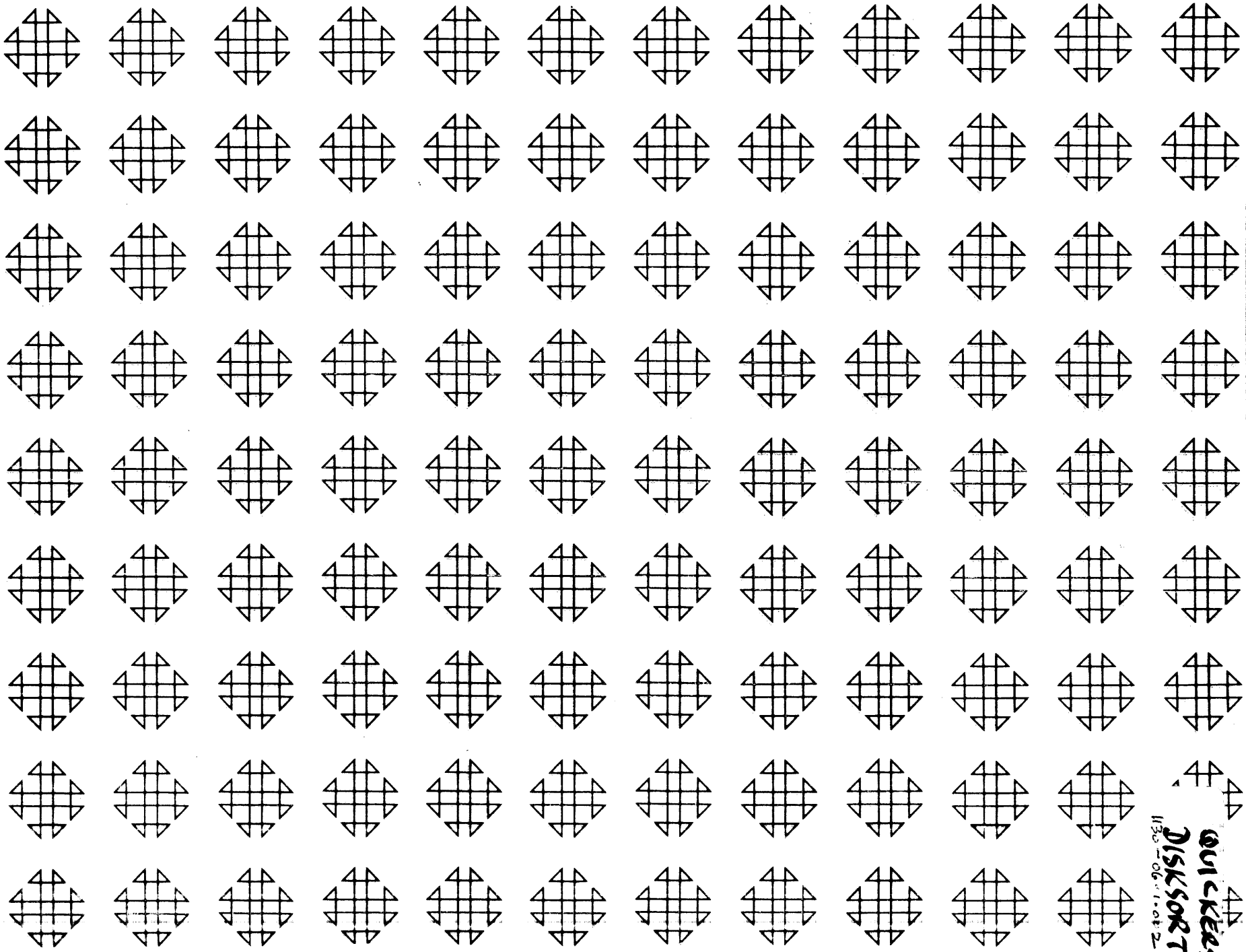


Sort Subprograms for IBM 1130, Quickersort, Disksort 1 and Disksort 2

1130-06.1.002

DISCLAIMER

Although each program has been tested by its contributor, no warranty, express or implied, is made by the contributor or any User's Group, as to the accuracy and functioning of the program and related program material, nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the contributor or any User's Group, in connection therewith.

Sort Subprograms for the IBM 1130; Quickersort, Disksort 1 and Disksort 2

Robert N. Kubik
The Babcock-Wilcox Company
1201 Kemper Street
Lynchburg, Virginia

March 21, 1967

Modifications or revisions to this program, as they occur, will be announced in the appropriate Catalog of Programs for IBM Data Processing Systems. When such an announcement occurs, users should order a complete new program from the Program Information Department.

PROGRAM ABSTRACT

1. TITLE (If subroutine, state in Title): QSORT, DSRT 1, DSRT 2 (3 subroutines)
Subject Classification: 06.1

2. Author; Organization: Mr. Robert N. Kubik, The Babcock-Wilcox Company

Date: 3/26/67 Users Group Membership Code:

3. Direct Inquiries to Name: Mr. Robert N. Kubik, The Babcock-Wilcox Company
1201 Kemper Street, Lynchburg, Virginia Phone: 703 846-7371

4. Description/Purpose: (5. Method: 6. Restriction/Range; When Applicable):
Three sort subprograms written in FORTRAN for the IBM 1130 are included. First, sort based on Algorithm 271 as published in the ACM Communications requires only a minimum 1130. Second, a disk sort which requires 8K memory and disk. Third, a disk sort which requires 4K memory and disk but is less effecient. These would not be practical for sorting large files. These subprograms are available at this time for the 1130.

