Sortieren mit TUSTEP (Teil 2)

In der letzten Nummer der BI wurden die Grundprobleme der alphabetischen Sortierung einer Bibliographie besprochen, wobei die richtige Sortierung nach DIN 5007 im Vordergrund stand. In Fortsetzung dazu sollen hier anhand derselben Beispieldaten weitere Probleme des Sortierens besprochen werden: das Sortieren von Zahlen, das Sortieren nach rein inhaltlichen Kriterien und das Sortieren von Gruppen innerhalb einer Datei. Bevor diese Probleme in Beispielen erläutert werden, zunächst Bemerkungen zur Aufbereitung der Quelldaten.

Die Aufbereitung der Quelldaten

Die Beispiele in der letzten BI zeigen, daß es für eine sinnvolle Sortierung notwendig ist, auf die einzelnen Bestandteile eines Bibliographieeintrags zugreifen zu können, d. h. das Programm muß in der Lage sein, die Bestandteile *Autor, Titel, Jahr,* etc. eindeutig zu erkennen.

In gedruckten Bibliographien sind die Bestandteile durch ihre Reihenfolge gekennzeichnet und durch bestimmte Satzzeichen voneinander getrennt (Komma nach Autor, Punkt nach Titel, Doppelpunkt vor Zeitschrift, Klammern bei Reihen, etc.) oder typographisch hervorgehoben (Kapitälchen, Kursive). Dem verstehenden Leser genügen diese Kennzeichnungen. Er weiß, daß *Mayer* ein Personennamen ist, *Tübingen* der Verlagsort und nur dann der Titel sein kann, wenn noch ein weiterer Ort genannt ist. Der Computer dagegen hat dieses inhaltliche Wissen nicht und ist beim Erkennen der einzelnen Teile auf *eindeutige* Kennzeichen angewiesen. Die in gedruckten Bibliographien praktizierte Unterscheidung durch Satzzeichen und typographische Merkmale reicht dafür nicht aus.

Zweckmäßig ist es, in der Quelldatei für jede Kategorie (Autor, Titel, Jahr, etc.) eines Bibliographieeintrages eigens ein eindeutiges Kennzeichen einzutragen, am besten eine Kombination aus einem Sonderzeichen und weiteren Zeichen, die innerhalb der Daten sonst nicht vorkommt. Bewährt hat sich, als erstes Zeichen jeder Kennung dasselbe Sonderzeichen zu wählen (z. B. @, &, %), und die Katgorien durch weitere Zeichen, z. B. eine Zahl oder Buchstaben, die man sich leicht merken kann (im vorliegenden Beispiel **a** für Autor, **t** für Titel, **j** für Jahr, **u** für Untertitel, etc.), zu unterscheiden. Die Wahl der Kennungen ist beliebig.

Der Aufwand, der für die Aufbereitung der Quelldaten getrieben wird, lohnt die Mühe. Die Kennungen werden nicht nur für die Sortierung benötigt: Sie dienen z. B. im EDITOR als Feldmarkierungen für die Datenbanksuche. Oder es können anhand der Kennungen Plausibilitätstests zur Vollständigkeit, Reihenfolge und Richtigkeit der Einträge vorgenommen werden. Sie erlauben auch, Kommentare und Zitate in der Bibliographie mitzuverwalten. Die Kennungen können für die Ausgabe - sei es mit FORMATIERE in Schreibmaschinenschrift oder mit SATZ bei der professionellen Lichtsatzaufbereitung - in typographische Anweisungen und Steuerzeichen ausgetauscht werden, über die man sich bei der Erfassung noch keine Gedanken machen muß.

Die Erfassung der Daten mit den Kennungen ist nicht zeitaufwendig. Mit Hilfe eines Editor-Makros läßt sich eine Eingabemaske jeweils auf den Bildschirm schreiben. Beim Eintragen springt man mit dem Cursor von Kennung zu Kennung. Diese Art der Eingabe hat den Vorteil, daß man die Kennungen vor sich hat und schnell sieht, ob der Bibliographieeintrag vollständig erfaßt ist. Bleibt eine Kategorie leer, so stört es das Eintragen und die folgenden Programme nicht.

