

Neues zum Thema Sortieren

In Listing 1 sehen Sie die Quicksort-Version von Kurt Sörensen, die ein Feld von 100 Elementen in nur noch 20 Sekunden sortiert, was eine Verkürzung der Sortierzeit um knapp 35 Prozent zur letzten Quicksort-Version bedeutet. Eine solche Sortierzeit ist natürlich von Programmen, wie Mischsort oder Supersort nicht mehr zu schlagen, weshalb wir an dieser Stelle nochmals ausdrücklich die Behauptungen in früheren Heften redigieren müssen, es gäbe ein schnelleres Sortierprogramm als Quicksort (Programme für Spezialanwendungen dürfen in diesem Wettbewerb natürlich nicht antreten; hier zählt einzig und allein der Grundgedanke eines Sortieralgorithmus, der ohne Zusatz oder »Tuning« geprüft wird).

Nun zu einigen speziellen Problemen, die vielfach aufgetreten sind. Wir wollen an dieser Stelle einmal einige Antworten und Lösungsvorschläge geben.

Das Problem mit mehrdimensionalen Feldern

Zuerst zum Problem des Sortierens mehrdimensionaler Felder. Hier kamen einige Anfragen, die sich generell mit diesem Problem beschäftigten. An dieser Stelle muß dazu gesagt werden, daß es eine allgemeingültige Lösung nicht gibt; und zwar aus folgendem Grund: Ein mehrdimensionales Feld hat computerintern eine gewisse Reihenfolge, in der die Elemente abgelegt werden (siehe Bild 1). Nach dieser Reihenfolge könnte man ohne weiteres sortieren. Mehrdimensionale Felder werden jedoch in der Regel hierarchisch nach ganz anderen Gesichtspunkten sortiert, wobei in jeder Dimension des Feldes eine andere Art von Daten abgelegt werden (siehe Bild 2).

Wollten Sie ein solches Feld, wie in Bild 2 dargestellt, sortieren, so bräuchten Sie für jede Anwendung ein Spezialsortierprogramm, das auf die ganz bestimmten Eigenheiten Ihres Feldes zugeschnitten ist. Im Prinzip ist es in so einem Fall einfacher, das gesamte Datenfeld in eindimensionale Felder zu zerlegen,

Die Krönung unseres Sortierkurses. Wir stellen Ihnen heute zwei optimierte Quicksort-Routinen in Basic und in Maschinsprache vor.

Bild 3. Würde man die Kartei von Bild 2 nach dem computer-internen Standard der Felddanlage sortieren, so könnte zum Beispiel folgendes passieren:	DIM A\$(4,4,4)	
	A\$(1,1,1)	Adresse 3
	A\$(1,1,2)	Adresse 1
	A\$(1,1,3)	Name 4
	A\$(1,1,4)	Bestellung 128
	A\$(1,2,1)	Adresse 2
A\$(1,2,2)	Telefonnummer 3	
...		

DIM A\$(4,4,4)		
1. Eintrag:	A\$(1,1,1):	Name
	A\$(1,1,2):	Adresse
	A\$(1,1,3):	Telefonnummer
	A\$(1,1,4):	Kundennummer
	...	
2. Eintrag:	A\$(2,1,1):	Name
	A\$(2,1,2):	Adresse
...		
4. Eintrag	A\$(4,1,1):	Name
...		
	A\$(4,4,4):	Bestellung 128

Bild 2. So werden Feldelemente gerne vom Benutzer angeordnet, um einen einfachen Zugriff zu erreichen

```

10000 REM SUPER-QUICKSORT <143>
10010 DIM LG(100),RG(100):TI$="000000":LG( <123>
1) = 1:RG(1) = A:Z = 0:GOSUB 10012 <101>
10011 GOTO 50000 <101>
10012 Z = Z + 1: IF LG(Z) >= RG(Z) THEN 10025 <127>
10013 X = LG(Z): Y = RG(Z): IF Y <= X + 1 THEN 10021 <010>
10014 B = INT(X + Y) / 2: VG$ = A$(B) <101>
10015 IF X > Y THEN 10023 <021>
10016 IF A$(X) < VG$ THEN X = X + 1: GOTO 10016 <207>
10017 IF A$(Y) > VG$ THEN Y = Y - 1: GOTO 10017 <165>
10018 IF X > Y THEN 10023 <024>
10019 S$ = A$(X): A$(X) = A$(Y): A$(Y) = S$ <137>
10020 X = X + 1: Y = Y - 1: GOTO 10015 <124>
10021 IF A$(X) <= A$(Y) THEN 10025 <217>
10022 S$ = A$(X): A$(X) = A$(Y): A$(Y) = S$: GOTO 1 <158>
0025 <160>
10023 RG(Z + 1) = Y: LG(Z + 1) = LG(Z): GOSUB 10012 <160>
10024 LG(Z + 1) = RG(Z + 1) + 1: RG(Z + 1) = RG(Z): GOSU <186>
B 10012 <134>
10025 Z = Z - 1: RETURN
    
```