7. Specifications (Check or fill in appropriate spaces):

a. Storage used by Program: 4K, 8K, 4K, respectively

b. Equipment required by program: Card____; Magnetic Tape____; Number of Drives____; Paper Tape____; Disk File 2&3; Number of Drives____; TNS, TNF, MF,____; Auto divide____; Indirect addressing____; Floating Point Hardware ____; 1620 Model I ____; Model II____; 1443 Printer____; Index Registers___; Binary Capabilities____; other (specify)

Can program be used on lesser machine? NO . Specify which requirements can be easily removed

c. Programmed in: Fortran without Format____; Fortran with Format____; Fortran II____; Other Fortran (specify)_____; SPS (specify assembler used) _____; Other (specify) 1130 Fortran

d. Type of Program: Mainline, Complete____; Subroutine X ; If subroutine, for use with SPS (specify type of SPS)_____; Fortran (specify type of Fortran _____; Other (specify)

8. Additional Remarks:

Table of Contents

DECK KEY

Deck #				
1	FORTRAN	deck	QSORT	cards 1 - 110
2	FORTRAN	deck	DSRT1	cards 1 - 168
3	FORTRAN	deck	MAINL	cards 1 - 30
4	Sample problem input			cards 1 - 7
5	FORTRAN	deck	DSRT2	cards 1 - 94

card sequence numbers are in columns 76 - 80

Program Abstract

Three sort subprograms are included in this report. A FORTRAN version of Algorithm 271 Quickersort, published in CACM, vol. 8, No. 11, Oct., 1965, by R. S. Scowen; which is a very fast internal sort for randomly distributed keys. It is herein called QSORT. It requires minimal storage, i.e. storage for the records to be sorted plus one record. Run time in favorable cases can be approximated by:

$$\left(\frac{\text{number of records}}{80}\right)^{1.3} = \text{run time in seconds,}$$

for instance 1200 records of two words each were sorted in 30.4 seconds. However, unfavorable key distribution can result in substantially longer run time. Cases have been observed where the exponent in the above formula approaches 1.9.

DSRTI is a disk sort for files exceeding core storage available. It uses QSORT to create ordered strings of data. Then a series of passes are made which can best be described as combining the logic of pair exchange and merging. DSRTI requires 2560 plus the number of words in a record, words of working storage. Run time is approximately

$$\frac{(\text{number of records})^2 \times \text{number of words/record}}{10{,}000} = \text{run time in seconds.}$$

DSRT2 is a disk sort for files exceeding fast storage size available. It is a direct application of the same algorithm as QSORT. It requires 1632 words working storage, thereby minimizing storage requirement. In all cases tested to date it has proven substantially slower than DSRT1. It is possible, however, it would prove to be faster than DSRT1 on relatively large files with large records, (80 words per record or more).

These sorts are intended for sorting keys and tags (e.g. record numbers) so that one can find records in a larger file in proper sequence for updating or report preparation. They can, however, be used as any sort routines would be - provided the records have less than 321 words/record; the keys are integers and are grouped in major to minor sequence as the first items of each record. Suggestions are included for circumventing this restriction on the keys and for speeding up these routines. These subroutines should be suitable for the IBM 1800 although they have not been tested on that machine.

March 21, 1967

This program and its documentation were written by Mr. Robert N. Kubik, (the Babcock-Wilcox Company, 1201 Kemper Street, Lynchburg, Virginia, 703 846-7371). They have been submitted to the Program Information Department for general distribution in the expectation that they may prove useful to other members of the data processing community. The program and its documentation are, essentially, in the author's original form and have not been subject to any formal testing. IBM only serves as the distribution agency in supplying this program. It is the user's responsibility to determine the usefulness and technical accuracy of the program in his own environment. This program is not part of the Programming Systesm(Type I) and Application Programs (Type II). Questions concerning the use of the program should be directed to the author. Any changes to the program will be reflected in the appropriate catalog of programs; however, the changes will not be distributed automatically to users.

USER INFORMATION

Purpose: QSORT is an internal sort for files which can be held in core storage. DSRT1 and DSRT2 are for files exceeding the size of available core storage.

Advantages: QSORT minimizes working storage and is very fast for randomly distributed keys. DSRT1 is relatively efficient. DSRT2 minimizes working storage requirement. Sorting large files is awkward at best on a one disk system. Usually the most efficient solution is to sort only keys and tags (e.g. record numbers) and then access the file in its existing order by means of the table so generated. These subroutines appear to be adequate for that purpose.

Method: QSORT and DSRT2 continually split the file into parts such that the keys of all records in one part are less than the keys of all records in the other part, with a third part in the middle consisting of a single record. A record with key t is chosen arbitrarily, i and j are the lower and upper records of the segment being split. After the split has taken place the qth record will have been found such that the key of the qth record is t and the key of the Ith record $\leq t \leq$ the key of the Jth record, for all i and j such that $i \leq I < q < J \leq j$. The program then performs operations on the two segments; the ith thru (q-1)th records and the (q+1)th thru jth records as follows. The smaller segment is split and the position of the larger segment is stored in the lt (lower temporary) and ut (upper temporary) arrays. If the segment to be split has two or fewer elements it is sorted and another segment obtained from the lt and ut arrays. When no more segments remain, the array is completely sorted.

