G5").Select  
With Selection  
    .HorizontalAlignment = xlCenter  
    .VerticalAlignment = xlCenter  
    .MergeCells = True  
End With  
Range("G6:G9").Select  
With Selection
```

```
.HorizontalAlignment = xlCenter
.VerticalAlignment = xlCenter
.MergeCells = True
End With
Range("G10:G13").Select
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .MergeCells = True
End With

Range("J2:J5").Select
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .MergeCells = True
End With
Range("J6:J9").Select
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .MergeCells = True
End With
Range("J10:J13").Select
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .MergeCells = True
End With

Range("L2:L5").Select
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .MergeCells = True
End With
Range("L6:L9").Select
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .MergeCells = True
End With
Range("L10:L13").Select
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
```

```
.MergeCells = True
End With
End Sub

Sub Nutzwert_Testdaten()
Cells(1, 4) = 2
Cells(1, 7) = 3
Cells(1, 10) = 3

Cells(2, 3) = 3
Cells(3, 3) = 4
Cells(4, 3) = 3
Cells(5, 3) = 4
Cells(6, 3) = 3
Cells(7, 3) = 5
Cells(8, 3) = 4
Cells(9, 3) = 4
Cells(10, 3) = 4
Cells(11, 3) = 3
Cells(12, 3) = 4
Cells(13, 3) = 5

Cells(2, 6) = 3
Cells(3, 6) = 2
Cells(4, 6) = 2
Cells(5, 6) = 3
Cells(6, 6) = 3
Cells(7, 6) = 3
Cells(8, 6) = 3
Cells(9, 6) = 4
Cells(10, 6) = 2
Cells(11, 6) = 2
Cells(12, 6) = 4
Cells(13, 6) = 3

Cells(2, 9) = 4
Cells(3, 9) = 5
Cells(4, 9) = 4
Cells(5, 9) = 4
Cells(6, 9) = 2
Cells(7, 9) = 2
Cells(8, 9) = 3
Cells(9, 9) = 3
Cells(10, 9) = 1
Cells(11, 9) = 2
Cells(12, 9) = 2
```

```
Cells(13, 9) = 1
End Sub

Sub Nutzwert_Bestimmung()

    ActiveWorkbook.Names.Add Name:="A_1", _
        RefersToR1C1:="=Nutzwert!R2C3:R5C3"
    ActiveWorkbook.Names.Add Name:="B_1", _
        RefersToR1C1:="=Nutzwert!R6C3:R9C3"
    ActiveWorkbook.Names.Add Name:="C_1", _
        RefersToR1C1:="=Nutzwert!R10C3:R13C3"
    ActiveWorkbook.Names.Add Name:="A_2", _
        RefersToR1C1:="=Nutzwert!R2C6:R5C6"
    ActiveWorkbook.Names.Add Name:="B_2", _
        RefersToR1C1:="=Nutzwert!R6C6:R9C6"
    ActiveWorkbook.Names.Add Name:="C_2", _
        RefersToR1C1:="=Nutzwert!R10C6:R13C6"
    ActiveWorkbook.Names.Add Name:="A_3", _
        RefersToR1C1:="=Nutzwert!R2C9:R5C9"
    ActiveWorkbook.Names.Add Name:="B_3", _
        RefersToR1C1:="=Nutzwert!R6C9:R9C9"
    ActiveWorkbook.Names.Add Name:="C_3", _
        RefersToR1C1:="=Nutzwert!R10C9:R13C9"

    Cells(2, 4) = "=MAX(A_1)"
    Cells(6, 4) = "=MAX(B_1)"
    Cells(10, 4) = "=MAX(C_1)"

    Cells(2, 7) = "=MAX(A_2)"
    Cells(6, 7) = "=Max(B_2)"
    Cells(10, 7) = "=Max(C_2)"

    Cells(2, 10) = "=MAX(A_3)"
    Cells(6, 10) = "=Max(B_3)"
    Cells(10, 10) = "=Max(C_3)"

    Cells(2, 12) = Cells(2, 4) * Cells(1, 4) + _
        Cells(2, 7) * Cells(1, 7) + _
        Cells(2, 10) * Cells(1, 10)
    Cells(6, 12) = Cells(6, 4) * Cells(1, 4) + _
        Cells(6, 7) * Cells(1, 7) + _
        Cells(6, 10) * Cells(1, 10)
    Cells(10, 12) = Cells(10, 4) * Cells(1, 4) + _
        Cells(10, 7) * Cells(1, 7) + _
        Cells(10, 10) * Cells(1, 10)