Listing 1. Das »Super-Quicksort«. Dieses Programm ist um zirka 35 Prozent schneller als sein Vorgänger.

und diese entweder einzeln oder in Abhängigkeit eines jeweils anderen Feldes zu sortieren. So etwas beherrscht »Exsort« aus der 64'er, Ausgabe 11, 1984. Eine Sortierung des mehrdimensionalen Feldes aus Bild 2, mit den Methoden, die in unserem Kurs vorgestellt wurden, könnte nur nach dem Gesichtspunkt der computerinternen Ordnung (Bild 1) geschehen und würde Ihnen ein Chaos, wie Sie es in Bild 3 sehen können, hinterlassen.

Vermeiden Sie also prinzipiell die Anwendung mehrdimensionaler Felder, wenn Sie Sortierungen vornehmen wollen. Jedes mehrdimensionale Feld läßt sich nämlich auch durch entsprechende eindimensionale Felder ersetzen.

Die Anwendungen der Sortierprogramme

Unter vielen Anfragen zum Sortierkurs erreichten uns auch einige, die sich mit der Anwendung der Sortieralgorithmen selbst befaßten. Hier war oft unklar, wie die gezeigten Programme denn nun angewendet werden können. Nun, dieses Problem kann sehr einfach gelöst werden.

DIM A\$(4,4,4)	
A\$(1,1,1)	
A\$(1,1,2)	
A\$(1,1,3)	
A\$(1,1,4)	
A\$(1,2,1)	
A\$(1,2,2)	
A\$(1,2,3)	
A\$(1,2,4)	
A\$(1,3,1)	
...	
A\$(1,4,4)	
A\$(2,1,1)	
A\$(2,1,3)	
A\$(2,1,4)	
A\$(2,2,1)	
...	
A\$(2,4,4)	
A\$(4,4,4)	

Bild 1. So sind dreidimensionale Felder im Computer angeordnet

Vielleicht ist Ihnen aufgefallen, daß alle abgedruckten Sortieralgorithmen die Zeilennummern 10000 bis höchstens 19999 besaßen. Diese Zeilennummern waren so gehalten, daß man die eingetippten Sortieralgorithmen sofort mit dem Hauptprogramm, das einige Male mit abgedruckt wurde, testen konnte. Dieses Hauptprogramm hatte nämlich bei den Zeilennummern 10000 bis 19999 keine Programmzeile und so mußte das Sortierprogramm lediglich dort eingefügt werden.

Wollen Sie nun einen Sortieralgorithmus verwenden, weil er Ihnen zusagt, so müssen Sie einfach das Hauptprogramm weglassen und sich die Variablen ansehen, unter denen das Feld steht. Bei den Sortieralgorithmen war das zu sortierende Feld stets unter A\$ und die Größe des A\$-Feldes unter A gespeichert. Verwenden Sie andere Variablenamen, so müssen diese nur im Sortierprogramm entsprechend angepaßt werden,

und schon läuft die ganze Sache. Vorsicht auch bei Variablen, die das Sortierprogramm verwendet und die unter Umständen mit Variablen in Ihrem Programm »kollidieren«. Die Sortieralgorithmen sind ja in der Regel sehr kurz, so daß Änderungen schnell gemacht sind.