Durch das Eintragen mit einer Maske stehen die Kategorien wie in Abb. 1 gleich übersichtlich je in einer neuen Zeile und immer in derselben Reihenfolge. (Hat man die Quelldaten bereits in anderer Form vorliegen, kann man mit dem KOPIERE-

Aufruf aus Beispielprogramm Abb. 4 eine übersichtliche Zeileneinteilung herstellen und mit dem Programm aus Abb. 5 eine feste Reihenfolge erreichen.)

Im ersten Beispiel hier soll gezeigt werden, wie man eine bestimmte Reihenfolge der Kategorien per Programm herstellt. Bedingung ist, daß der Anfang (oder das Ende) eines Bibliographieeintrags eindeutig zu erkennen ist. Die Kategorien können dann in den Beispieldaten auch in unterschiedlicher Reihenfolge stehen.

Das Umstellen der einzelnen Kategorien innerhalb eines jeden Bibliographieeintrags läßt sich durch gruppenweises Sortieren der jetzigen Quelldaten erreichen. Für das Beispiel sollen die Kategorien in die Reihenfolge *Signatur - Autor - Jahr - Zeitschrift - Band - Ort - Titel - Kommentar* gebracht werden, wie es sinnvoll sein könnte, um für einen Standortkatalog die Signatur in exponierter Stellung zu haben.

Bei dieser Aufgabe wird eine Sortiereinheit nicht vom ganzen Bibliographieeintrag gebildet, sondern von jeder einzelnen Kategorie. Dazu darf in einem TUSTEP-Satz nicht mehr als eine Kategorie stehen. Die Daten von Abb. 1 erfüllen diese Bedingung. Auch steht jede Kategorie in genau einem Satz; so braucht man sich bei SORTIER-VORBEREITE um die Sortiereinheiten nicht zu kümmern.

Die Katgorien werden nach ihren Kennungen - d. h. nach den Kennbuchstaben - sortiert. Allerdings entspricht die alphabetische Reihenfolge der Kennbuchstaben nicht der gewünschten Reihenfolge. Sie muß im Programm festgelegt werden. Eine Lösung um die Reihenfolge festzulegen wäre, die Kennungen in Zahlen auszutauschen, wie im nächsten Beispiel beschrieben wird.

Man kann auch einen anderen Weg gehen. Da die Unterscheidung der Kategorien durch genau ein Zeichen erfolgt, kann man für diese Zeichen ein eigenes Sortieralphabet definieren. Man legt für das Sortieralphabet mit Parameter A1 die Wertigkeit der Kennbuchstaben fest. Werden die Kennbuchstaben nach dem Sortieralphabet **sajzbotk** sortiert, so entsteht die gewünschte Reihenfolge der Kategorien.

Würde man das Programm so starten, wäre das Ergebnis völlig unbrauchbar: zu Beginn stünden alle Signaturen, dann folgten alle Bandangaben, dann alle Autoren, d. h. die Bibliographieeinträge wären auseinandergerissen, denn man hätte die *ganze* Datei nach Kategorien sortiert. Gewünscht wird aber eine Sortierung innerhalb eines Bibliographieeintrags, d. h. es muß jeweils eine Gruppe von Sortiereinheiten (die zusammen einen Bibliographieeintrag bilden) in sich sortiert werden. Dazu gibt man in SORTIER-VORBEREITE mit Parameter NSN an, daß bei **&a** jeweils eine neue Gruppe von Sortiereinheiten beginnt, die in sich sortiert werden sollen. (Durch **&a** ist in den Quelldaten der Beginn eines Bibliographieeintrags gekennzeichnet.) Das Programm vergibt für jede Sortiergruppe eine neue laufende Nummer, die dreistellig (Angabe mit Parameter SNL) an den Anfang des Sortierfeldes geschrieben wird, so daß sie beim Sortieren das erste Kriterium bildet. Der aufgebaute Sortierschlüssel wird erst als zweites Kriterium herangezogen. Die Kategorien eines Bibliographieeintrags bleiben auf diese Weise zusammen, werden aber in sich richtig sortiert.