End Sub
```

Sie können in die Auswertungsprozedur noch eine farbliche Kennung der ausgewählten Zellen integrieren.

## 8.4 Arbeiten auf Listenstrukturen

### Beispiel: Bubble-Sortierverfahren

Der Algorithmus beruht auf dem Prinzip des fortgesetzten Vergleichs und Tauschen von Nachbarelementen. Dabei wird die Sortierliste mehrmals durchlaufen, bis keine Vertauschung mehr stattfindet. Bei dem Vertauschen steigen die Elemente wie Blasen (Bubble) nach oben, auf den entsprechenden Platz der Sortierfolge.

Dieser Algorithmus lässt sich noch verbessern, wenn man sich merkt, ob während des Durchlaufs überhaupt eine Umordnung stattfindet. Oder, falls eine Vertauschung stattfand, in welcher Position.

Tab. 8-2 Bubblesort

Bubblesort									
i = 2, 1, n									
<table border="1"> <tr> <td colspan="2">j = n, -1, i</td> </tr> <tr> <td colspan="2">Ist <math>a(j-1) &gt; a(j)</math></td> </tr> <tr> <td>Ja</td> <td>Nein</td> </tr> <tr> <td>Vertausche <math>a(j-1) \leftrightarrow a(j)</math></td> <td>./.</td> </tr> </table>		j = n, -1, i		Ist $a(j-1) > a(j)$		Ja	Nein	Vertausche $a(j-1) \leftrightarrow a(j)$	./.
j = n, -1, i									
Ist $a(j-1) > a(j)$									
Ja	Nein								
Vertausche $a(j-1) \leftrightarrow a(j)$	./.								

Der Algorithmus ist übersichtlich und einfach zu programmieren.

Code 8-3 BubbleSort

```
Option Explicit
Sub BubbleSort_Test()
```

```
Dim x As Double
Dim i As Integer

x = Timer
Randomize (x)
For i = 1 To 100
    Cells(i, 1) = Rnd(x)
Next i
End Sub

Sub BubbleSort_Start()
    Dim MyDoc As Object
    Dim von, bis As Integer

    Set MyDoc = ThisWorkbook.Worksheets("Bubblesort")
    von = 1
    bis = MyDoc.UsedRange.Rows.Count
    Call BubbleSort
End Sub

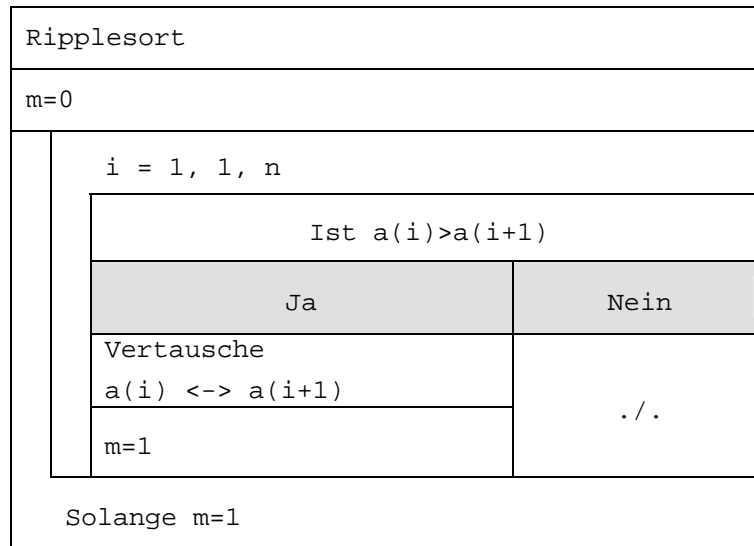
Sub BubbleSort()
    Dim i, j As Integer
    Dim x As Double

    For i = 2 To 100
        For j = 100 To i Step -1
            If Cells(j - 1, 1) > Cells(j, 1) Then
                x = Cells(j, 1)
                Cells(j, 1) = Cells(j - 1, 1)
                Cells(j - 1, 1) = x
            End If
        Next j
    Next i
End Sub
```

### Beispiel: Ripple-Sortierverfahren

Sind Listen zu sortieren, bei denen lediglich wenige Elemente auf der falschen Position sind, dann empfiehlt sich der RippleSort. Er ist die schnellste Methode für den Trivialfall.

**Tab. 8-3** Ripplesort

**Code 8-4** RippleSort

