

### Das Sortieren mit TUSTEP: dargestellt am Beispiel einer Bibliographie

Bader, Winfried

Veröffentlichungsversion / Published Version  
Zeitschriftenartikel / journal article

Zur Verfügung gestellt in Kooperation mit / provided in cooperation with:  
GESIS - Leibniz-Institut für Sozialwissenschaften

#### Empfohlene Zitierung / Suggested Citation:

Bader, W. (1989). Das Sortieren mit TUSTEP: dargestellt am Beispiel einer Bibliographie. *Historical Social Research*, 14(3), 139-160. <https://doi.org/10.12759/hsr.14.1989.3.139-160>

#### Nutzungsbedingungen:

Dieser Text wird unter einer CC BY Lizenz (Namensnennung) zur Verfügung gestellt. Nähere Auskünfte zu den CC-Lizenzen finden Sie hier:  
<https://creativecommons.org/licenses/by/4.0/deed.de>

#### Terms of use:

This document is made available under a CC BY Licence (Attribution). For more information see:  
<https://creativecommons.org/licenses/by/4.0>

## COMPUTER - SECTION

---

### **Das Sortieren mit TUSTEP: Dargestellt am Beispiel einer Bibliographie**

*Winfried Bader\**

Unter Sortieren versteht man in der Textdatenverarbeitung das Ordnen von Daten in eine bestimmte Reihenfolge. Sortierbefehle gibt es in vielen Computer-Programmen. Meist verbirgt sich dahinter aber nur eine Anordnung des Wortlauts in der Reihenfolge des ASCH-Zeichensatzes. Intelligendere Programme ordnen die deutschen Umlaute zu den entsprechenden nichtumgelauteten Vokalen. Doch damit beginnen die eigentlichen Probleme der Sortierung wissenschaftlicher Textdaten erst. Zu deren Lösung genügt dann nicht mehr ein einfacher Sortierbefehl, sondern man muß die Möglichkeit haben, seine speziellen Sortierwünsche dem Sortierprogramm mitteilen zu können.

#### **Textdaten-Verarbeitung mit TUSTEP**

Das »Tübinger System von Textverarbeitungsprogrammen« TUSTEP - entwickelt seit 1966 in der Abteilung für »Literarische und Dokumentarische Datenverarbeitung« (LDDV) am Zentrum für Datenverarbeitung der Universität Tübingen - stellt für die wichtigsten Grundoperationen

---

\* Address all Communications to Winfried Bader, Zentrum für Datenverarbeitung (ZDV), Brunnenstr. 27, D-7400 Tübingen.

TUSTEP gibt es als PC-Version unter MS-DOS (IBM PC's und kompatibel). Auf Großrechnern läuft TUSTEP derzeit unter den Betriebssystemen MVS (IBM), VM/CMS (IBM), VMS (DEC VAX); die Implementierung unter BS2000 (SIEMENS) und UNIX ist geplant.

Für Hochschulen gibt es für TUSTEP Campus-Lizenzen. Nähere Informationen über den Vertrieb von TUSTEP und den Bezug des Handbuchs sind bei Prof. Dr. Ott am Zentrum für Datenverarbeitung der Universität Tübingen zu erhalten. Für die Beratung in Fragen zu TUSTEP stehen die Mitarbeiter der Abteilung »Literarische und Dokumentarische Datenverarbeitung« am ZDV in Tübingen zur Verfügung.

der wissenschaftlichen Textdatenverarbeitung Programme bereit, deren Leistung der Benutzer über Parameter spezifizieren kann. Der modulare Aufbau von TUSTEP erlaubt es dem Wissenschaftler, die einzelnen Module wie Bausteine eines Baukastens in vielfältiger Weise für die Lösung verschiedenster Aufgabenstellungen zu kombinieren und komplexe Probleme zu lösen.

In dem vorliegenden Artikel soll ein wichtiger Vorgang der Textdatenverarbeitung vorgestellt werden: das Sortieren. Dafür benötigt man in TUSTEP die beiden Bausteine (Grundoperationen, Module) Vorbereiten der Sortierung (Kommando SORTIER·VORBEREITE) und das eigentliche Sortieren (Kommando SORTIERE). Die Anwendung der Bausteine soll anhand von einfachen, aber praxisnahen Beispielen vorggeführt werden. Komplexere Probleme, z. B. das automatische Aufbereiten der Quelldaten oder die Weiterverarbeitung zur Druckausgabe, zu deren Lösung man noch weitere Bausteine benötigt, werden lediglich angedeutet.

Das Ziel des Artikels ist, allgemein in die Probleme des Sortierens einzuführen, einen Einblick in die Flexibilität von TUSTEP zu geben, wobei es zu berücksichtigen gilt, daß im folgenden nur ein kleiner Teil der Möglichkeiten der Sortierung in TUSTEP vorgestellt wird. Für denjenigen, der TUSTEP bereits benutzt, ist der Artikel als Einführung in die Programmerstellung mit dem Kommando SORTIER·VORBEREITE gedacht. Die vorgestellten Beispielpprogramme lassen sich dann leicht für andere Zwecke abändern und erweitern.

### **Grundprobleme des Sortierens**

Sortierte Daten begegnen einem überall: Namen im Telefonbuch, Kärtchen, einer Kartei, Listen und Register. All die Sortierungen haben den Zweck, die Daten in eine festgelegte Reihenfolge zu bringen, so daß jeder einzelne Eintrag innerhalb der Liste einen eindeutigen Platz bekommt und auffindbar ist.

Die mit der Sortierung verbundenen Probleme beginnen aber nicht erst bei der Frage, wie die Umlaute und der Buchstabe ß zu behandeln sind, sondern schon vorher bei der Abgrenzung der Einträge, die von der Sortierung betroffen sind (Sortiereinheit), bei der Festlegung des Teils eines Eintrags, der für die Sortierung relevant ist (Sortiertext), und erst dann ist die Frage nach den Alphabetisierungsregeln zu stellen (Sortierschlüssel).

### Die Sortiereinheit

Die Sortiereinheit ist jeweils der ganze Eintrag, der zusammengehörende Text, der als Ganzes von der Sortierung betroffen ist. Im Telefonbuch ist eine solche Einheit Name, Vorname, ggf. Beruf, Firma, Adresse und die Telefonnummer. In einer Kartei ist die Sortiereinheit eine Karteikarte. In den Regalen einer Bibliothek ist die Sortiereinheit das Buch, das nach Signaturen geordnet in den Regalen steht. In einer Bibliographie schließlich bildet Autor, Titel, Jahr, Ort, Signatur, ggf. Zeitschrift, Band, Kommentar, etc., eine Sortiereinheit kurz alles, was zu einer kompletten Arbeitsbibliographie an Information und Notizen dazugehört (vgl. unten das Beispiel).

Diese selbstverständlichen Erfahrungen mit Sortiereinheiten sind auch bei Sortierungen mit Computer-Programmen zu beachten. TUSTEP kennt bei der Bereitstellung der Quelldaten feste, vorgegebene Einheiten, sogenannte Sätze. Das ist jeweils der in der Regel einer Eingabezeile entsprechende Text, der im EDITOR mit einer eigenen Satznummer versehen ist (vgl. unten die Beispieldaten). TUSTEP-Programme betrachten bei der Verarbeitung der Daten, soweit nichts anderes angegeben ist, einen solchen Satz als eine Einheit zur Bearbeitung. Beim Kommando SORTIER VORBEREITE gilt ein TUSTEP-Satz als eine Sortiereinheit.