Und immer wieder Quicksort

Noch eine Bemerkung zu Quicksort in Maschinensprache, das ebenfalls in Ausgabe 12/85 abgedruckt wurde. Es wurde damals schon erwähnt, daß es bei diesem Programm unbedingt notwendig ist, die zu sortierenden Daten in einem Stringarray (zum Beispiel A\$,V\$, oder ähnliche) abzulegen und dieses Feld als erstes im Basic-Programm zu dimensionieren. Wird das nicht gemacht, so kann es passieren, daß Quicksort-M die Variablenorganisation des Computers durcheinanderbringt, wobei ein »Aussteigen« des

Computers die Folge ist. Wollen Sie das Quicksort-Programm als eigenes Maschinenprogramm stehen lassen, so müssen Sie es jeweils vor (!) dem Basic-Programm laden und danach »NEW« eintippen, um die Basic-Zeiger im Computer wieder herzustellen.

Natürlich ist kein Programm absolut perfekt, und so finden sich immer wieder Leser, die ihre Ansprüche ein wenig höher stecken. Die Ergebnisse zeigen sich dann in der Regel in einer stark verbesserten Version von abgedruckten Programmen oder in Neuentwicklungen von Problemlösungen.

So haben wir auch in diesem Artikel ein Bonbon für Sie vorbereitet. Es handelt sich, wie sollte es auch anders sein, um ein neues Quicksort in Maschinensprache, das uns Kurt Sörensen aus Hamburg zugesandt hat (Listing 2 und 3). Dieses Quicksort ist noch einmal mehr als viermal so schnell als unser Quicksort-M. Durch seine hohe Geschwindig-

keit dürfte es wohl eines der schnellsten Sortierprogramme sein, die es für einen 6502-Mikroprozessor gibt. Zusätzlich zu diesen Vorteilen kommt noch dazu, daß das neue Quicksort-M um einiges kürzer ist als sein Vorgänger und selbst bei sehr komplizierten Datenfeldern keinen Stack-Überlauf mehr verursacht.

Auch bei dem neuen Quicksort in Maschinensprache ist das zu sortierende Stringfeld stets das, das als erstes in einem Programm dimensioniert wurde. Das Starten des Sortiervorgangs erfolgt dabei mit »SYS 50000«.

Das neue Quicksort-M dürfte wohl den allermeisten Anwendungen genügen. Wenn es jedoch jemanden gibt, der immer noch nicht das optimale Sortierprogramm gefunden hat, dem bleibt dann bloß noch die Eigeninitiative (vielleicht können wir irgendwann einmal das perfekte Sortierprogramm als »Listing des Monats« begrüßen?).