DSRT1 first sorts groups of records of approximate cylinder size by use of QSORT. It then handles the file in groups of records of approximately one half cylinder size. It makes successive merge passes merging the second and third group, the fourth and fifth group and so on; and then merges the first and second group, the third and fourth and so on. These merge passes are alternated until the file is in order.

Restrictions and range:

-number of records $\leq$ 32,767
-number of words/record $\leq$ 320
-keys numeric, the first words in the record and in major to minor sequence
-sorts file 1 as defined in FORTRAN DEFINE FILE statement
-integers are considered to be one word each

-in QSORT and DSRT2 p is the position of an arbitrary record in the file. The best value of p is one which splits the segment into two halves of equal size, thus if the file (segment) is roughly sorted, the middle element is an excellent choice. If the file (segment) is completely random the middle element is as good as any other. If, however, the file (segment) is such that the parts i....p and p+1..j are both sorted the middle element could be very bad. Accordingly in some circumstances p = (i + j)/2 should be replaced by p = (i + 3 * j)/4 or by a p selected at random. The method of selecting p used in these routines will prove to be best in the majority of cases, however.

Program Requirements: These subroutines as distributed are designed for use within FORTRAN programs operated under the 1130 Disk Monitor System. QSORT requires a 4K machine only, although the more space that is available the larger the file that can be sorted. DSRT1 requires 8K storage and a disk. DSRT2 requires 4K storage and a disk.

Timing: In QSORT the number of words in the key influence run time very little; the number of words in a record somewhat more, but not substantially. The largest effects on run time are the number of records and the distribution of keys. A number of cases run with random key distribution have exhibited a run time characterized by the expression:

$$\left(\frac{\text{number of records}}{80}\right)^{1.3} = \text{run time in seconds,}$$

typical of these cases are the following:

number of records	words/record	words/key	time
300	2	2	5.4 sec.
600	2	2	12.3 sec.
1200	2	2	30.4 sec.

Under certain circumstances run time can be substantially longer (see Restrictions and Range) in fact series of cases have been observed which fit the expression:

$$\left(\frac{\text{number of records}}{80}\right)^{1.9} = \text{run time in seconds.}$$

Run time for larger records (say eight words or more) will be underestimated by this formula.

DSRT1 timings can be approximated by:

$$\frac{(\text{number of records})^2 \cdot \text{number of words per record}}{10{,}000} = \text{run time in seconds.}$$

Run time for large records (say eighty words or more) will be underestimated by this formula.

DSRT2 took longer on all cases run and no formula for predicting time has been derived. It is conjectured however, that on large files with 80 or more words per record it may prove more efficient than DSRT1. Some cases which have been timed follow.

records	words/record	DSRT1	DSRT2
128	80	202 sec.	230 sec.
250	20	120 sec.	315 sec.
1000	7	730 sec.	

Program Modification Aids

All coding which compares keys has been surrounded by comment cards with ***** on them and text explaining "insert your own code for non numeric keys or keys which are not the initial words of the record - from here ....- to here."

An assembly language version of QSORT would be worthwhile and should improve the speed of that routine by a factor of two. The algorithm should be clear from the FORTRAN code listing and flow chart.

DSRT1 could be speeded up for sorting small records by replacing the FORTRAN read/write statements by a routine which reads and writes blocks of records and delivers these to the DSRT1 subprogram on call. This would, however, be no advantage for large records and would impose additional restrictions on the program which creates the file to be sorted.

DSRT1 and DSRT2 sort FILE1. This could be changed by changing the file number in the disk read/write statement.

Input-Output Description

QSORT is called by -
CALL QSORT (NREC, NWREC, NWKEY, A, T)
where NREC is the number of records
NWREC is the number of words/record
NWKEY is the number of words/key
A is a one dimensional array containing the file to be sorted
T is a working storage array of NWREC words.
It leaves the first NREC records in A sorted in ascending sequence.

DSRT1 is called by -
CALL DSRT1 (NRFIL, NWREC, NWKEY, ARRAY, TEMP)
where NRFIL is the number of records in the file
NWREC is the number of words/record
NWKEY is the number of words/key
ARRAY is a 2560 word array for working storage
TEMP is a working storage array of NWREC words
it leaves the first NRFIL records in FILE 1 sorted in ascending sequence.

DSRT2 is called by -
CALL DSRT2 (NREC, NWREC, NWKEY)
where NREC is the number of records in the file
NWREC is the number of words/record
NWKEY is the number of words/key
it leaves the first NREC records in FILE 1 sorted in ascending sequence.

Sample Problem

A driver program called MAINL is included with the deck distributed. It reads test data, creates a file and prints it, calls DSRT1 (which calls QSORT) and prints the sorted file. This same driver may be used for DSRT2 by replacing the integer statement with INTEGER ARRAY (2560) and the call statement with CALL DSRT2 (NRFIL, NWREC, NWKEY)

Operating Instructions

Illogical branches in DSRT1 result in a stop at PAUSE 1. If this occurs something is wrong with the machine or the program.