Das Programm, das die Datei BIB in der gewünschten Form sortiert, ist in Abb. 5 wiedergegeben.

Die Sortiereinheit *&o St. Ottilien* (Eintrag *Alfons Auer*) hat z. B. als Sortiertext **o**, ihre Sortiernummer ist 1 (es handelt sich in der Datei BIB um die erste Sortiergruppe). Sortiert wird nach dem Sortierfeld **001o**; die Sortiernummer wurde auf drei Stellen festgelegt; intern wird das **o** durch die Angabe des eigenen Sortieralphabets in einen entsprechenden Wert ausgetauscht, damit die Sortierung stimmt.

Als Ergebnis erhält man die Datei BIB-KAT (siehe Abb. 6). Die Kategorien in jedem Bibliographieeintrag stehen in der gewünschten Reihenfolge.

Das Definieren eines eigenen Sortieralphabets wird man in der Praxis vor allem benötigen, um Sprachen mit nicht-lateinischen Alphabeten (z. B. Griechisch, Hebräisch, Arabisch, Russisch, Syrisch) zu sortieren. Auch für erweiterte lateinische Alphabete wie z. B. Norwegisch (*æ, ø, å*) oder Spanisch (*ch, ll, ñ*) wird es - in Kombination mit dem Austauschen (Parameter XS1, vgl. Teil 1) - benötigt.

Sortieren in eine inhaltliche Reihenfolge (2. Beispiel)

Im letzten Beispiel wird gezeigt, wie es möglich ist, eine Sortierung nach Kriterien vorzunehmen, die nicht der alpha-numerischen Reihenfolge des für die Sortierung

ausgewählten Textes entsprechen. Das ist z. B. bei der Sortierung von Bibelstellen der Fall. In der Reihenfolge der biblischen Bücher, nach der üblicherweise sortiert wird, kommt *Genesis* vor *Amos*, was alphabetisch nicht zu erreichen ist. Von der Entscheidung des Wissenschaftlers hängt auch ab, ob für die Reihenfolge der biblischen Bücher die Hebräische, die Griechische, die Lateinische, die Lutherische oder die Katholische heranzuziehen ist.

Als Beispiel für dieses Vorgehen sollen hier die Einträge der Bibliographie auf folgende Weise nach den Signaturen sortiert werden: erstes Sortierkriterium sind die Bibliotheken in der Reihenfolge Landesbibliothek (LB), Universitätsbibliothek (UB) und zum Schluß das Historische Institut (HI). Diese Reihenfolge hat nichts mit der alphabetischen Ordnung zu tun, sondern wurde nach rein inhaltlichen Gesichtspunkten vom Bearbeiter festgelegt.

Werden die Signaturen in erster Linie nach den Bibliotheken sortiert, so gibt es anschließend auch keine Probleme mit der Anordnung der Signaturen in sich, da je Bibliothek das Signaturensystem einheitlich ist. Zu beachten ist aber die Sortierung der arabischen und römischen Zahlen: Bei gewöhnlicher alphabetischer Sortierung von links nach rechts fortschreitend spielt die erste Position die entscheidende Rolle. Wendet man dies auf arabische Zahlen an, so wird in erster Linie nach der ersten Ziffer sortiert, und die für die mathematische Reihenfolge wichtige Stellenzahl bleibt unberücksichtigt, was eine Reihenfolge *1, 11, 2* ergibt. Für eine mathematische Sortierung müssen alle Zahlen auf die gleiche Stellenzahl mit führenden Nullen ergänzt werden. Die Anzahl der Stellen legt man in SORTIER-VORBEREITE mit dem Parameter DEZ (DEZimalstellen) fest.