(K. Sörensen/ks)

```

10 -;      --- QUICKSORT.ASS ---
11 -;
900 -;      VARIABLEN IN DER ZERO PAGE
901 -;
902 -.BA 50 000 ; START ADRESSE
903 -.EQ XADR=10 ; ADR. A$(X) IN (10,11)
904 -.EQ YADR=12 ; ADR. A$(Y) IN (12,13)
905 -.EQ VADR=14 ; ADR. A$(V) IN (14,15)
906 -.EQ XDES=16 ; DESCR. A$(X) IN (16,17,18)
907 -.EQ YDES=19 ; DESCR. A$(Y) IN (19,20,21)
908 -.EQ VDES=22 ; DESCR. A$(V) IN (22,23,24)
909 -.EQ LG = 25 ; LINKE GRENZE IN (25,26)
910 -.EQ RG = 27 ; RECHTE GRENZE IN (27,28)
911 -.EQ VERL=29 ; VERGLEICHS LAENGEREADY.
912 -.EQ STIN=252 ; STACK EINGANGSWERT
913 -.EQ STMI=253 ; STACK MINIMUM
914 -.EQ STUG=254 ; STACK UNTERGRENZE
915 -;
1000 -; ----- BASIC-DATEN RETTEN
1100 -;STACK-POINTER -----
1110 -      TSX
1120 -      STX STIN
1125 -      STX STMI
1130 -      TXA ; STACK -
1140 -      SEC ; UNTERGRENZE
1150 -      SBC #82 ; BERECHNEN
1160 -      BCS SPR
1170 -      RTS ; ABRUCH
1180 -SPR ADC #2 ; 3 BYTE
1190 -      STA STUG ; UEBER 0
1195 -;
1200 -; ZERO-PAGE -----
1210 -      LDX #19
1220 -ZPWEG LDA 10,X
1230 -      PHA
1240 -      DEX
1250 -      BPL ZPWEG
1260 -      TSX
1270 -      STX STIN
2000 -; ----- ANFANGS - BEDINGUNGEN
2100 -; Z = 0 -----
2110 -SORT LDA #0
2120 -      PHA
2130 -      PHA
2200 -; LG = 1 -----
2210 -      CLC
2220 -      LDA 47 ; FELDFANFANG
2230 -      ADC #10 ; IN (47,48)
2240 -      STA LG ; + 10 BYTE
2250 -      LDA 48 ; VORSPANN
2260 -      ADC #0 ; ERGIBT
2270 -      STA LG+1 ; ADR A$(1)
2300 -; RG = A -----
2310 -      CLC ; FELDLAENGE
2320 -      LDY #2 ; STEHT IM
2330 -      LDA (47),Y ; VORSPANN
2340 -      ADC 47 ; PLUS
2350 -      TAX ; FELDFANFANG
2360 -      INY
2370 -      LDA (47),Y
2380 -      ADC 48
2390 -      TAY
2400 -      SEC
2410 -      TXA ; MINUS
2420 -      SBC #3 ; 3 BYTE
2430 -      STA RG ; FUER LETZTEN
2440 -      PHA ; DESCRIPTOR
2450 -      TYA ; ERGIBT
2460 -      SBC #0 ; ADR A$(A)
2470 -      STA RG+1
2480 -      PHA
2490 -      BNE VSTRING
2500 -; Z = 1 -----
2510 -; BEDEUTET RG(1) AUF STACK LEGEN
2520 -; SIEHE ZEILE 2440 UND 2480
2530 -BRUECKE4 BNE SORT
2540 -;
3000 -;----- VERGLEICHSTRING BERECHNEN
3100 -;X=LG:Y=RG -----
3110 -VSTRING LDX #3
3120 -LADXY LDA LG,X
3130 -      STA XADR,X ; LADE -
3140 -      DEX ; SCHLEIFE
3150 -      BPL LADXY
3200 -;V = X + Y -----
3210 -      CLC
3220 -      LDA XADR
3230 -      ADC YADR
3240 -      TAX
3250 -      LDA XADR+1
3260 -      ADC YADR+1
3300 -;V = INT(V/2) -----
3310 -      LSR ; HIGH BYTE
3320 -      STA VADR+1 ; RECHTS SHIFT
3330 -      TXA ; LOW BYTE
3340 -      ROR ; ROTATION
3350 -      BCC SPR1 ; RECHTS
3360 -      SBC #1 ; INT -
3370 -      BCS SPR1 ; MODULO
3380 -      DEC VADR+1 ; 3
3390 -SPR1 STA VADR
3400 -;V$ = A$(V) -----