References

C20-1639-0 Sorting Techniques, IBM Data Processing Techniques.

Hoare, C. A. R. Algorithms 63 and 64. Comm. ACM 4 (July 1961), 321.

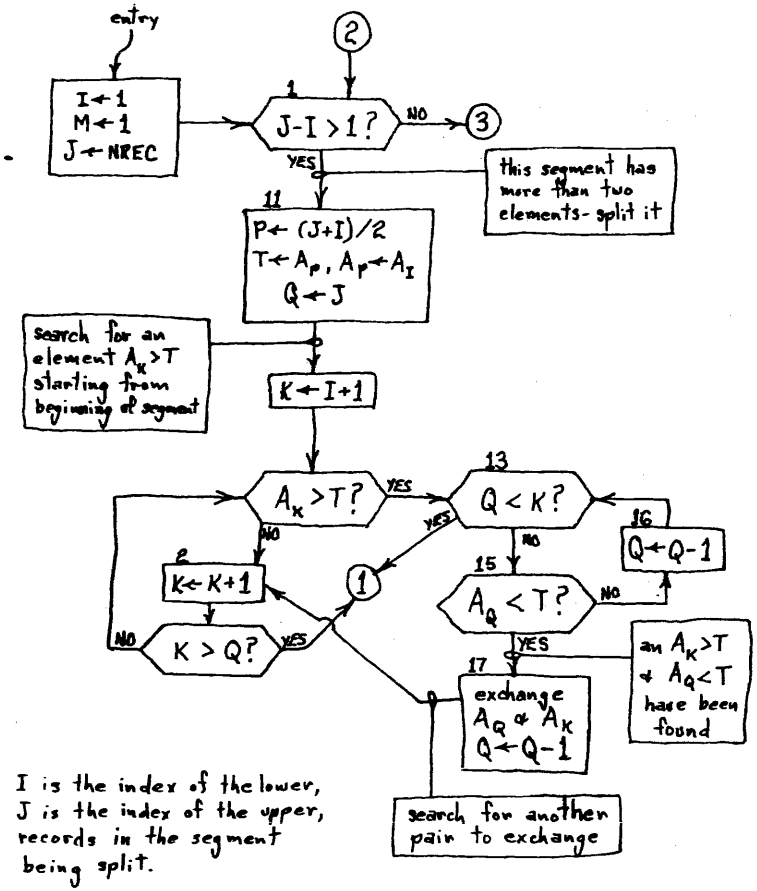
Hibbard, Thomas N. Some combinatorial properties of certain trees with applications to searching and sorting. J. ACM 9 (Jan. 1962), 13.

Hibbard, Thomas N. An empirical study of minimal storage sorting. Comm. ACM 6 (May 1963), 206-213.

Scowen, R. S. Algorithm 271 (QUICKERSORT) Comm. ACM 11 (Nov. 1965), 669-670.