Auch römische Zahlen können nicht einfach alphabetisch sortiert werden: *IX* soll nach *VIII* kommen, was nach der alphabetischen Reihenfolge nicht der Fall ist. Die richtige Sortierung erreicht man, indem römische Zahlen für den Sortierschlüssel in (vierstellige) arabische Zahlen umgewandelt werden, was mit Parameter R1 möglich ist, wenn vor der römischen Zahl eine eindeutige Kennung steht, die selbst für die Sortierung nicht relevant ist. Ist eine solche Kennung nicht vorhanden, kann sie, wenn die römischen Zahlen durch ihre Umgebung eindeutig zu erkennen sind, beim Erstellen des Sortiertextes mit Parmeter XX1 eingefügt werden. Im vorliegenden Beispiel sind die römischen Zahlen innerhalb der Signatur eindeutig bestimmt durch die Bedingung: beliebig viele Zeichen aus der Gruppe der römischen Zahlzeichen, denen ein Kleinbuchstabe, gefolgt von Blank, vorausgeht.

Die Signaturen haben noch eine Besonderheit: Zeitschriften stehen in dem Beispiel jeweils unter einer Signatur. Sie werden in sich durch ihre Bandnummer unterschieden, die zur Sortierung heranzuziehen ist.

Der Sortiertext zur Lösung des Problems setzt sich zusammen aus der Signatur (&s bis zum nächsten & - Parameter AK1 und EK1) und der Bandangabe (&b bis zum nächsten & - Parameter AK2 und EK2). Wenn kein Band angegeben ist, wirkt sich das auf die Sortierung nicht aus. Der Sortiertext ist lediglich kürzer.

In Abb. 7 ist das Sortierprogramm wiedergegeben; Quelldatei ist die Datei von Abb. 1. (Man kann mit dem Programm auch die Datei in Abb. 6 bearbeiten, was eventuell sinnvoller ist: die Signatur steht dort an erster Stelle, was für einen echten Standortkatalog u. U. übersichtlicher ist.)

Der Programmablauf sei an der dritten Sortiereinheit (*Auer*) verdeutlicht: Der Sortiertext beginnt nach &s und endet vor dem nächsten &, das in dieser Sortiereinheit nicht gefunden wird, da die Signatur als letzte Kategorie steht. Dies stört nicht: wird keine Endekennung gefunden, so endet der ausgewählte Teil am Ende der Sortiereinheit. Der zweite Teil des Sortiertextes beginnt nach &b (das folgende Blank gehört noch zur Anfangskennung und noch nicht zum ausgewählten Text) und endet vor dem nächsten &, d. h. der zweite Teil besteht in diesem Fall, da kein &b gefunden wird, aus einer leeren Zeichenfolge (nichts). Zwischen dem ersten Teil und dem zweiten Teil werden - auch wenn der zweite Teil leer ist - zwei Blanks ergänzt; damit setzt sich der zweite Teil bei der Sortierung vom ersten Teil ab. Der ausgewählte Sortiertext ist:

LB: Vi VIII 229

wobei abschließend zwei Blanks stehen. In dem ausgewählten Sortiertext werden Zeichenfolgen ausgetauscht: LB: wird in 1 ausgetauscht; zur Kennzeichnung der römischen Zahl wird # ergänzt, das durch die Bedingung *Kleinbuchstabe - Blank - römisches Zahlzeichen* gefunden wird. Der Sortiertext lautet:

1 Vi #VIII 229

Daraus wird der Sortierschlüssel erstellt. Die arabischen Zahlen werden auf fünf Stellen mit führenden Nullen aufgefüllt, die Zeichen nach # als römische Zahl interpretiert und in eine vierstellige arabische Zahl umgewandelt. Der Sortierschlüssel, nach dem sortiert wird, lautet:

00001 vi 0008 00229

Dies liefert das gewünschte Ergebnis. Die Sortiereinheiten stehen in der Zieldatei in folgender Reihenfolge (angegeben ist die Seitennummer der Quelle sowie die Signatur und Bandangabe, nach der sortiert wurde):