```

Listing 2. Das Assembler-Listing der neuen Version von Quicksort in Maschinensprache. Die Sortierung beginnt mit »SYS 50000«.

```

3410 - LDY #0 ; DESCRIPTOR
3420 - LDA (VADR),Y ; A$(V)
3430 - STA VDES ; IN DER
3440 - INY ; ZERO PAGE
3450 - LDA (VADR),Y ; SPEICHERN
3460 - STA VDES+1
3470 - INY
3480 - LDA (VADR),Y
3490 - STA VDES+2
3495 -;
4000 -; ----- X - VERGLEICHSSCHLEIFE
4100 -; X$ = A$(X) -----
4110 -VERGLX LDY #0 ; DESCRIPTOR
4120 - LDA (XADR),Y ; A$(X)
4130 - STA XDES ; IN DER
4140 - INY ; ZERO PAGE
4150 - LDA (XADR),Y ; SPEICHERN
4160 - STA XDES+1
4170 - INY
4180 - LDA (XADR),Y
4190 - STA XDES+2
4200 -; VERGLEICHS-LAENGE -----
4210 - LDX #0 ; X$ KUERZER:
4220 - LDA XDES ; XREG=0
4230 - CMP VDES ; VERL=LEN(X$)
4240 - BCC SPR2 ; V$ KUERZER:
4250 - INX ; XREG=1
4260 - LDA VDES ; VERL=LEN(V$)
4270 -STA VERL
4300 -; X$ <?> V$ -----
4310 - LDY #0 ; HAUPTSCHLEIFE
4320 -VERGLX1 LDA (XDES+1),Y ; -----
4330 - CMP (VDES+1),Y ; BYTE UM BYTE
4340 - BNE VERGLX2 ; VERGLEICHEN
4350 - INY
4360 - CPY VERL ; VERGL. LAENGE
4370 - BCC VERGLX1 ; PRUEFEN
4380 - CPX #1
4390 -VERGLX2 BCS VERGLY
4400 -; X = X + 1 -----
4410 - CLC
4420 - LDA XADR ; LOW-BYTE
4430 - ADC #3 ; + 3
4440 - STA XADR ; ZEIGT AUF
4450 - BCC VERGLX ; NAECHSTEN
4460 - INC XADR+1 ; DESCRIPTOR
4470 - BCS VERGLX
4480 -; SPRUNG -----
4490 -BRUECKE1 BCC VSTRING
4491 -BRUECKE5 BNE BRUECKE4
4495 -;
5000 -; ----- Y - VERGLEICHSSCHLEIFE
5100 -; Y$ = A$(Y) -----
5110 -VERGLY LDY #0 ; DESCRIPTOR
5120 - LDA (YADR),Y ; A$(Y)
5130 - STA YDES ; IN DER
5140 - INY ; ZERO PAGE
5150 - LDA (YADR),Y ; SPEICHERN
5160 - STA YDES+1
5170 - INY
5180 - LDA (YADR),Y
5190 - STA YDES+2
5200 -; VERGLEICHS-LAENGE -----
5210 - LDX #0 ; V$ KUERZER:
5220 - LDA VDES ; XREG=0
5230 - CMP YDES ; VERL=LEN(V$)
5240 - BCC SPR3 ; Y$ KUERZER:
5250 - INX ; XREG=1
5260 - LDA YDES ; VERL=LEN(Y$)
5270 -STA VERL
5300 -; Y$ <?> V$ -----
5310 - LDY #0 ; HAUPTSCHLEIFE
5320 -VERGLY1 LDA (YDES+1),Y ; -----
5330 - CMP (YDES+1),Y ; BYTE UM BYTE
5340 - BNE VERGLY2 ; VERGLEICHEN
5350 - INY
5360 - CPY VERL ; VERGL. LAENGE
5370 - BCC VERGLY1 ; PRUEFEN
5380 - CPX #1
5390 -VERGLY2 BCS TAUSCH

```

Programmierer gesucht!

Zur Verstärkung unserer Entwicklungsabteilung für Unterhaltungsprogramme suchen wir noch einige freischaffende Mitarbeiter, wobei wir auch gerne mit Schülern und Studenten zusammenarbeiten:

- Wenn **Sie** ein Spiel programmiert haben oder gerade programmieren,
- wenn **Sie** die Idee zu einem neuartigen und interessanten Spiel haben,
- wenn **Sie** an der Umsetzung eines bestehenden Spiels auf einen anderen Computer interessiert sind,
- wenn **Sie** gute Kenntnisse in 6502, Z80 oder 68000 Assembler haben und ein Spiel entwickeln möchten, zu dem Sie oder wir das Konzept liefern,
- wenn **Sie** spezielle Fähigkeiten in den Bereichen Grafik und Sound haben,

dann sollten **Sie** sich schnell mit uns in Verbindung setzen!

Als international tätiges Softwarehaus mit Vertriebswegen in 9 Ländern bieten wir Ihnen eine optimale Vermarktung, an der Sie über garantierte Festzahlungen und Umsatzbeteiligungen optimal beteiligt sind. Durch fertige Standardroutinen und unsere Sound- und Grafikspezialisten können wir Sie bei der Entwicklung jederzeit unterstützen.

Weitere Informationen erhalten Sie schriftlich oder telefonisch über unsere Entwicklungsabteilung:

Rainbow Arts, An der Johanneskirche 27, 4830 Gütersloh
Hotline der Entwicklungsabteilung: 0 52 41-168 88



```

5400 -;Y = Y - 1 -----
5410 - SEC
5420 - LDA YADR ; LOW-BYTE
5430 - SBC #3 ; - 3
5440 - STA YADR ; ZEIGT AUF
5450 - BCS VERGLY ; VORHERIGEN
5460 - DEC YADR+1 ; DESCRIPTOR
5470 - BCC VERGLY
5480 -;SPRUNG -----
5490 -BRUECKE2 BCS VERGLX
5491 -BRUECKE3 BCC BRUECKE1
5492 -BRUECKE6 BNE BRUECKE5
5495 -;
6000 -; ----- STRINGS VERTAUSCHEN
6100 -;IF X>=Y -----
6110 -TAUSCH LDA YADR+1 ; HIGH
6120 - CMP XADR+1 ; BYTES
6130 - BCC ZSTUFEN ; X>Y
6140 - BNE SWAP ; X<Y
6150 - LDA XADR ; LOW
6160 - CMP YADR ; BYTES
6170 - BCS ZSTUFEN ; X>=Y
6200 -;SWAP X$,Y$ -----
6210 -SWAP LDX #2
6220 - LDY #2
6230 -SWAP1 LDA XDES,X ; X-DESCRIPTOR
6240 - STA (YADR),Y ; NACH A$(Y)
6250 - LDA YDES,X ; Y-DESCRIPTOR
6260 - STA (XADR),Y ; NACH A$(X)
6270 - DEX
6280 - DEY
6290 - BPL SWAP1
6300 -;X = X + 1 -----
6310 - CLC
6320 - LDA XADR ; SIEHE
6330 - ADC #3 ; ZEILE
6340 - STA XADR ; 4400
6350 - BCC SPR4
6360 - INC XADR+1
6400 -;Y = Y - 1 -----
6410 -SPR4 SEC
6420 - LDA YADR ; SIEHE
6430 - SBC #3 ; ZEILE
6440 - STA YADR ; 5400
6450 - BCS SPR5
6460 - DEC YADR+1
6500 -;IF X <= Y -----
6510 -SPR5 LDA YADR+1
6520 - CMP XADR+1
6530 - BCC ZSTUFEN ; X>Y
6540 - BNE BRUECKE2 ; X<Y
6550 - LDA YADR
6560 - CMP XADR
6570 - BCS BRUECKE2 ; X<=Y
6580 - BCC ZSTUFEN ; X>Y
6600 -; SPRUNG -----
6610 -BRUECKE7 BNE BRUECKE6
7000 -; ----- STUFEN-VERZWEIGUNG
7100 -;Z=Z+1:RG=Y -----
7110 -ZPLUS LDA RG ; Z = Z + 1
7120 - PHA ; BEDEUTET
7130 - LDA RG+1 ; RG AUF STACK
7140 - PHA ; LEGEN
7150 - LDA YADR
7160 - STA RG ; RG
7170 - LDA YADR+1 ; =
7180 - STA RG+1 ; Y
7185 - CLC ; NEUEN V$
7190 - BCC BRUECKE3 ;BERECHNEN
7200 -;LG = LG + 1 -----
7210 -ZGLEICH CLC
7220 - LDA LG
7230 - ADC #3 ; SIEHE
7240 - STA LG ; ZEILE
7250 - BCC SPR6 ; 4400
7260 - INY
7270 -SPR6 STY LG+1
7300 -;IF LG < RG -----
7310 - CPY RG+1
7320 - BCC BRUECKE3 ; LG<RG
7330 - BNE ZMINUS ; LG>RG
7340 - CMP RG
7350 - BCC BRUECKE3 ; LG<RG
7360 - BCS ZMINUS ; LG>=RG
7400 -;IF LG < RG -----
7410 -ZSTUFEN LDA LG
7420 - LDY LG+1
7430 - CPY YADR+1
7440 - BCC STACK ; LG<RG
7450 - BNE ZGLEICH ; LG>RG
7460 - CMP YADR
7470 - BCC STACK ; LG<RG
7480 - BCS ZGLEICH ; LG>=RG
7500 -;STACK PRUEFEN -----
7510 -STACK TSX
7520 - CPX STMI ; STACKMINIMUM
7530 - BCS ZPLUS ; PRUEFEN
7540 - STX STMI
7550 - CPX STUG ; STACK-UG
7560 - BCS ZPLUS ; PRUEFEN
7570 - LDY STIN ; RUECKSPRUNG
7580 - TXS ; ZUM
7585 - BNE BRUECKE7 ; ANFANG
7600 -;Z = Z - 1 -----
7610 -ZMINUS PLA ; Z = Z - 1
7620 - STA RG+1 ; BEDEUTET
7630 - PLA ; RG VOM STACK
7640 - STA RG ; HOLEN
7650 - LDY RG+1
7660 - CPX #0 ; RG HIGHBYTE
7670 - BNE ZGLEICH ; > 0
7680 -; ; = 0
8000 -; -----BASIC-DATEN SPEICHERN
8100 -;STACK-POINTER -----
8110 -AUS LDX STIN
8120 - TXS
8200 -;ZERO-PAGE -----
8210 - LDX #0
8220 -ZPRUECK PLA
8230 - STA 10,X
8240 - INX
8250 - CPX #20
8260 - BCC ZPRUECK
8270 - RTS

```

READY.