Es wird aber sicher oft vorkommen (vgl. unten das Beispiel), daß in einem Satz nicht genau das steht, was man inhaltlich als eine Sortiereinheit betrachten will, daß sich ein Bibliographieeintrag über mehrere Sätze erstreckt. In diesem Fall kann man angeben, welches Kennzeichen (eindeutige Zeichenfolge) in den Quelldaten den Beginn einer neuen Sortiereinheit kennzeichnet. Im vorliegenden Beispiel markiert die Zeichenfolge **&a** jeweils den Beginn einer Sortiereinheit. Über Parameter, die dem Kommandoaufruf folgen, wird dem Programm SORTIER VORBEREITE diese Zeichenfolge als Beginn der Sortiereinheit mitgeteilt (AA /&a/: Abschnitt Anfang bei **&a**). Damit gilt für die weitere Verarbeitung der Text zwischen zwei solchen Kennungen als Sortiereinheit, die als ganzes durch die Sortierung ihren bestimmten Platz im Ergebnis erhält.

### Der Sortiertext

Auch ein zweites wichtiges Problem der Sortierung macht man sich im alltäglichen Umgang mit sortierten Daten oft nicht richtig bewußt: es ist die Auswahl des Sortiertextes aus der Sortiereinheit. Nur der Sortiertext aber ist für die Sortierung relevant.

Der Bibliothekar braucht einen Standortkatalog seiner Bibliothek, der Benutzer dagegen wird in der Regel einen Autorenkatalog bevorzugen. Bei beiden Katalogen handelt es sich um die gleichen Karteikärtchen (Sortier-

einheiten), die jeweils unter anderer Hinsicht sortiert sind. Im Standardkatalog gilt als Kriterium für das Sortieren die Signatur, im Autorenkatalog der Autorenname, d. h. in den beiden Sortierungen besteht ein Unterschied im *Sortiertext*. Die Wahl des Sortiertextes bestimmt die Hinsicht, unter der die Sortierung erfolgen soll, denn in gedruckten Listen kann man nicht mehrere Kriterien ineinander mischen.

Es ist auch möglich, verschiedene Teile der Sortiereinheit auszuwählen, die zusammen den Sortiertext bilden. Damit wird durch den Sortiertext nicht nur das Hauptkriterium der Sortierung angegeben, sondern es werden gleich auch untergeordnete Kriterien berücksichtigt, die dann greifen, wenn durch den ersten Teil des Sortiertextes zwei Sortiereinheiten sich nicht unterscheiden. Wird eine Bibliographie nach Autoren sortiert, so muß man sich überlegen, in welcher Reihenfolge mehrere Werke eines Autors angeordnet werden. Man kann z. B. als Kriterium (d. h. als zweiten Teil des Sortiertextes) die alphabetische Ordnung der Titel wählen, oder das Erscheinungsjahr.

Ausgewählt wird der Sortiertext aus der Sortiereinheit durch die Angabe von eindeutigen Zeichenfolgen (Kennungen), die in den Quelldaten stehen. Durch Parameter wird dem Programm SORTIER·VORBEREITE mitgeteilt, welche Zeichenfolgen den Anfang und das Ende des Sortiertextes kennzeichnen. Im Beispiel unten beginnen die Autoren mit der Zeichenfolge &a und enden vor dem nächsten &, das Erscheinungsjahr beginnt mit &j und endet vor dem nächsten &. Die Zeichenfolgen werden in den Parametern AK1, EK1 und AK2, EK2 etc. (Anfang bzw Ende des Kopierens für den 1., 2. ... Teil des Sortiertextes) angegeben, womit dann der Anfang und das Ende des ersten und zweiten Teils des Sortiertextes festgelegt ist. Der davon eingeschlossene Text wird als erster bzw. zweiter Teil des Sortiertextes übernommen. Über einen weiteren Parameter (AEI; Anfang Ende-Index) kann man für die Behandlung der Kennungen angeben, ob sie zum Sortiertext gehören sollen, oder wie bei dem angedeuteten Beispiel der Bibliographie nicht dazugerechnet werden sollen (näheres dazu kann man dem TUSTEP-Handbuch entnehmen).

Diese Art der Auswahl von Textteilen aus den Quelldaten für bestimmte Zwecke setzt natürlich eine gewisse Aufbereitung der Daten voraus, ohne die aber eine maschinelle Bearbeitung prinzipiell unmöglich ist. Näheres zu dem Problem der Aufbereitung der Quelldaten ist weiter unten in einem eigenen Abschnitt aufgeführt.

### Der Sortierschlüssel

Sind die Sortiereinheiten festgelegt und die Sortiertexte daraus isoliert, so stellt sich nun die Frage nach den Alphabetisierungsregeln, welches Alphabet gelten soll, wie Umlaute zu behandeln sind, was mit Akzenten und

Sonderzeichen geschehen soll. Diese Probleme werden mit Hilfe des Sortierschlüssels gelöst, der durch das Programm SORTIERVORBEREITE aus dem Sortiertext aufgebaut wird. Er ist dann allein für die eigentliche Sortierung mit dem Kommando SORTIERE verantwortlich.

Der Sortierschlüssel hat eine doppelte Funktion. Zunächst eine inhaltliche: bei der Erstellung des Sortierschlüssels aus dem Sortiertext kann der Benutzer all die im Folgenden genannten Probleme und Regeln der Alphabetisierung berücksichtigen. Zum anderen eine technische: der Sortierschlüssel hat eine Form, die es dem eigentlichen Sortierprogramm erlaubt, mit schnell arbeitenden Algorithmen die Sortierung vorzunehmen. Sortiert werden nicht die Zeichen des Sortiertextes, sondern jedem dieser Zeichen wird für den Sortierschlüssel ein Wert zugewiesen, der dann die Grundlage für die Sortierung ist.

Für das Aufbauen des Sortierschlüssels hat man u. a. die Möglichkeit, ein ganz neues Sortieralphabet anzugeben, d. h. die Reihenfolge der Zeichen gänzlich neu zu bestimmen. Wem es sinnvoll erscheint, der kann z. B. mit dem Parameter A1 (Alphabet für den 1. Sortierschlüssel) für seine Sortierung die Reihenfolge qwertyuiopasdfghjklzxcvbnm (das ist die Reihenfolge der Schreibmaschinentastur) festlegen; alle Wörter werden dann in der Reihenfolge dieses angegebenen Alphabets sortiert. Gebraucht wird die Angabe eines eigenen Sortieralphabets bei der Sortierung fremdsprachiger Texte wie Griechisch, Hebräisch, Arabisch, Kyrillisch, Syrisch, Phonetik (um die Zeichensätze zu nennen, die von TUSTEP unterstützt werden), die eine vom Deutschen stark abweichendes Alphabet als Grundlage der Sortierung haben.

Weitere Probleme:

Die Umlaute **ä**, **ö**, **ü** werden nach DIN 5007 ae, oe, ue gleichgesetzt und wie diese doppelten Buchstaben hinter ad, od, ud eingeordnet (**Hadek** - **Häberle** - **Haffner**). Das gleiche gilt für den Buchstaben **ß**, der wie ss behandelt wird (**Masrich** - **Maßen** - **Mast**). Die gewünschte Anordnung kann im Programm dadurch erreicht werden, daß man die Zeichen **ä**, **ö**, **ü**, **ß** des Sortiertextes für den Sortierschlüssel in die beiden entsprechenden Zeichen austauscht, so daß im Sortierschlüssel statt **Häberle** die Schreibweise **Haeberle**, und **Massen** statt **Maßen** steht. Damit ergibt sich bei alphabetischer Sortierung die richtige Reihenfolge. Dem Programm SORTIER-VORBEREITE teilt man das Austauschen durch Parameter XS1 (EXchange im Sortierschlüssel 1) mit (XS1 /ä/ae/; XS1 /ö/oe/ etc.)