```

Option Explicit

Sub Ripplesort_Test()
    Dim x As Double
    Dim i As Integer

    x = Timer
    Randomize (x)
    For i = 1 To 100
        Cells(i, 1) = Rnd(x)
    Next i
End Sub

Sub Ripplesort_Start()
    Dim MyDoc As Object
    Dim von, bis As Integer

    Set MyDoc = ThisWorkbook.Worksheets("Ripplesort")
    von = 1
    bis = MyDoc.UsedRange.Rows.Count
    Call RippleSort
End Sub

Sub RippleSort()
    Dim i, m As Integer
    Dim x As Double

```

```

Do
  m = 0
  For i = 1 To 99
    If Cells(i, 1) > Cells(i + 1, 1) Then
      x = Cells(i, 1)
      Cells(i, 1) = Cells(i + 1, 1)
      Cells(i + 1, 1) = x
      m = 1
    End If
  Next i
Loop While m = 1
End Sub

```

**Beispiel: Rezepturen**

Ein Beispiel aus der Technik für Rezepturen sind Mischungen für Gummibeschichtungen, wie sie für Gummi-Metall-Verbindungen (z.B. auf Walzen) aufgebracht werden. Ausgegangen wird von einer Richtrezeptur, die bezogen auf eine Standardeinheit festgelegt ist. Die benötigte Menge wird dann über eine Abwiegevorschrift ermittelt. Dazu wird eine Liste aller Artikel (Zutaten) erstellt. In ihr werden

**Tab. 8-4** Beispiel für eine Rezepturliste

Artikel	Dichte $\delta$ [kg/dm <sup>3</sup> ]	Preis $P_k$ [€/ % kg]	Gew.-Teile der Richt- Rezeptur in pHr [%]
A	0,990	476,00	100,00
B	1,100	746,51	1,00
C	5,600	278,00	3,00
usw.			
Summen			254,30

Die Berechnungen zur Richtrezeptur und Abwiegevorschrift sehen dann wie folgt aus.

1. Eingabe aller Artikel  $a_i$ ,  $i=1, \dots, n$ , mit Artikeldichte  $\delta_i$ , Kalkulationspreis  $P_{k,i}$  und Gewichtsanteile der Richtrezeptur  $G_{r,i}$ .
2. Berechnung der Volumenanteile der Richtrezeptur:

$$V_{r,i} = \frac{G_{r,i}}{\delta_i}; i=1, \dots, n \quad (8.4.1)$$

Beispiel:  $V_{r,i} = \frac{100,0}{0,99} = 101,01$

3. Mit der Eingabe aller Gewichtsanteile ergibt sich auch deren Summe  $\Sigma G_r$ .
4. Mit der Berechnung aller Volumenanteile der Richtrezeptur ergibt sich auch deren Summe  $\Sigma V_r$ .
5. Bestimmung der Mischungsdichte lt. Richtrezeptur

$$\delta_r = \frac{\Sigma G_r}{\Sigma V_r} \quad (8.4.2)$$

Beispiel mit  $\Sigma G_r = 254,30; \Sigma V_r = 186,947$

$$\delta_r = \frac{254,30}{186,947} = 1,360$$

6. Bestimmung der Mischgröße aus der Angabe des Nutzinhaltes  $V_n$  in Liter und dem Sollfüllgrad  $f_s$  in %

$$V_m = \frac{V_n \cdot f_s}{100} \quad (8.4.3)$$

Beispiel mit  $V_n = 36,5 \text{Ltr.}; f_s = 100$

$$V_m = \frac{36,5 \cdot 100}{100} = 36,5$$

7. Berechnung der Summe der Gewichts-Einzelanteile der Richtrezeptur in kg

$$\sum G_e = V_m \cdot \delta_r \quad (8.4.4)$$

Beispiel  $\sum G_e = 36,5 \cdot 1,360 = 49,65$

8. Berechnung der Gewichts-Einzelanteile der Richtrezeptur