Listing 2. (Schluß)

```

PROGRAM : QUICKSORT.COD C350 C4F6
-----
C350 : BA 86 FC 86 FD 8A 38 E9 46
C358 : 52 B0 01 60 69 02 85 FE 09
C360 : A2 13 B5 0A 48 CA 10 FA 4C
C368 : BA 86 FC A9 00 48 48 18 6D
C370 : A5 2F 69 0A 85 19 A5 30 61
C378 : 69 00 85 1A 18 A0 02 B1 7B
C380 : 2F 65 2F AA C8 B1 2F 65 25
C388 : 30 A8 38 BA E9 03 85 1B 6F
C390 : 48 98 E9 00 85 1C 48 D0 9B
C398 : 02 D0 D0 A2 03 85 19 95 F8
C3A0 : 0A CA 10 F9 18 A5 0A 65 F4
C3AB : 0C AA A5 0B 65 0D 4A 85 C7
C3B0 : 0F 8A 6A 90 06 E9 01 B0 C6
C3B8 : 02 C6 0F B5 0E A0 00 B1 DB
C3C0 : 0E 85 16 C8 B1 0E 85 17 FF
C3CB : C8 B1 0E 85 18 A0 00 B1 87
C3D0 : 0A 85 10 C8 B1 0A 85 11 5E
C3D8 : C8 B1 0A 85 12 A2 00 A5 2E
C3E0 : 10 C5 16 90 03 E8 A5 16 A5
C3E8 : 85 1D A0 00 B1 11 D1 17 3D
C3F0 : D0 07 C8 C4 1D 90 F5 E0 FE
C3F8 : 01 B0 11 18 A5 0A 69 03 EF
C400 : 85 0A 90 C9 E6 0B B0 C5 FD
C408 : 90 91 D0 8D A0 00 B1 0C 2F
C410 : 85 13 C8 B1 0C 85 14 C8 56
C418 : B1 0C 85 15 A2 00 A5 16 C0
C420 : C5 13 90 03 E8 A5 13 85 06
C428 : 1D A0 00 B1 17 D1 14 D0 BE
C430 : 07 C8 C4 1D 90 F5 E0 01 AE
C438 : B0 13 38 A5 0C E9 03 85 5C
C440 : 0C 00 C9 C6 0D 90 C5 B0 BD
C448 : 84 90 BD D0 BD A5 0D C5 66
C450 : 0B 90 69 D0 06 A5 0A C5 59
C458 : 0C B0 61 A2 02 A0 02 B5 02
C460 : 10 91 0C B5 13 91 0A CA 6E
C468 : 88 10 F4 18 A5 0A 69 03 8F
C470 : 85 0A 90 02 E6 0B 38 A5 52
C478 : 0C E9 03 85 0C B0 02 C6 C6
C480 : 0D A5 0D C5 0B 90 35 D0 0B
C488 : BE A5 0C C5 0A B0 B8 90 FF
C490 : 2B D0 B8 A5 1B 48 A5 1C C9
C498 : 48 A5 0C 85 1B A5 0D 85 85
C4A0 : 1C 18 90 A5 1B A5 19 69 87
C4A8 : 03 85 19 90 01 C8 84 1A 63
C4B0 : C4 1 90 95 D0 26 C5 1B E5
C4B8 : 8F B0 20 A5 19 A4 1A 2A
C4C0 : 0D 90 0B D0 DE C5 0C 63
C4C8 : 90 02 B0 D8 BA E4 FD B0 CD
C4D0 : C2 86 FD E4 FE 80 BC A6 A7
C4D8 : FC 9A D0 B5 68 85 1C 68 00
C4E0 : 85 1B A6 1C E0 00 D0 BC EB
C4E8 : A6 FC 9A A2 0D 68 95 0A B5
C4F0 : EB E0 14 90 F8 60 00 00 F2

```

Listing 3. Das MSE-Listing für Quicksort in Maschinensprache