Probleme treten aber auf, wenn es - wie in dem Beispiel unten - neben einem **Jäger** auch noch einen **Jaeger** gibt, der sich mit **ae** schreibt. Nach dem Austauschen der Umlaute lauten im Sortierschlüssel beide Namen gleich **Jaeger**. Sie werden zwar beide an die richtige Stelle nach **Jader**

einsortiert, können aber in sich nicht unterschieden werden. Als Abhilfe bietet sich an, aus dem Sortiertext zusätzlich einen weiteren Sortierschlüssel aufzubauen, der zur Unterscheidung von **ae** von **ä** dient. Die gewünschte Reihenfolge ist nach DIN 5007 **ä** nach **ae**, **ö** nach **oe** etc. In SORTIER-VORBEREITE tauscht man mit Parameter XS2 (die 2 steht für den zweiten Sortierschlüssel) die Umlaute **ä**, **ö**, **ii** aus in **az**, **oz**, **uz**, entsprechend **ß** in **sz**. Diese Veränderung ist für die meisten Wörter irrelevant, da sie sich bereits im ersten Sortierschlüssel unterscheiden und damit alle weiteren Zeichen (zweiter Sortierschlüssel) keine Rolle mehr spielen: *Azalee* steht vor *Baum*, denn nachdem der erste Buchstabe die Reihenfolge festlegt, ist vollkommen uninteressant, welcher als nächstes folgt). Bei *Jaeger* und *Jäger* aber, die sich im ersten Sortierschlüssel nicht unterscheiden, wird der zweite Sortierschlüssel wichtig: es wird *Jaeger* mit *Jazger* (entstanden aus *Jäger*) verglichen, womit sich im Ergebnis die richtige Reihenfolge ergibt.

Ähnlich kann man mit den Akzenten beim Aufbauen des Sortierschlüssels verfahren. In erster Linie sollen die Akzente beim Sortieren nicht berücksichtigt werden, d. h. *é* ist wie *e* hinter *d* einzuordnen. In TUSTEP sind die Akzente als fliegende Akzente vor dem jeweiligen Buchstaben codiert (z. B. *%e*). Für den ersten Sortierschlüssel ist diese Codierung einfach ersatzlos zu streichen. Das geschieht in SORTIER-VORBEREITE mit Parameter XS1 (XS1 \%/\%). Ähnlich wie bei den Umlauten soll aber *é* und *e* bei ansonsten vollkommen gleichen Wörtern durchaus unterschieden werden; die gewünschte Reihenfolge ist *Denoix* vor *Đenoix*. Dies erreicht man wieder durch den zweiten Sortierschlüssel: die Codierung des Akzentes wird ausgetauscht in ein *z*. Sollen verschiedene Akzente in sich noch unterschieden werden, so kann man sie im zweiten Sortierschlüssel durch eine Zahl, die dem *z* folgt, unterscheiden. Man tauscht im zweiten Sortierschlüssel *é*, *è*, *ê*, *ë* aus in *z1e*, *z2e*, *z3e*, *z4e*. Verglichen wird nun im zweiten Sortierschlüssel *Denoix* mit *Dz1enoix* (entstanden aus *Denoix*) und *Dz3enoix* (entstanden aus *Đenoix*), wodurch sich die gewünschte Reihenfolge ergibt. Im ersten Sortierschlüssel lauten alle einheitlich *Denoix* und unterscheiden sich bereits hier von *Dubois*.

Ein weiteres Problem sind Zahlen. Bei gewöhnlicher alphabetischer Sortierung von links nach rechts fortschreitend spielt die erste Position immer die entscheidende Rolle; die Länge des Sortiertextes bleibt weitgehend unberücksichtigt. Das gilt auch für Zahlen: sie werden in erster Linie nach der ersten Ziffer sortiert, und die für die mathematische Reihenfolge wichtige Stellenzahl bleibt dagegen unberücksichtigt. Alphabetisch sortiert ergibt sich bei Zahlen eine Reihenfolge 7, 75, 2. Für eine mathematische Sortierung muß man die Zahlen alle auf die gleiche Stellenzahl mit führenden Nullen ergänzen. Im Programm SORTIER-VORBEREITE gibt man mit dem Parameter DEZ (DEZimalstellen) an, auf wieviele Stellen Zahlen mit führenden Nullen aufgefüllt werden sollen.

Auch römische Zahlen können nicht einfach alphabetisch sortiert werden, denn IX soll ja nach VIII kommen, was nach der alphabetischen Reihenfolge nicht der Fall ist. Die richtige Sortierung ergibt sich, wenn man römischen Zahlen für den Sortierschlüssel in (vierstellige) arabische Zahlen umwandelt. In SORTIER·VORBEREITE kann man für diesen Zweck mit Parameter R1 Zeichenfolge angeben, die in den Quelldaten unmittelbar vor dem Beginn einer römischen Zahl steht. Hinter diesen Zeichenfolgen werden die römischen Zahlzeichen I, V, X, L, C, D, M erkannt und der entsprechende Wert in arabische Zahlen umgewandelt.

Ein letztes Problem bei der Sortierung ist das Festlegen einer Reihenfolge aufgrund rein inhaltlicher Kriterien, die nichts mit dem alphanumerischen Wert der Zeichen des entsprechenden Wortes zu tun hat. In dem Beispiel unten ist das bei den Signaturen der Fall (eine andere Anwendung ist z. B. das Sortieren von Bibelstellen). Sollen die Signaturen in dem unteren Beispiel sortiert werden, so ist für die drei Bibliotheken Landesbibliothek (LB), Universitätsbibliothek (UB) und Historisches Institut (HI) eine Reihenfolge festzulegen. Dazu tauscht man in SORTIER·VORBEREITE die entsprechenden Namen in den Quelldaten in ein Zeichen oder eine Zahl aus, die dann bei der anschließenden alphanumerischen Sortierung die richtige Reihenfolge festlegt.

### Die Aufbereitung der Quelldaten

Aus dem bisher Gesagten wurde bereits klar, daß man für eine sinnvolle Sortierung auf die einzelnen Bestandteile eines Bibliographieeintrags zugreifen können muß, d. h. das Programm muß in der Lage sein, die Bestandteile *Amor*, *Titel*, *Jahr*, etc. zu erkennen. Dies gelingt aber nur, wenn jeder der Bestandteile nach eindeutigen Kriterien erkennbar ist.

In gedruckten Bibliographien sind die Bestandteile gekennzeichnet durch ihre Reihenfolge. Sie sind durch bestimmte Satzzeichen voneinander getrennt (Komma nach Autor, Punkt nach dem Titel, Doppelpunkt vor einer Zeitschrift, Klammern bei Reihen, etc.), oder auch typographisch hervorgehoben (Kapitälchen, Versalien, Fettdruck). Dem verstehenden Leser reichen diese Kennzeichnungen zur schnellen Orientierung aus. 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 ist beim Erkennen der einzelnen Teile auf *eindeutige* Kennzeichen angewiesen. Die praktizierte Unterscheidung durch Satzzeichen ist aber nicht eindeutig, denn auch innerhalb des Titels können solche vorkommen, was vom Programm nicht ohne weiteres erkannt werden kann.