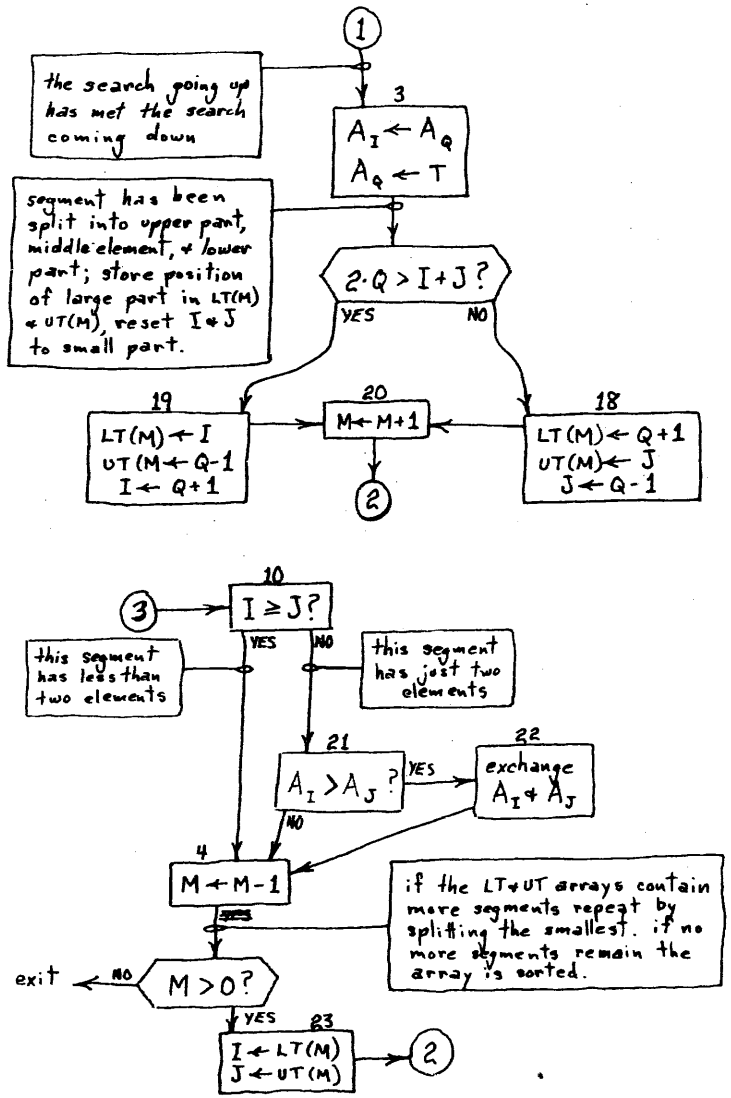
Systems Material

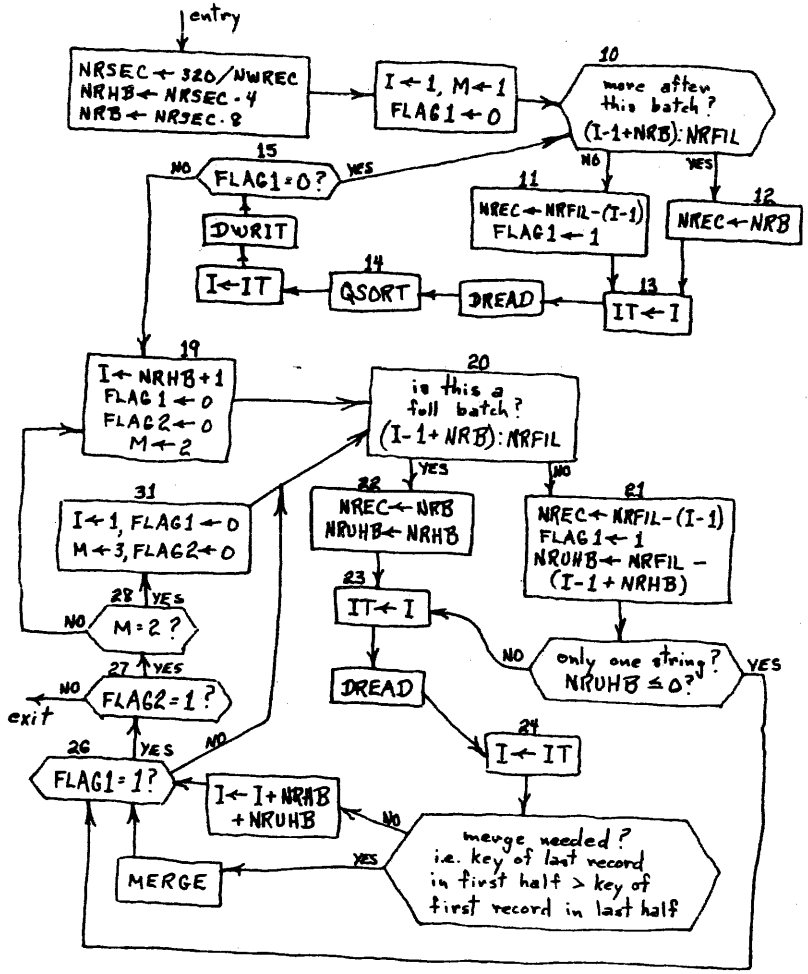
Statement numbers from the FORTRAN code appear above the appropriate box in these flowcharts as a cross index with the code. QSORT and DSRT2 have the same logic and the interpretation of that flowchart for either should pose no difficulty.



I is the index of the lower, J is the index of the upper, records in the segment being split.

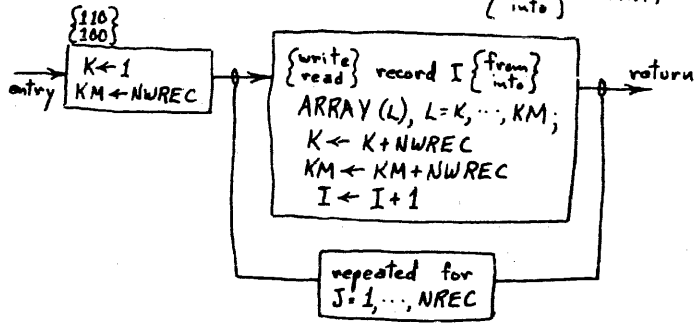
QSORT and DSRT2 Flowchart



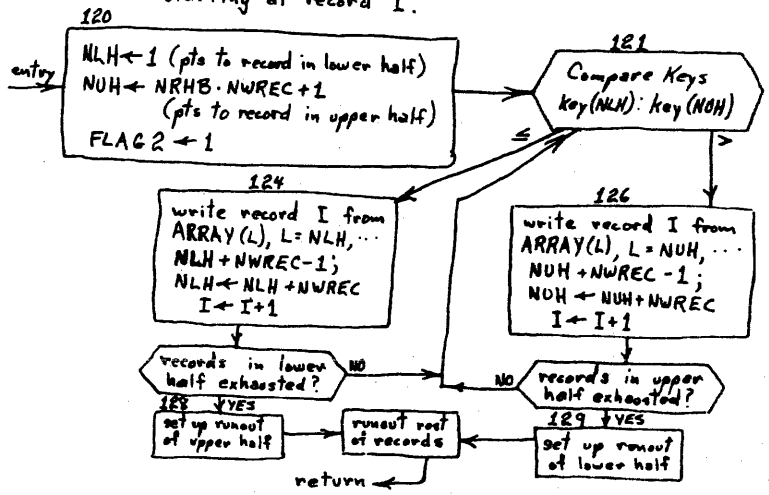


DSRT1 Flowchart

{DWRIT writes on / DREAD reads from} disk starting at record I
NREC words of NWREC words each {from / into} ARRAY



MERGE merges NRHB records from lower half of ARRAY with NRUHB records from upper half of ARRAY and writes them on disk starting at record I.