$$G_{e,i} = G_{r,i} \cdot \frac{\sum G_e}{\sum G_r}; i = 1, \dots, n \quad (8.4.5)$$

Beispiel  $G_{e,1} = 100 \cdot \frac{49,65}{254,30} = 19,524$

9. Berechnung der Gewichts-Einzelanteile der Abwiegevorschrift durch Eingabe eines Prozentsatzes ps

$$G_{a,i} = G_{r,i} \cdot \frac{ps}{100}; i = 1, \dots, n \quad (8.4.6)$$

Beispiel  $G_{a,1} = 100,0 \cdot \frac{20}{100} = 20,0$

10. Mit der Berechnung der Einzelanteile der Abwiegevorschrift ergibt sich auch deren Summe

$$\sum G_a = \sum_{i=1}^n G_{a,i} \quad (8.4.7)$$

Im Beispiel ist  $\sum G_a = 50,86$

11. Berechnung der Summe der pHr-Einzelanteile der Abwiegevorschrift

$$\sum p_a = \sum G_a \cdot \frac{100}{ps} \quad (8.4.8)$$

Beispiel  $\sum p_a = 50,86 \cdot \frac{100}{20} = 254,30$

12. Berechnung der pHr-Einzelanteile der Abwiegevorschrift

$$p_{a,i} = G_{a,i} \cdot \frac{\sum p_a}{\sum G_a}; i = 1, \dots, n \quad (8.4.9)$$

Beispiel  $p_{a,1} = 20,0 \cdot \frac{254,30}{50,86} = 100,0$

13. Berechnung der Volumen-Einzelanteile der Abwiegevorschrift

$$V_{a,i} = \frac{P_{a,i}}{\delta_i}; i = 1, \dots, n \quad (8.4.10)$$

Beispiel  $V_{a,1} = \frac{100,0}{0,99} = 101,01$

14. Berechnung der Summe der Volumen-Einzelanteile der Abwiegevorschrift

$$\sum V_a = \sum_{i=1}^n V_{a,i} \quad (8.4.11)$$

15. Berechnung der Dichte der Abwiegevorschrift

$$\delta_a = \frac{\sum G_a}{\sum V_a} \quad (8.4.12)$$

Beispiel  $\delta_a = \frac{254,30}{186,948} = 1,360$

16. Berechnung des Ist-Füllgrades

$$f_i = \frac{\frac{\sum G_a}{\delta_a}}{\frac{V_n}{100}} \quad (8.4.13)$$

Beispiel  $f_i = \frac{50,86 \cdot 100}{1,36 \cdot 36,5} = 102,458$

17. Berechnung der Preisanteile aus dem kalkulierten Preis

$$P_{t,i} = \frac{P_{a,i} \cdot P_{k,i}}{100}; i = 1, \dots, n \quad (8.4.14)$$

$$\text{Beispiel } P_{t,2} = \frac{1,0 \cdot 746,51}{100} = 7,47$$

18. Die Summe der Preisanteile ergibt sich aus

$$\sum P_t = \sum_{i=1}^n P_{t,i} \quad (8.4.15)$$

$$\text{Beispiel } \sum P_t = 760,81$$

19. Berechnung des Rezepturpreises per % kg

$$P_g = \frac{\sum P_t}{\frac{\sum G_a}{100}} \quad (8.4.16)$$

$$\text{Beispiel } P_g = \frac{760,81}{\frac{254,30}{100}} = 299,18$$

20. Berechnung des Rezepturpreises per % Ltr.

$$P_v = \frac{\sum P_t}{\frac{\sum V_a}{100}} \quad (8.4.17)$$

$$\text{Beispiel } P_v = \frac{760,81}{\frac{186,948}{100}} = 406,96$$

#### Code 8-5 Rezepturen

```
Option Explicit

Sub Rezeptur_Formblatt()

    ThisWorkbook.Worksheets("Rezeptur").Cells.Clear

    'Tabellenkopf
    Cells(1, 1) = "Artikeldaten"
    Cells(2, 1) = "Nr"
    Cells(2, 2) = "Dichte" & Chr(10) & "[kg/dm" & ChrW(179)
        & "]"
    Cells(2, 3) = "Preis" & Chr(10) & "[€/ % kg]"
    Cells(1, 4) = "Richtrezeptur"
```