Zweckmäßig ist es, für jede Kategorie (Autor, Titel, Jahr, etc.) eines Bibliographieeintrages eigens ein eindeutiges Kennzeichen in die Quelldaten einzutragen. Man wählt dafür am besten ein Sonderzeichen oder eine Zeichenkombination, die innerhalb der Daten sonst nicht vorkommt. Dieses Sonderzeichen bleibt als erstes Zeichen für die Kennung aller Kategorien konstant (im Beispiel unten ist es das Zeichen &). Die verschiedenen Kategorien werden nun durch ein weiteres Zeichen, das dem Sonderzeichen unmittelbar folgt, unterschieden. Man kann hierfür einfach laufende Nummern verwenden, oder man nimmt dafür Buchstaben, die gleichzeitig als Abkürzung für die jeweilige Kategorie dienen (a für Autor, t für Titel, j für Jahr, u für Untertitel, etc.), so daß man sich die Kennzeichen leicht merken kann. Prinzipiell ist die Wahl der Kennungen beliebig. Die hier im Beispiel verwendete Form erweist sich aber als zweckmäßig.

Einige Anmerkungen dazu:

1) Die vorgeschlagene inhaltliche Auszeichnung der Bibliographieeinträge hat mit den TUSTEP-Programmen selbst nichts zu tun. Gleichgültig mit welchem Programm man arbeitet, sollte eine solche eindeutige und differenzierte Aufbereitung der Quelldaten jedoch eine Selbstverständlichkeit sein. Der Vorteil von TUSTEP liegt darin, daß sämtliche Programmbausteine (EDITOR Abfragen, Sortieren, Registererstellung, Druckausgabe mit FORMATIERE und SATZ, algorithmischen Programmierung mit KOPIERE) diese Art der Gliederung und Auszeichnungen durch Zeichenfolgen im fließenden Text sehr komfortabel unterstützen.

2) Der Aufwand, der für Aufbereitung der Quelldaten getrieben wird, lohnt die Mühe. Denn die Kennungen werden nicht nur für die Sortierung gebraucht: Sie können genauso gut im EDITOR abgefragt werden. Es können anhand dieser Kennungen Plausibilitätstests zur Vollständigkeit und Richtigkeit der Einträge vorgenommen werden. Man ist in der Lage, beliebig lange Kommentare und Zitate in der Bibliographie selbst mitzuverwalten. Man kann die Kennungen nachher bei der gedruckten Ausgabe - sei es mit FORMATIERE in Schreibmaschinenschrift oder mit SATZ bei der professionellen Lichtsatzaufbereitung - direkt in typographische Anweisungen und Steuerzeichen umsetzen, über die man sich bei der Erfassung noch keine weiteren Gedanken machen muß.

3) Die Erfassung der Daten mit diesen Kennungen ist nicht zeitaufwendig. Man kann sich eine Eingabemaske mit den Kennungen schreiben, sie in einer Datei beliebig oft verfältigen, und springt dann beim Eintragen mit dem Cursor von Kennung zu Kennung. Das hat den weiteren Vorteil, daß man bei jedem Eintrag stets darin erinnert wird, nichts zu vergessen. Bleibt ein Teil leer, so stört es das Eintragen und die folgenden Programme nicht, es kann aber in Plausibilitätstests abgeprüft werden.

4) Falls Daten schon nach dem klassischen System der Trennung der Kategorien durch Satzzeichen erfaßt sind, kann man versuchen, sie mit dem Kommando KOPIERE automatisch umzuändern (das Programm KOPIERE bietet mehr Möglichkeiten als SORTIER VORBEREITE, Bedingungen zu prüfen und Zeichenfolgen zu ersetzen).

Zeichnet man die Kategorien auf diese Weise aus, spielt auch die Reihenfolge der Kategorien keine Rolle mehr, obwohl sie natürlich für die eigene

Orientierung zweckmäßig ist. (In den Beispieldaten unten wurde auf eine feste Reihenfolge bewußt verzichtet. Das dritte Programmbeispiel zeigt dann, wie man daraus eine feste Reihenfolge herstellt.) Wichtig ist nur, daß der Beginn eines Bibliographieeintrages eindeutig ist, damit das Programm bei der Verarbeitung einen Eintrag als eine Einheit erkennen kann (ein Eintrag muß also jeweils mit derselben Kategorie beginnen, oder er muß eigens durch eine Anfangskennung gekennzeichnet sein; im folgenden Beispiel kennzeichnet die Kategorieerkennung für den Autor **&a** den Beginn eines Eintrags).

Das Beispiel einer Bibliographie, an der im folgenden die Programme demonstriert werden, verwendet folgende Kennungen für die einzelnen Kategorien: **a** - *Autor*; **t** - *Titel*; **z** - *Zeitschrift*; **b** - *Band*; **o** - *Ort*; **J** - *Jahr*; **k** - *Kommentar*; **s** - *Signatur*. Weitere Kategorien sind leicht hinzuzufügen.

Es wurde in den Beispieldaten für die Demonstration von SORTIERVORBEREITE bewußt auf eine regelmäßige Zeileneinteilung verzichtet. Zur EDITOR-Arbeit mit den Daten wäre sie aber sinnvoll (die Beispielprogramm zeigen, wie sich eine übersichtliche Zeileneinteilung erzeugen läßt).

### Programmbeispiele für das Sortieren mit TUSTEP

Im folgenden werden einige Standardprobleme der Sortierung mit TUSTEP anhand von drei Beispielen mit den Daten der kleinen Bibliographie vorgestellt. Es werden dabei lauffähige Programme wiedergegeben, die Grundprobleme der Sortierung lösen. Der TUSTEP-Benutzer kann sie leicht für andere Quelldaten und anspruchsvollere Probleme abwandeln. Einzelheiten über die Syntax und die genaue Arbeitsweise der Kommandos und Parameter, mit denen man TUSTEP die Benutzerwünsche mitteilt, sind dem TUSTEP-Handbuch zu entnehmen. Für den, der mit TUSTEP selbst nicht arbeitet, sollen die Beispiele als Demonstration dienen, wie flexibel und doch verhältnismäßig einfach das Programm zu handhaben ist. Es darf dabei nicht vergessen werden, daß die vorgestellten Kommandosequenzen in TUSTEP dauerhaft abgespeichert und jederzeit - wenn gewünscht sogar nur mit einer Funktionstaste - wieder aufgerufen und ausgeführt werden können. Die beschriebene Arbeit braucht man sich daher in der Regel nur wenige Male zu machen, bis man sich eine »Programm-Bibliothek« seiner Standard-Lösungen erarbeitet hat.