(4): *&s LB: Vi VIII 25*
(3): *&s LB: Vi VIII 229*
(8): *&s LB: Vi IX 37*
(6): *&s UB: 9 C 10387*
(1): *&s UB: 10 A 6384*
(7): *&s Historisches Institut: Ac 35.16 &b 23*
(5): *&s Historisches Institut: Ze 1.09 &b 34*
(2): *&s HI: Ze 1.09*

Mit den drei Beispielen wurden typische Anwendungen des Sortiere-Programms vorgestellt. Kompliziertere Anwendungen lassen sich durch Kombination der vorgestellten Lösungsstrategien erarbeiten.

Abbildung 6

```
1.1   &s UB: 10 A 6384
1.2   &a Denoix, Ren%/e
1.3   &j 1980
1.4   &o München
1.5   &t Geschichte der Stadt Trier
1.6   &k Hier steht beliebig vieler
      Kommentar: Zitate,
      Bearbeitungshinweise, etc.
2.1   &s HI: Ze 1.09
2.2   &a Zenger, Zacharias
2.3   &j 1968
2.4   &z Historische Quartalschrift
2.5   &b 34
2.6   &o Koblenz
2.7   &t Statistik zur Geschichte
3.1   &s LB: Vi VIII 229
3.2   &a Auer, Alfons
3.3   &j 1988
3.4   &o St. Ottilien
3.5   &t Die Geschichte der Ethik
4.1   &s LB: Vi VIII 25
4.2   &a Jäger, Jakob
4.3   &j 1953
4.4   &o Göttingen
4.5   &t Leben und Werk Karls V.
5.1   &s Historisches Institut: Ze 1.09
5.2   &a Jaeger, Jakob
5.3   &j 1957
5.4   &z Historische Quartalschrift
5.5   &b 23
5.6   &o Koblenz
5.7   &t Württemberg und Hohenzollern
5.8   &k Auch hier steht Kommentar
6.1   &s UB: 9 C 10387
6.2   &a Zenger, Zacharias
6.3   &j 1970
6.4   &o Trier
6.5   &t Hessen und seine Wirtschaft
7.1   &s Historisches Institut: Ac 35.16
7.2   &a D%/enoix, Ren%/e
7.3   &j 1971
7.4   &o Köln
7.5   &t Einführung in die Geschichte
8.1   &s LB: Vi IX 37
8.2   &a Jäger, Gustav
8.3   &j 1987
8.4   &o Reutlingen
8.5   &t Mein Tübingen
```

Abbildung 5

```
#sv,bib, -std-,-,+,*
        Sortiergruppe
nsn       /&a/
snl       3
        Sortiertext ist der Kennbuchstabe
ak1       /&/
ek1       / /
aei       11
        Sortieralphabet
a1        sajzbotk
ssl       1
*eof
#so,-std-,-std-,1+4,+,1+4
#ko,-std-,bibkat,+,+,*
sa        /&s/
*eof
```

Abbildung 7

```
#sv,bib, -std-,-,+,*
        Sortiereinheit
aa        /&a/
        Sortiertext: Signatur und Band

ak1       /&s /
ek1       / &/
ak2       /&b /
ek2       / &/
aei       11  11
        Röm. Zahlen mit Kennung # versehen
>1z       ivxlcdm
xx1       />* >1/>=01 #<=01/
        Reihenfolge der Bibliotheken
xx1       /lb:/1/
xx1       /ub:/2/
xx1       /historisches institut:/3/
xx1       /hi:/3/
        Nach # röm. in arab. Zahl wandeln
r1        /#/
        Stellenzahl für arab. Zahlen
dez       5
        Länge für Sortierschlüssel
ssl       30
*eof
#so,-std-, -std-,1+30,+,1+30
#ko,-std-, bibsig,+,+,*
za        /&/
sa        /&a/
*eof
```