```
Cells(2, 4) = "Gew-Teile" & Chr(10) & "[pHr]"
Cells(2, 5) = "Gew-Teile" & Chr(10) & "[kg]"
Cells(1, 6) = "Abwiegevorschrift"
Cells(2, 6) = "[pHr]"
Cells(2, 7) = "[kg]"
Cells(1, 8) = "Volumen laut"
Cells(2, 8) = "Richtr." & Chr(10) & "[dm" & ChrW(179) & "]"
Cells(2, 9) = "Abwiegev." & Chr(10) & "[dm" & ChrW(179) & "]"
Cells(1, 10) = "Preisanteil" & Chr(10) & "[€]"
Range("A1:J2").Select
Selection.Font.Bold = True
Selection.Font.Italic = True
Range("A1:J2").Select
Range("A2").Activate
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlBottom
End With
Range("D1:E1").Select
Range("E1").Activate
With Selection
    .MergeCells = True
End With
Range("F1:G1").Select
Range("G1").Activate
With Selection
    .MergeCells = True
End With
Range("H1:I1").Select
With Selection
    .MergeCells = True
End With
Range("H2").Select
With Selection
    .MergeCells = False
End With
Range("J1:J2").Select
With Selection
    .MergeCells = True
End With

'Spaltenbreiten
Columns("A:A").ColumnWidth = 4.6
Columns("B:B").ColumnWidth = 8.5
```

```
Columns("C:C").ColumnWidth = 9.3
Columns("K:K").ColumnWidth = 1.2

'Spaltenformate
Columns("A:A").Select
Range("A2").Activate
Selection.NumberFormat = "0"
Columns("B:B").Select
Range("B2").Activate
Selection.NumberFormat = "0.000"
Columns("C:D").Select
Range("C2").Activate
Selection.NumberFormat = "0.00"
Columns("E:E").Select
Range("E2").Activate
Selection.NumberFormat = "0.000"
Columns("F:F").Select
Range("F2").Activate
Selection.NumberFormat = "0.0"
Columns("G:I").Select
Range("G2").Activate
Selection.NumberFormat = "0.000"
Columns("J:J").Select
Range("J2").Activate
Selection.NumberFormat = "0.00"
Range("M8").Select
Selection.NumberFormat = "0.000"
Range("M9").Select
Selection.NumberFormat = "0.00"
Range("M10").Select
Selection.NumberFormat = "0.000"
Range("M11").Select
Selection.NumberFormat = "0.0"
Range("M12:M13").Select
Selection.NumberFormat = "0.00"

Range("A1:C1").Select
Range("C1").Activate
With Selection
    .MergeCells = True
End With

'Formularanteil
Cells(3, 12) = "Anzahl Artikel"
Cells(4, 12) = "Nutzzinhalt in Ltr."
Cells(5, 12) = "Sollfüllgrad in %"
```

```
Cells(6, 12) = "%-Satz der Abwieg."

Cells(8, 12) = "Dichte der Rezeptur"
Cells(9, 12) = "Mischgröße in Ltr."
Cells(10, 12) = "Dichte der Abwiegev."
Cells(11, 12) = "Istfüllgrad in %"
Cells(12, 12) = "Rezepturpreis per % kg"
Cells(13, 12) = "Rezepturpreis per % Ltr."
Range("L3:L13").Select
Selection.Font.Bold = True
Selection.Font.Italic = True
Columns("L:L").EntireColumn.AutoFit

Range("M3:M13").Select
With Selection
    .HorizontalAlignment = xlRight
    .VerticalAlignment = xlBottom
End With

'Startposition
Range("M3").Select
End Sub

Sub Rezeptur_Testdaten()
Cells(3, 13) = 13
Cells(4, 13) = 36.5
Cells(5, 13) = 100
Cells(6, 13) = 20

Cells(3, 1) = 1
Cells(3, 2) = 0.99
Cells(3, 3) = 476
Cells(3, 4) = 100

Cells(4, 1) = 2
Cells(4, 2) = 1.1
Cells(4, 3) = 746.51
Cells(4, 4) = 1

Cells(5, 1) = 3
Cells(5, 2) = 5.6
Cells(5, 3) = 278
Cells(5, 4) = 3

Cells(6, 1) = 4
Cells(6, 2) = 0.84
```

```
Cells(6, 3) = 150.53  
Cells(6, 4) = 2
```

```
Cells(7, 1) = 5  
Cells(7, 2) = 2.4  
Cells(7, 3) = 29.2  
Cells(7, 4) = 56
```