Abbildung 1

1.1	&a Denoix, Ren%/e
1.2	&t Geschichte der Stadt Trier
1.3	&j 1980
1.4	&o München
1.5	&s UB: 10 A 6384
1.6	&k Als Kommentar kann man an dieser Stelle jeweils beliebig viel Text eingeben: eigene Anmerkungen, Zitate, Bearbeitungshinweise, etc.
2.1	&a Zenger, Zacharias
2.2	&j 1970
2.3	&b
2.4	&t Hessen und seine Wirtschaft
2.5	&o Trier
2.6	&s UB: 9 C 10387
3.1	&a Auer, Alfons &t Die Geschichte der Ethik &z &b  &j 1988
3.2	&o St. Ottilien
3.3	&s LB: Vi VIII 229
4.1	&a Jäger, Jakob
4.2	&t Leben und Werk Karls V.
4.3	&j 1985
4.4	&o Göttingen
4.5	&s LB: Vi VIII 25
5.1	&a Jaeger, Jakob
5.2	&t Württemberg, Baden und Hohenzollern
5.3	&z Historische Quartalschrift
5.4	&b 23
5.5	&j 1957
5.6	&o Koblenz
5.7	&k Auch zu diesem Eintrag wurde ein Kommentar erstellt.
5.8	&s Historisches Institut: Ze 1.09
6.1	&a Zenger, Zacharias &t Statistische Untersuchungen zur Geschichte des Saarlands &z Historische Quartalschrift &b 34 &j 1968 &o Koblenz &s HI: Ze 1.09
7.1	&a D%/enoix, Ren%/e &t Geschichte: Einführung in die Methode &j 1971 &b &z &o Köln &s Historisches Institut: Ac 35.16
8.1	&a Jäger, Gustav &s LB: Vi IX 37
8.2	&j 1987 &o Reutlingen
8.3	&t Mein Tübingen

### 1. Beispiel: Sortieren nach Autoren und Jahreszahl

Im ersten Beispiel soll die Bibliographie (vgl. Abbildung 1) sortiert werden nach Namen und Vornamen der Autoren: mehrere Werke eines Autors werden nach der Jahreszahl angeordnet. Um das Vorgehen und die Arbeitsweise des Programms SORTIER-VORBEREITE zu demonstrieren, wird das Problem in drei Versuchen gelöst, die das Ergebnis jeweils verbessern. Für einige Anwendungen bringen bereits die Teillösungen ein befriedigendes Ergebnis.

#### Erster Schritt: Sortierung nach dem gegebenen Wortlaut

Die einfachste Art der Sortierung verwendet nur den Wortlaut der Sortiereinheiten, d. h. jeder Bibliographieeintrag wird nach seinem Beginn alphabetisch sortiert. Wenn die Daten bereits eine einheitliche Form haben, bringt das bereits ein sinnvolles Ergebnis.

Wichtig ist aber auch hierbei die Festlegung der Sortiereinheiten. Schaut man in die Quelldatei (Abb. 1), stellt man fest, daß sich die Bibliographieeinträge jeweils über mehrere Sätze erstrecken. Nur ein Eintrag steht bereits in genau einem Satz (6.1) und würde von TUSTEP ohne weitere Angaben als Sortiereinheit erkannt. Man muß dem Programm den Beginn einer Sortiereinheit mitteilen; in den Beispieldaten beginnt eine Sortiereinheit jeweils mit der Kennung für den Autor &a.

Liegt die Sortiereinheit fest, braucht für eine Sortierung nach dem Wortlaut nur noch angegeben zu werden, wieviele Zeichen im Sortierschlüssel berücksichtigt werden sollen. Für die anschließende Sortierung ist eine genau definierte Anzahl von Zeichen relevant. Wird für den Sortierschlüssel ausschließlich mit Parameter SSL die Länge angegeben - im vorliegenden Fall kann man die Länge z. B. auf 50 festlegen - dann heißt dies, daß die ersten 50 Zeichen der Sortiereinheit nach dem Standardsortieralphabet als Sortierschlüssel definiert wurden, und in der Zieldatei von SORTIER-VORBEREITE vor dem eigentlichen Text auf den Positionen 1-50 jeder Sortiereinheit abgespeichert werden.

Die anschließende Sortierung mit dem Kommando SORTIERE muß dann genau nach diesen 50 Zeichen erfolgen (Spezifikation *Sortierfeld*), die nach der Sortierung ihren Zweck erfüllt haben und wieder getilgt werden können (Spezifikation *Tilgen*), so daß die Sortiereinheiten in der Zieldatei von SORTIERE in ihrem Wortlaut gegenüber der ursprünglichen Quelle unverändert sind. Der anschließende Aufruf des Kommandos KOPIERE dient dann dazu, die Satznummern wieder aufsteigend zu vergeben, damit die Datei im EDITOR bearbeitet werden kann (denn durch SORTIER-VORBEREITE und SORTIERE blieben die Sortiereinheiten unverändert, d. h. auch die ursprünglichen Satznummern haben sich noch nicht geändert).

Mit folgenden Kommandos erreicht man die gewünschte Sortierung. Die Quelldatei ist die Datei BIB (vgl. Abb. 1), das Ergebnis steht nachher in der Datei BIBSO, wobei die Daten in BIB weiterhin unverändert zugänglich sind, denn der Bearbeitung von Daten mit TUSTEP-Programmen werden die Daten stets von einer Datei in eine andere kopiert, so daß die ursprüngliche Fassung nicht verlorengeht. Bei ·STD· handelt es sich ebenfalls um eine Datei; in sie werden die Zwischenergebnisse kopiert. Sie wird von TUSTEP selbst eingerichtet und braucht den Benutzer nicht weiter zu interessieren.

### Abbildung 2

```
#sv,bib,-std,-,-,+,*
      Beginn der Sortiereinheit
aa      |&a|
      Länge des Sortierschlüssels
ssl      50
*eof
#so,-std,-,std-,1+50,+,1+50
#ko,-std-,bibso,+,+
```

In der Ergebnisdatei BIBSO stehen die Bibliographieeinträge nun in folgender Reihenfolge (angegeben wird hier zur besseren Lesbarkeit die laufende Nummer des Eintrags in der Quelldatei, sie entspricht der Satznummer in der Quelldatei): Nr. 3 - 7 - 1 - 5 - 8 - 4 - 2 - 6 . *Auer* (Nr. 3) ist erwartungsgemäß an erster Stelle; die beiden *Denoix* sind durch die Codierung des Akzentes unterschieden, \* (Nr. 7) kommt vor e (Nr. 1). Nr. 5 ist *Jaeger* mit ae. Er steht zuerst, da hier das e mit dem g der übrigen Jäger verglichen wird, da ä und a als gleichwertig behandelt werden. *Jäger, Gustav* führt die *Jäger* mit ä wegen seines Vornamens an. Die beiden *Zenger* unterscheiden sich durch die Fortführung nach dem Autorennamen: bei Nr. 2 folgt die Kategorie Jahreszahl, bei Nr. 6 die Kategorie Titel. Alphabetisch unterschieden werden die beiden Einträge durch den Kennbuchstaben j und t der Kategorien.

### Sortierung der Namen mit Berücksichtigung der Umlaute

Der erste Versuch einer Sortierung brachte zwar ein Ergebnis, das eine erste alphabetische Orientierung gibt, doch ist es noch nicht befriedigend, was die Behandlung der Umlaute und Akzente und auch die Reihenfolge gleicher Autoren betrifft.

Da die Fortführung der Bibliographieeinträge nach den Autoren nicht einheitlich ist, ist es auch nicht sinnvoll, diesen Teil in der Sortierung zu berücksichtigen, da dadurch - wie bei den beiden *Zenger* - unkontrollierte

Ergebnisse entstehen. Man kann daher zunächst einmal versuchen, die Sortierung ausschließlich auf die Autoren zu beschränken. Die Beschränkung des Sortiertextes auf die Autoren erfolgt über die Kennungen in den Quelldaten. Der Sortiertext beginnt hinter **&a** (Parameter AK1) und endet vor der Kennung für die nächste Kategorie **&** (Parameter EK1).

Bei der Sortierung der Umlaute und Akzente verfährt man so wie oben geschildert. Man braucht dafür zwei Sortierschlüssel, wobei die Umlaute in den beiden Sortierschlüssel jeweils anders ausgetauscht werden (z. B. **ä** zu **ae** bzw. **az**).

Für die beiden Sortierschlüssel ist jeweils noch ihre Länge anzugeben. Das für die Sortierung wichtige Sortierfeld setzt sich dann aus beiden Sortierschlüsseln zusammen.