GSORT, DSRT1, MAINL, completeability run

```
// JOB T
// FOR
```

PAGE 01

```
*      NAME QSORT
*      ONE WORD INTEGERS
*      LIST ALL
```

PAGE 02

```
      SUBROUTINE QSORT (NREC,NWREC,NWKEY,A,T)
C     QUICKSORT ALGORITH 271 CACM VOL 8 NO 11
C      NREC IS THE NUMBER OF RECORDS, NWREC IS THE NUMBER
C      OF WORDS PER RECORD, NWKEY IS THE NUMBER OF WORDS
C      PER KEY, A IS THE ARRAY OF RECORDS WHICH TAKES
C      NREC*NWREC WORDS, T IS WORKING STSRAGE OF NWREC WORDS
C      NREC*NWREC WORDS, T IS WORKING STORAGE OF NWREC WORDS
      INTEGER A(32767), UT(15),LT(15),T(16383),Q,P,X
      M=1
      I=1
      J=NREC
C     N
    1 IF(J-I-1)10,10,11
   11 P=(J+ I)/2
      NP=(P-1)*NWREC+1
      NI=(I-1) *NWREC+1
      DO 110 N=1,NWREC
      T(N)=A(NP)
      A(NP)=A(NI)
      NP=NP+1
  110 NI=NI+1
      Q=J
      I1=I+ 1
      DO 2 K=I1,Q
C*****INSERT YOUR OWN CODE FOR NON NUMERIC KEYS
C*****OR KEYS WHICH ARE NOT THE INITIAL WORDS
C*****OF THE RECORD -FROM HERE
      NK=(K-1)*NWREC+1
      DO 130 N=1,NWREC
      IF (A(NK)-T(N))2,130,13
  130 NK=NK+1
      GO TO 2
C*****                 -TO HERE
   13 IF (Q-K)14,15,15
C*****INSERT YOUR OWN CODE FOR NON NUMERIC KEYS
C*****OR KEYS WHICH ARE NOT THE INITIAL WORDS
C*****OF THE RECORD -FROM HERE
   15 NQ=(Q-1)*NWREC+1
      DO 150 N=1,NWKEY
      IF(A(NQ)-T(N)) 17,150,16
  150 NQ=NQ+1
      GO TO 16
C*****                       -TO HERE
   17 NK=(K-1)*NWREC+1
      NQ=(Q-1)*NWREC+1
      DO 170 N=1,NWREC
      X    =A(NK)
      A(NK)=A(NQ)
      A(NQ)=X
      NK=NK+1
  170 NQ=NQ+1
      Q=Q-1
      GO TO 2
```

```
   16 Q=Q-1
      GO TO 13
   14 GO TO 3
C     L
    2 CONTINUE
C     M
    3 NI=(I-1) *NWREC+1
      NQ=(Q-1)*NWREC+1
      DO 300 N=1,NWREC
      A(NI)=A(NQ)
      A(NQ)=T(N)
      NI=NI+1
  300 NQ=NQ+1
      IF((2*Q)-(I+ J))18,18,19
   19 LT(M)=I
      UT(M)=Q-1
      I=Q+1
      GO TO 20
   18 LT(M)=Q+1.
      UT(M)=J
      J=Q-1
   20 M=M+1
      GO TO 1
   10 IF(I-J)21,4,4
C*****INSERT YOUR OWN CODE FOR NON NUMERIC KEYS
C*****OR KEYS WHICH ARE NOT THE INITIAL WORDS
C*****OF THE RECORD    -FROM HERE
   21 NI=(I-1) *NWREC+1
      NJ =(J-1)*NWREC+1
      DO 211 N=1,NWREC
      IF (A(NI)-A(NJ))4,210,22
  210 NI=NI+1
  211 NJ=NJ+1
      GO TO 4
C*****                -TO HERE
   22 NI=(I-1) *NWREC+1
      NJ =(J-1)*NWREC+1
      DO 220 N=1,NWREC
      X    =A(NI)
      A(NI)=A(NJ)
      A(NJ)=X
      NI=NI+1
  220 NJ=NJ+1
C     P
    4 M=M-1
      IF(M)24,24,23
   23 I=LT(M)
      J=UT(M)
      GO TO 1
   24 RETURN
      END
```



```
VARIABLE ALLOCATIONS
 UT    =0010  LT    =001F  Q     =0020  P     =0021  X     =0022  N     =0023  I     =0024  J     =0025  NP    =0026  NI    =0027
 NQ    =0028  I1    =0029  K     =002A  NK    =002B  NJ    =002C  NJ    =002D

STATEMENT ALLOCATIONS
 1     =0172  11    =037A  110   =00BE  130   =00FF  13    =010F  15    =0115  150   =013A  17    =014A  170   =0164  16    =01AA
 14    =01A2  2     =01A4  3     =01AC  300   =01E7  19    =0206  18    =021F  20    =0236  10    =023E  21    =0244  210   =0274
 211   =027A  22    =029A  220   =02C4  4     =02D2  23    =02DC  24    =02ED

FEATURES SUPPORTED
 ONE WORD INTEGERS

CALLED SUBPROGRAMS
 SUMSC    SUBIN

INTEGER CONSTANTS
      1=0032       2=0033

CORE REQUIREMENTS FOR QSORT
 COMMON       0  VARIABLES     50  PROGRAM     702

END OF COMPILATION
```

```
// DUP                                                          0109
*STORE        WS  UA  DSORT                                     0110
1492 002A
```

```
// FOR                                                          0001
                                                                PAGE 01
*       NAME DSRT1                                              0002
*       ONE WORD INTEGERS                                       0003
*       LIST ALL                                                0003A
```

```
      SUBROUTINE QSRT1(NRFIL,NWREC,NWKEY,ARRAY,TEMP)
C     A SUBROUTINE WHICH SORTS (IN ASCENDING
C     SEQUENCE) THE FIRST NRFIL RECORDS
C     OF NWREC WORDS EACH IN FILE 1.