```
Cells(8, 1) = 6  
Cells(8, 2) = 2  
Cells(8, 3) = 204.54  
Cells(8, 4) = 25
```

```
Cells(9, 1) = 7  
Cells(9, 2) = 1.63  
Cells(9, 3) = 285  
Cells(9, 4) = 1.5
```

```
Cells(10, 1) = 8  
Cells(10, 2) = 1.25  
Cells(10, 3) = 344  
Cells(10, 4) = 2
```

```
Cells(11, 1) = 9  
Cells(11, 2) = 1.1  
Cells(11, 3) = 278.61  
Cells(11, 4) = 15.5
```

```
Cells(12, 1) = 10  
Cells(12, 2) = 1.06  
Cells(12, 3) = 879  
Cells(12, 4) = 11
```

```
Cells(13, 1) = 11  
Cells(13, 2) = 1.3  
Cells(13, 3) = 790  
Cells(13, 4) = 2
```

```
Cells(14, 1) = 12  
Cells(14, 2) = 1.19  
Cells(14, 3) = 875.01  
Cells(14, 4) = 0.3
```

```
Cells(15, 1) = 13  
Cells(15, 2) = 2  
Cells(15, 3) = 83
```

```
Cells(15, 4) = 35
End Sub

Sub Rezeptur_Auswertung()
  Dim n, i, z As Integer
  Dim SuGr, Vri, SuVr As Double
  Dim dr, Vn, fs, Vm As Double
  Dim SuGe, Gei, ps As Double
  Dim Gai, SuGa, Supa As Double
  Dim pai, Vai, SuVa As Double
  Dim fi, da, Pti As Double
  Dim SuPt, Pg, Pv As Double

  n = Cells(3, 13)
  z = 4 + n

  SuGr = 0
  For i = 1 To n
    SuGr = SuGr + Cells(2 + i, 4)
  Next i
  Cells(z, 4) = SuGr

  SuVr = 0
  For i = 1 To n
    Vri = Cells(2 + i, 4) / Cells(2 + i, 2)
    Cells(2 + i, 8) = Vri
    SuVr = SuVr + Vri
  Next i
  Cells(z, 8) = SuVr

  dr = SuGr / SuVr
  Cells(8, 13) = dr
  Vn = Cells(4, 13)
  fs = Cells(5, 13)
  Vm = Vn * fs / 100
  Cells(9, 13) = Vm
  SuGe = Vm * dr
  Cells(z, 5) = SuGe

  For i = 1 To n
    Gei = Cells(2 + i, 4) * SuGe / SuGr
    Cells(2 + i, 5) = Gei
  Next i

  ps = Cells(6, 13)
  SuGa = 0
```

```
For i = 1 To n
    Gai = Cells(2 + i, 4) * ps / 100
    Cells(2 + i, 7) = Gai
    SuGa = SuGa + Gai
Next i
Cells(z, 7) = SuGa

Supa = SuGa * 100 / ps
Cells(z, 6) = Supa
For i = 1 To n
    pai = Cells(2 + i, 4) * ps / 100 * Supa / SuGa
    Cells(2 + i, 6) = pai
Next i

SuVa = 0
For i = 1 To n
    Vai = Cells(2 + i, 6) / Cells(2 + i, 2)
    SuVa = SuVa + Vai
    Cells(2 + i, 9) = Vai
Next i
Cells(z, 9) = SuVa

da = Supa / SuVa
Cells(10, 13) = da
fi = (SuGa / da) / (Vn / 100)
Cells(11, 13) = fi