Ers läßt **sich** damit folgendes Programm zusammenstellen, das die Datei BIB (vgl. Abb. 1) nun mit einer verbesserten Sortierung in die Datei BIB-SOL sortiert.

Abbildung 3

```
#sv, bib, -std-, -, •, *
      Sortiereinheit
aa      |&a|
      Sortiertext ist der Autor
ak1     |&a |
ek1     | &|
aei     11
      Austauschen Umlaute/Akzente für I.Schlüssel
xs1     |ä|ae|ö|oe|ü|ue|ß|ss|
xs1     |%/| |%/| |%»| |%:| |
xs2     |ä|az|ö|oz|ü|uz|ß|sz|
xs2     |%/|z1| |%/|z2| |%»|z3| |%:|z4|
ssl     25 25
*eof
#so, -std-, -std-, 1+50, •, 1+50
#ko, -std-, bibsol, +, •
```

In der Ergebnisdatei **stehen die** Bibliographieeinträge nun in der Reihenfolge: Nr. 3 - 1 - 7 - 8 - 5 - 4 - 2 - 6. Die beiden **Denox** (Nr. 1 und 7) stehen richtig: **die** Schreibweise mit Akzent folgt der **ohne** Akzent. Bei den **Jägers** sieht man, daß **ae** und **ä** gleichbehandelt werden. Deren Unterscheidung funktioniert **aber** nur bei den **Jakobs** (Nr. 5 vor 4), dagegen **wird Gustav** (Nr. 8) davor einsortiert. Wie ist das zu erklären? Sortiertext ist der Autor mit Name und Vorname. Er **wird** mit den angegebenen Ersetzungen in einen ersten und zweiten Sortierschlüssel übernommen. Verglichen **wird** der Sortierschlüssel, der bei den **drei Jägern** lautet:

Abbildung 4

jaeger, gustav	jazger, gustav
jaeger, jakob	jaeger, jakob
jaeger, jakob	jazger, jakob

Bevor die Zeichen des zweiten Sortierschlüssels, der für die Unterscheidung ä zu ae verantwortlich ist, verglichen werden, wird der Vorname relevant. Daher wird *Gustav* vor *Jakob* sortiert, unabhängig davon, daß er gemäß dem zweiten Sortierschlüssel dahinter kommen sollte.

Wird gewünscht, daß zunächst ausschließlich nach den Nachnamen sortiert wird, und erst bei vollkommener Identität der Nachnamen die Vornamen herangezogen werden, so muß man die Lösung wie im folgenden Versuch abändern. Anzustreben ist folgender Sortierschlüssel:

Abbildung 5

jaeger	jazger	gustav
jaeger	jaeger	jakob
jaeger	jazger	jakob

Man sieht, daß bei diesem Sortierschlüssel die ersten beiden Namen bei der Sortierung ihre Plätze tauschen.

Ein weiteres Problem hat der aktuelle Lösungsversuch noch nicht bewältigt: es ist keine Unterscheidung innerhalb der Werke eines Autors vorgesehen. Nr. 2 und Nr. 6, die beiden Werke von *Zenger*, stehen im Ergebnis noch in der Reihenfolge der Quelldatci. Als Kriterium zu ihrer Unterscheidung soll laut Aufgabenstellung die Jahreszahl herangezogen werden.

Und eine Ergänzung: so wie die Kommandofolge jetzt eingerichtet ist, steht in der Zieldatei eine Sortiereinheit in einem Satz (das ist die einzige Veränderung der Daten gegenüber der Quelldatei). U. u. ist es Übersichtlicher, wenn jeweils eine Kategorie eines Bibliographieeintrages in einer eigenen Zeile steht, und ein Bibliographieeintrag jeweils die gleiche Seitennummer erhält. Auch das läßt sich im dritten Versuch erreichen.

### Sortierung nach Name, Vorname und Jahreszahl

Bei diesem Lösungsversuch werden nun alle Wünsche berücksichtigt.

Als Sortiertext wird wieder der Autor (&a bis &) isoliert. Zusätzlich wird als Sortiertext die Jahreszahl (&j bis &) genommen. Als erstes Sortierkriterium werden die Namen herangezogen, wobei die Umlaute in die entsprechenden doppelten Buchstaben ausgetauscht werden. Als zweites Kriterium dient der Name, wobei die Umlaute in die entsprechende Kombination mit z ausgetauscht werden. Erst dann werden die Vornamen und danach die Jahreszahlen berücksichtigt.

Erreichen kann man die Anordnung, indem man für den Sortierschlüssel aus dem Sortiertext mit den Parametern AS1, ESI, AS2, ES2 und AS3, ES3 (Anfang bzw. Ende des Sortierschlüssels 1-3) bestimmte Teile auswählt. Der erste Sortierschlüssel wird auf den Namen beschränkt; die Austauschungen für die Umlaute werden dann nur hier vorgenommen. Im zweiten Sortierschlüssel wird alles übernommen. Da es bei den Vornamen keine Unterscheidungen durch die Umlaute gibt, stören auch die Austauschungen nicht.

Schrittweise hat man sich die Auswahl so vorzustellen:  
Aus der gesamten Sortiereinheit wird der Sortiertext erstellt, z. B.:

*Jäger, Jakob 1985*

Daraus erstellt man die Sortierschlüssel. Der erste Sortierschlüssel wird auf den Namen beschränkt (Ende ist das Komma). Hier werden die Umlaute ausgetauscht. Erster Sortierschlüssel ist:

*jaeger*

Als zweiter Sortierschlüssel dient der gesamte Sortiertext. Hier werden die Umlaute in die entsprechende Kombination mit z ausgetauscht. Zweiter Sortierschlüssel ist:

*jazger, jakob 1985*

Die beiden Sortierschlüssel bilden dann zusammen das für die Sortierung relevante Sortierfeld. Es enthält folgenden Text:

*jaeger jazger, jakob 1985*

Der Aufbau der Sortierschlüssel läßt sich kontrollieren, indem man bei SORTIER VORBEREITE ein Testprotokoll ausgibt. Damit kann man zur Fehlersuche die Bearbeitung jeder Sortiereinheit nachvollziehen.

Das Programm für den dritten Versuch der Problemlösung ist in Abbildung 6 dargestellt.

Das Programm liefert das gewünschte Ergebnis. In der Zieldatei BIBS03 stehen folgende Daten (vgl. Abbildung 7):

Einige weiterführende Anmerkungen:

Das vorgestellte Programm macht klar, wie kompliziertere Sortieraufgaben zu lösen sind, z. B. die Behandlung von körperschaftlichen Autoren, die Behandlung von Herausgebern, etc. Man muß dafür bei der Erfassung die Kategorien

Abbildung 6

```
|sv,bib,-std-,-,«,*,*
aa      |&a|
ak1     |&a |
*k1     | A|
*k2     |*j |
*k2     | A|
*ei     11 11
«S1     |, |
xs1     |'a|ae|ö|oe|ü|ue|ß|ss|
XS1     |%|/||%>>||%:|{|%|}|
xs2     |'a|az|Ö|OZ|Ü|U2|ß|S2|
xs2     |%|z11%| |z2|%»|z3|%: |z4|
ssl     15 35
*eof
#so,-std-,-std-,1*50,*#1*50
#ko,-std-,bibso3,*,*,*
za      1&l
sa      |aa|
*aof
```

weiter differenzieren, und bei der Erstellung des Sortiertextes differenziert darauf eingehen. Falls dazu SORTIER-VORBEREITE nicht ausreicht, kann man einen KOPIERE-Aufruf mit seinen erweiterten Möglichkeiten der Bearbeitung vorschallen.