C      ARRAY IS A WORKING STORAGE VECTOR OF 2560 WORDS
C      TEMP IS A WORKING STORAGE VECTOR OF NWREC WORDS
C      THERE ARE NWKEY WORDS IN THE KEY WHICH
C      ARE INTEGERS, THE FIRST IN THE RECORD
C      AND IN MAJOR TO MINOR SEQUENCE
      INTEGER FLAG1,FLAG2,ARRAY(2560) ,TEMP(320)
      NRSEC = 320 / NWREC
      NRHB = NRSEC * 4
      NRB = NRSEC * 8
C     I COUNTS RECORDS IN THE FILE
C     J COUNTS RECORDS IN A BATCH
C     K COUNTS WORDS IN A BATCH
C     L COUNTS WORDS IN A RECORD
C     M IS A CONTROL INDEX
C     NLH POINTS TO RECORD IN LOWER HALF
C     NUH POINTS TO RECORD IN UPPER HALF
C     KEYL POINTS TO A KEY WORD IN NLH'TH RECORD
C     KEYU POINTS TO A KEY WORD IN NUH'TH RECORD
C     N COUNTS WORDS IN KEY
C     PHASE 1 TRANSFORM FILE INTO ORDERED
C     STRINGS OF NRB RECORDS
      I = 1
      M = 1
      FLAG1 = 0
C     MORE AFTER THIS BATCH+
   10 IF ((I-1+NRB)-NRFIL) 12,11,11
   11 NREC = NRFIL - (I-1)
      FLAG 1 = 1
      GO TO 13
   12 NREC = NRB
   13 IT = I
C     DREAD
      GO TO 100
   14 CALL QSORT (NREC,NWREC,NWKEY,ARRAY,TEMP)
      I = IT
C     DWRIT
      GO TO 110
   15 CONTINUE
      IF (FLAG1) 99,10,19
C     PHASE 1 COMPLETE
C     PHASE 2 SUCCESIVE MERGE PASSES
   19 I=NRHB+1
      FLAG1=0
      FLAG2=0
      M=2
C     IS THIS A FULL BATCH +
   20 IF((I-1+NRB)-NRFIL) 22,21,21
   21 NREC=NRFIL-(I-1)
      FLAG1=1
      NRUHB=NRFIL-(I-1+NRHB)
C     ONLY ONE STRING +
      IF (NRUHB) 26,26,23
   22 NREC = NRB
      NRUHB=NRHB
   23 IT=I
C     DREAD
      GO TO 100
   24 I = IT
C     PAGE REFORD
C*****INSERT YOUR OWN CODE FOR NON NUMERIC KEYS
C*****OR KEYS WHICH ARE NOT THE INITIAL WORDS
C*****OF THE RECORD   -FROM HERE
      KEYL= (NRHB-1)*NWREC+1
      KEYU=NRHB*NWREC+1
      DO 25 K=1,NWKEY
      IF (ARRAY(KEYL)-ARRAY(KEYU)) 30,29,120
   29 KEYL=KEYL+1
   25 KEYU=KEYU+1
C*****                -TO HERE
   30 I=I+NRHB+NRUHB
   26 IF (FLAG1)99,20,27
   27 IF (FLAG2)99,60,28
   28 GO TO (99,31,19 ),M
   31 I=1
      FLAG1=0
      FLAG2=0
      M=3
      GO TO 20
   60 RETURN
   99 PAUSE 1
      CALL EXIT
C     DREAD
C     ROUTINE TO READ FROM DISK STARTING AT
C     RECORD I NREC RECORDS OF NWREC
C     WORDS EACH INTO ARRAY, RETURN
C     INDEX IS M
  100 K=1
      KM=K+NWREC
      DO 101 J=1,NREC
      READ (1'I) (ARRAY(L),L=K,KM)
      I=I+1
      K=K+NWREC
  101 KM=KM+NWREC
      GO TO (14,24,241),M
C     DWRIT
C     ROUTINE TO WRITE ON DISK STARTING
C     AT RECORD I NREC RECORDS OF NWREC
C     WORDS EACH INTO ARRAY, RETURN
C     INDEX IS M
  110 K=1
      KM=NWREC
      DO 111 J=1,NREC
      WRITE (1'I) (ARRAY(L),L=K,KM)
      I=I+1
      K=K+NWREC
  111 KM=KM+NWREC
      GO TO 15
C     MERGE
C     ROUTINE TO MERGE NRHB RECORDS
C     FROM THE LOWER HALF OF ARRAY
C     WITH NRUHB RECORDS FROM THE
C     UPPER HALF OF ARRAY AND WRITE
C     THESE ON DISK STARTING AT
C     RECORD I, NWREC WORDS PER RECORD,
C     NWKEY WORDS IN KEY, M IS RETURN INDEX
  120 NLH=1
```

```
      NUH=NRHB*NWREC+1                                                  0124
      FLAG2=1                                                           0125
C     COMPARE KEYS IN TWO RECORDS                                       0126
C*****INSERT YOUR OWN CODE FOR NON NUMERIC KEYS                         0127
C*****OR KEYS WHICH ARE NOT THE INITIAL WORDS                           0128
C*****OF THE RECORD   =FROM HERE                                        0129
  121 KEYL=NLH                                                          0130
      KEYU=NUH                                                          0131
      DO 123 N=1,NWKEY                                                  0132
      IF(ARRAY(KEYL)-ARRAY(KEYU))124,122,126                            0133
  122 KEYL = KEYL+1                                                     0134
  123 KEYU = KEYU+1                                                     0135
C*****                        =TO HERE                                0136
C     RECORD FROM LOWER HALF IS WRITTEN                                 0137