SuPt = 0
For i = 1 To n
    Pti = Cells(i + 2, 6) * Cells(i + 2, 3) / 100
    SuPt = SuPt + Pti
    Cells(i + 2, 10) = Pti
Next i
Cells(z, 10) = SuPt
Pg = SuPt / (Supa / 100)
Cells(12, 13) = Pg
Pv = SuPt / (SuVa / 100)
Cells(13, 13) = Pv
End Sub
```

In Abbildung 8-2 sehen Sie das Ergebnis der Auswertung. Das Programm ist so ausgelegt, dass eine Abwiegevorschrift für jede beliebige Anzahl von Rezeptanteilen erstellt werden kann.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Artikeldaten		Richtrezeptur		Abwiegenvorschrift		Volumen laut						
2	Nr	Dichte [kg/dm³]	Preis [€% kg]	Gew-Teile [g]	Gew-Teile [kg]	[g]	[kg]	Richtr. [dm³]	Abwieg. [dm³]	Preisanteil [€]			
3	1	0,990	476,00	100,000	19,524	100,000	20,000	101,010	101,010	476,00		Anzahl Artikel	13
4	2	1,100	746,51	1,000	0,195	1,000	0,200	0,909	0,909	7,47		Nutzhalt in Ltr.	36,5
5	3	5,600	278,00	3,000	0,586	3,000	0,600	0,536	0,536	8,34		Sollfüllgrad in %	100
6	4	0,840	150,53	2,000	0,390	2,000	0,400	2,381	2,381	3,01		%-Satz der Abwieg.	20
7	5	2,400	29,20	56,000	10,954	56,000	11,200	23,333	23,333	16,35			
8	6	2,000	204,54	25,000	4,881	25,000	5,000	12,500	12,500	51,14		Dichte der Rezeptur	1,380
9	7	1,630	285,00	1,500	0,293	1,500	0,300	0,920	0,920	4,28		Mischgröße in Ltr.	36,50
10	8	1,250	344,00	2,000	0,390	2,000	0,400	1,600	1,600	6,88		Dichte der Abwieg.	1,380
11	9	1,100	278,61	15,500	3,026	15,500	3,100	14,091	14,091	43,19		Isotillgrad in %	102,4
12	10	1,050	879,00	11,000	2,148	11,000	2,200	10,377	10,377	95,69		Rezepturpreis per % kg	299,19
13	11	1,300	790,00	2,000	0,390	2,000	0,400	1,538	1,538	15,80		Rezepturpreis per % Ltr.	408,96
14	12	1,190	875,01	0,300	0,059	0,300	0,060	0,252	0,252	2,63			
15	13	2,000	83,00	35,000	6,833	35,000	7,000	17,500	17,500	29,05			
16													
17				254,300	49,650	254,300	50,860	186,948	186,948	760,81			
18													

Abb. 8-5 Abwiegedaten zur Testrezeptur

**Beispiel: Stücklistenorganisation**

Ein einfaches Beispiel, wie eine Baumstruktur grafisch und farblich im Blatt dargestellt werden kann, zeigt die nachfolgende Prozedur.

Darin wird die Stufe abgefragt und in deren Abhängigkeit werden die Teilebezeichnungen an einer entsprechenden Position noch einmal eingestellt und farblich gekennzeichnet.

**Code 8-6** Baumstruktur in der Strukturstückliste

Option Explicit

```

Sub Struktur_Baum()
    Dim i, n, Stufe, Farbe As Integer
    Dim Bez As String
    Dim Mydoc As Object
    Set Mydoc = ThisWorkbook.Worksheets("S-P001")

    Columns("G:O").Select
    Selection.ColumnWidth = 5

    n = Mydoc.UsedRange.Rows.Count
    For i = 3 To n
        Bez = Cells(i, 4)
        Stufe = Right(Cells(i, 1), 1)
        Cells(i, 6 + Stufe) = Bez
        Cells(i, 6 + Stufe).Select
        Select Case Stufe
            Case 1, 2, 3
                Farbe = 47 - Stufe
            Case 4, 5, 6
                Farbe = 40 - Stufe
            Case 7
        End Select
    Next i

```

```

Farbe = 8
Case 8, 9
    Farbe = 50 - Stufe
Case Else
    Farbe = 0
End Select
Selection.Interior.ColorIndex = Farbe
Next i
End Sub

```

In der nachfolgenden Abbildung sehen Sie das Ergebnis. Es sind maximal 9 Stufen möglich.

	A	B	C	D	E	F	G	H	I
1	<b>S-P001- Baugruppe</b>								
2	Stufe	Pos.Nr.	Menge	Bezeichnung	Baugruppe	Material			
3	1	1	1	BG1	P002		BG1		
4	.2	1	1	ET1				ET1	
5	.2	2	2	ET2				ET2	
6	1	2	1	BG2	P003		BG2		
7	.2	1	2	ET1				ET1	
8	.2	2	3	ET3				ET3	
9	.2	3	1	BG3	P004			BG3	
10	.3	1	1	ET1					ET1
11	.3	2	4	ET1					ET1
12	.3	3	2	ET4					ET4
13	1	3	2	ET3			ET3		
14	1	4	1	ET4			ET4		

Abb. 8-6 Testdatenauswertung