So z. B. beim Erzeugen von Verweisen bei mehreren Autoren eines Werkes. Durch einen KOPIERE-Aufruf vervielfältigt man die entsprechenden Bibliographieeinträge, gibt sie als Verweis oder vollständig mit wechselndem ersten Autor aus. Jede dieser Vervielfältigungen bildet dann eine eigene Sortiereinheit, die durch die Sortierung ihren bestimmten Platz erhält.

## 2. Beispiel: Sortieren nach den Signaturen

Im zweiten Beispiel wird gezeigt, wie es möglich ist, eine Sortierung nach Kriterien vorzunehmen, die nicht der alphanumerischen Reihenfolge des für die Sortierung ausgewählten Textes entspricht.

Die Einträge der kleinen Beispielbibliographie sollen nach den Signaturen sortiert werden. Dabei sollen die Signaturen in erster Linie nach den drei Bibliotheken angeordnet werden: an erster Stelle die Signaturen der Landesbibliothek (LB), danach die Universitätsbibliothek (UB), und zum Schluß die Bücher aus dem Historischen Institut (HI).

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 Signaturesystem einheitlich ist. Es ist lediglich darauf zu achten, daß die arabischen Zahlen alle auf eine einheitliche Stellenzahl gebracht werden.

Eine weitere Besonderheit bilden die römischen Zahlen in der Signatur der Landesbibliothek. Um diese für die Sortierung in arabische Zahlen überführen zu können (Parameter R1) ist es notwendig, daß vor der römischen Zahl eine eindeutige Kennung steht, die selbst für die Sortierung nicht relevant ist. Man kann eine solche Kennung beim Erstellen des

Abbildung 7

1.1	&a Auer, Alfons
1.2	&t Die Geschichte der Ethik
1.3	&z
1.4	&b .
1.5	&j 1988
1.6	&o St. Ottilien
1.7	&s LB: Vi VIII 229
2.1	&a Denoix, Ren%/e
2.2	&t Geschichte der Stadt Trier
2.3	&j 1980
2.4	&o München
2.5	&s UB: 10 A 6384
2.6	&k Als Kommentar kann man an dieser Stelle jeweils beliebig viel Text eingeben: eigene Anmerkungen, Zitate, Bearbeitungshinweise, etc.
3.1	&a D%/enoix, Ren%/e
3.2	&t Geschichte: Einführung in die Methode
3.3	&j 1971
3.4	&b
3.5	&z
3.6	&o Köln
3.7	&s Historisches Institut: Ac 35.16
4.1	&a Jaeger, Jakob
4.2	&t Württemberg, Baden und Hohenzollern
4.3	&z Historische Quartalschrift
4.4	&b 23
4.5	&j 1957
4.6	&o Koblenz
4.7	&k Auch zu diesem Eintrag wurde ein Kommentar erstellt.
4.8	&s Historisches Institut: Ze 1.09
5.1	&a Jäger, Gustav
5.2	&s LB: Vi IX 37
5.3	&j 1987
5.4	&o Reutlingen
5.5	&t Mein Tübingen
6.1	&a Jäger, Jakob
6.2	&t Leben und Werk Karls V.
6.3	&j 1985
6.4	&o Göttingen
6.5	&s LB: Vi VIII 25
7.1	&a Zenger, Zacharias
7.2	&t Statistische Untersuchungen zur Geschichte des Saarlands
7.3	&z Historische Quartalschrift
7.4	&b 34
7.5	&j 1968
7.6	&o Koblenz
7.7	&s HI: Ze 1.09
8.1	&a Zenger, Zacharias
8.2	&j 1970
8.3	&b
8.4	&t Hessen und seine Wirtschaft
8.5	&o Trier
8.6	&s. UB: 9 C 10387

Sortiertextes einfügen, wenn - wie in dem vorliegenden Beispiel - die römischen Zahlen durch ihre Umgebung eindeutig zu erkennen sind.

Und noch eine zweite Besonderheit: Zeitschriften stehen in dem Beispiel jeweils unter einer Signatur (vgl. Bibliographieeintrag Nr. 5 und 6). Sie werden in sich durch ihre Bandnummer unterschieden, so daß zur Sortierung auch die Bandnummer heranziehen ist.

Der Sortiertext setzt sich damit zusammen aus der Signatur (&s bis &) und der Bandangabe (&b bis &). Eine fehlende Bandangabe beeinträchtigt die Sortierung nicht: der Sortiertext ist dann lediglich kürzer.

Damit läßt sich das Sortierprogramm zusammenstellen. Die Sortiereinheiten beginnen bei &a.

Abbildung 8

```
#sv,bib,-std-,-,+,* ,pr=-std-
Sortiereinheit
aa      |&a|
        Sortiertext ist Signatur (1) und Band (2)
ak1     |&s |
ek1     | & |
ak2     |&b |
ek2     | & |
aei     11 11
        Auszeichnen von römischen Zahlen mit Kennung #
>1z     ivxlcdm
xx1     |>* >1|>=01 #<=01|
        Reihenfolge der Bibliotheken
xx1     |lb:|1|
xx1     |ub:|2|
xx1     |historisches institut:|3|
xx1     |hi:|3|
        Nach # römische in arabische Zahl umwandeln
r1      |#|
        Einheitliche Stellenzahl für arabische Zahlen
dez     5
        Länge des Sortierschlüssels
ssl     30
*eof
#so,-std-,-std-,1+30,+,1+30
#ko,-std-,signatur,+,+,*
za      |&|
sa      |&a|
*eof
```

Greifen wir die dritte Sortiereinheit (Auer) zur Verdeutlichung des Programmablaufs heraus:

Der Sortiertext beginnt nach &s und endet vor dem nächsten &, das aber in dieser Sortiereinheit nicht gefunden wird, da die Signatur die letzte Kategorie bildet. Dies stört nicht, denn wird keine Endekennung gefun-

den, 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 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 unterscheidet sich der zweite Teil bei der Sortierung vom ersten Teil. Der ausgewählte Sortiertext ist:

**LB: Vi VIII 229**

wobei abschließend zwei Blanks stehen. In dem ausgewählten Sortiertext werden nun Zeichenfolgen ausgetauscht, **LB:** wird zu **1** und 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 dann sortiert wird, lautet:

**00001 Vi 0008 00229**

Dies liefert das gewünschte Ergebnis. (Die Reihenfolge der Sortiereinheiten nach dieser Sortierung ist unten bei der Wiedergabe des Ergebnisses des nächsten Beispiels aufgeführt, denn dort wird das vorliegende Ergebnis als Quelle verwendet, und die Bibliographieeinträge werden in ihrer Reihenfolge nicht mehr verändert.)

### 3. Beispiel: Sortieren der Kategorien in jedem Bibliographieeintrag

Für eine weitere Bearbeitung der Bibliographie (Abfragen im EDITOR, Übernahme in ein Datenbanksystem, Ausgeben auf Papier), aber auch einfach zur besseren Orientierung bei der laufenden Arbeit ist es sinnvoll, die Kategorien innerhalb eines Bibliographieeintrags jeweils in derselben Reihenfolge anzuordnen. Die Umgestaltung der jetzigen Quelldatei in eine solche Form läßt sich ebenfalls durch Sortieren erreichen. Für das Beispiel sollen die Kategorien in die Reihenfolge **Signatur - Band - Autor - Jahr - Titel - Ort - Zeitschrift - Kommentar** gebracht werden.

Eine Sortiereinheit wird bei dieser Aufgabe nicht vom ganzen Bibliographieeintrag gebildet, sondern jede einzelne Kategorie bildet eine Sortiereinheit. Dazu darf aber in einem TUSTEP-Satz nicht mehr als eine Kategorie stehen. Dies ist z.B. in den Ergebnisdateien der vorausgehenden Beispiele der Fall. (Man kann auch die Quelldatei durch einen Aufruf des Kommandos KOPIERE entsprechend vorbereiten.)

Steht jede Kategorie in genau einem Satz, so braucht man sich bei SORTIER VORBEREITE um die Sortiereinheiten nicht mehr zu kümmern. Man kann die Kategorien nun nach ihren Kennungen - d. h. nach den Kennbuchstaben - sortieren. Allerdings entspricht die alphabetische Reihenfolge der Kennbuchstaben nicht der gewünschten Reihenfolge. Man muß die gewünschte Reihenfolge eigens festlegen. Eine Lösung des Problems wäre, wie oben die Kennungen in Zahlen auszutauschen, die dann die Reihenfolge festlegen.

Man kann aber auch einen anderen Weg gehen. Da die Unterscheidung genau definiert jeweils nur durch ein Zeichen stattfindet, kann man für diese Zeichen ein eigenes Sortieralphabet definieren. Man legt durch das Sortieralphabet die Wertigkeit der Kennbuchstaben fest. Werden die Kennbuchstaben nach dem Sortieralphabet **sbajtozk** sortiert, so entsteht die gewünschte Reihenfolge der Kategorien.

Würde man das Programm so starten, wäre aber das Ergebnis völlig unbrauchbar: zu Beginn stünden alle Signaturen, dann folgen alle Bandangaben, dann alle Autoren, d. h. die Bibliographieeinträge wären auseinandergerissen, denn man hatte 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) nur in sich sortiert werden. Dazu gibt man daher in SORTIER VORBEREITE an, daß jeweils bei **&a** 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 an den Anfang des Sortierfeldes geschrieben wird, so daß sie beim Sortieren das erste Kriterium bildet. Der aufgebaute Sortierschlüssel wird dann erst als zweites Kriterium herangezogen. Dadurch bleiben die Kategorien eines Bibliographieeintrags zusammen, werden aber in sich richtig sortiert.

Das Programm, das die Ergebnisdatei SIGNATUR aus Beispiel 2) in der gewünschten Form sortiert, sieht folgendermaßen aus:

**Abbildung 9**

```
#sv,signatur, -std,-,+,*
      Sortiergruppe
nsn   |&a|
snl   3
      Sortiertext ist der Kennbuchstabe
ak1   |&|
ek1   | |
aei   11
```

```

Sortieralphabet
a1          sbajtozk
ssl        1
*eof
#so,-std-, -std-, 1+4, +, 1+4
#ko,-std-, gleich, +, +
    
```

Jede einzelne Kategorie eines Bibliographieeintrags bildet eine Sortiereinheit, die bereits in einem TUSTEPSatz steht. Jeder dieser Sätze beginnt mit einer Kategorie Kennung. Als Sortiertext, der dann für die Stellung eines jeden Satzes verantwortlich ist, wird ausschließlich der Kennbuchstabe der Kategorie herangezogen, der nach dem eigens definierten Sortieralphabet sortiert wird. Die Sortiernummer wird vor dem jeweiligen Sortierschlüssel ergänzt.

Die Sortiereinheit *&o St. Ottilien* (Bibliographieeintrag *Alfons Auer*) hat z. B. als Sortiertext *o*, ihre Sortiernummer ist 2 (es handelt sich fortlaufend gezählt um die zweite Sortiergruppe). Sortiert wird nach dem Sortierfeld *002o* (die Sortiernummer wurde auf drei Stellen festgelegt; intern wird das *o* durch die Angabe des eigenen Sortieralphabets in einen entsprechend ASCII-Wert ausgetauscht, damit die Sortierung stimmt).

Als Ergebnis erhält man in der Datei GLEICH folgende Daten, die in der Reihenfolge der Bibliographieeinträge das Ergebnis aus dem zweiten Beispiel wiedergeben:

Einige weiterführende Bemerkungen:

U. u. kann es sinnvoll sein, je Bibliographieeintrag die Kategorien nicht nur in einer festen Reihenfolge zu haben, sondern auch vollständig (d. h. jeweils alle Kategoriekennungen sollen vorkommen). Auch das kann man mit dem skizzierten Beispielprogramm leicht erreichen. Man braucht dazu nur einen vorbereitenden Aufruf des Kommandos KOPIERE die fehlenden Kennungen zu ergänzen. Dabei prüft man je Bibliographieeintrag das Vorhandensein jeder einzelnen Kategoriekennung ab, und ergänzt sie gegebenenfalls. Anschließend, nachdem auch die neuen Kennungen in einen je eigenen TUSTEPSatz aufgeteilt sind, startet man das obige Programm.

Mit den drei Beispielen wurde ein Teil der Leistungen der Sortierprogramme von TUSTEP gezeigt. Die vorgestellte Aufbereitung der Quelldaten ist lediglich als ein Vorschlag zu sehen und stellt keine vom Programm vorgegebene Notwendigkeit dar. Die Quelldaten müssen sich nicht nach einem durch die Programme fest vorgegebenen Schema richten, sondern umgekehrt: TUSTEP-Programme lassen sich über Parameter den jeweiligen Quelldaten und ihren speziellen Bedingungen anpassen; u. U. ist dafür die Verwendung von weiteren Programmbausteinen nötig, was aber bei dem modularen Aufbau von TUSTEP keinerlei Schwierigkeiten bereitet.

Abbildung 10

1.1	&s LB: Vi VIII 25
1.2	&a Jäger, Jakob
1.3	&j 1985
1.4	&t Leben und Werk Karls V.
1.5	&o Göttingen
1.6	&s LB: Vi VIII 229
1.7	&b
1.8	&a Auer, Alfons
1.9	&j 1988
1.10	&t Die Geschichte der Ethik
1.11	&o St. Ottilien
1.12	&z
1.13	&s LB: Vi IX 37
1.14	&a Jäger, Gustav
1.15	&j 1987
1.16	&t Mein Tübingen
1.17	&o Reutlingen
1.18	&s UB: 9 C 10387
1.19	&b
1.20	&a Zenger, Zacharias
1.21	&j 1970
1.22	&t Hessen und seine Wirtschaft
1.23	&o Trier
1.24	&s UB: 10 A 6384
1.25	&a Denoix, Ren%/e
1.26	&j 1980
1.27	&t Geschichte der Stadt Trier
1.28	&o München
1.29	&k Als Kommentar kann man an dieser Stelle jeweils beliebig viel Text eingeben: eigene Anmerkungen, Zitate, Bearbeitungshinweise, etc.
1.30	&s Historisches Institut: Ac 35.16
1.31	&b
1.32	&a D%/enoix, Ren%/e
1.33	&j 1971
1.34	&t Geschichte: Einführung in die Methode
1.35	&o Köln
1.36	&z
1.37	&s Historisches Institut: Ze 1.09
1.38	&b 23
1.39	&a Jaeger, Jakob
1.40	&j 1957
1.41	&t Württemberg, Baden und Hohenzollern
1.42	&o Koblenz
1.43	&z Historische Quartalschrift
1.44	&k Auch zu diesem Eintrag wurde ein Kommentar erstellt.
1.45	&s HI: Ze 1.09
1.46	&b 34
1.47	&a Zenger, Zacharias
1.48	&j 1968
1.49	&t Statistische Untersuchungen zur Geschichte des Saarlands
1.50	&o Koblenz
1.51	&z Historische Quartalschrift