  124 NLHM=NLH+NWREC-1                                                  0138
      WRITE (1'I) (ARRAY(L),L=NLH,NLHM)                                 0139
      I=I+1                                                             0140
      NLH=NLH+NWREC                                                     0141
C     ARE RECORDS IN LOWER HALF EXHAUSTED                               0142
      IF((NLHM/NWREC)-NRHB)121,128,99                                   0143
C     RECORD FROM UPPER HALF IS WRITTEN                                 0144
  126 NUHM = NUH+NWREC-1                                                0145
      WRITE (1'I) (ARRAY(L),L=NUH,NUHM)                                 0146
      I=I+1                                                             0147
      NUH=NUH+NWREC                                                     0148
C     ARE RECORDS IN UPPER HALF EXHAUSTED                               0149
      IF((NUHM/NWREC)-(NRUHB+NRHB)    )                                 0150
     1121,129,99                                                        0151
C     RUN OUT RECORDS IN UPPER HALF                                     0152
  128 NLIM = (NRUHB+NRHB )*NWREC                                        0153
      NRO=NUH                                                           0154
      GO TO 130                                                         0155
C     RUN OUT RECORDS IN LOWER HALF                                     0156
  129 NLIM = NRHB*NWREC                                                 0157
      NRO=NLH                                                           0158
C     RUN OUT                                                           0159
  130 NROM = NRO+NWREC-1                                                0160
      WRITE (1'I) (ARRAY(L),L=NRO,NROM)                                 0161
      I=I+1                                                             0162
      NRO=NRO+NWREC                                                     0163
      IF(NROM-NLIM) 130,26,99                                           0164
C     MERGE COMPLETE                                                    0165
      END                                                               0166
```



```
VARIABLE ALLOCATIONS
 FLAG1=0002  FLAG2=0003  NRSEC=0004  NRHB =0005  NKB  =0006  I    =0007  M    =0008  NREC =0009  IT   =000A  NRUHB=000B
 KEYL =000C  KEYU =000D  N    =000E  K    =000F  KM   =0010  J    =0011  L    =0012  NLH  =0013  NUH  =0014  NLHM =0015
 NUHM =0016  NLIM =0017  NRO  =0018  NROM =0019

STATEMENT ALLOCATIONS
  10   =0074  11   =007E  12   =008E  13   =0092  14   =0098  15   =00A5  19   =00AB  20   =00BD  21   =00C7  22   =00E7
  23   =00EF  24   =00F5  29   =0127  25   =012D  30   =013B  26   =0143  27   =0149  28   =014F  31   =0156  60   =0168
  90   =016A  100  =016D  101  =019B  110  =01B0  111  =01DF  120  =01EF  121  =0200  122  =0222  123  =0226  124  =0236
 126   =026E  128  =02AE  129  =02BD  130  =02C8

FEATURES SUPPORTED
 ONE WORD INTEGERS

CALLED SUBPROGRAMS
 USORT   SUBSC   PAUSE   SUBIN   SDRED   SDWRT   SDCOM   SDIX

INTEGER CONSTANTS
   320=001C       4=001D       8=001E       1=001F       0=0020       2=0021       3=0022

CORE REQUIREMENTS FOR DSRTB
 COMMON       0  VARIABLES      28  PROGRAM      738

END OF COMPILATION
```

```
// DUP                                                          0167
*STORE      WS  UA  DSRT1                                       0168
1ABC 002C
```

```
// FOR                                                          0001
                                                                PAGE 01
*     NAME MAINL                                                0002
*     ONE WORD INTEGERS                                         0003
*     IOCS (CARD,1132PRINTER,DISK)                              0004
*     LIST ALL                                                  0005
```

```
      INTEGER ARRAY (2560),TEMP(39)                                      0006
      DEFINE FILE 1 (500,39,U,IPT)                                       0007
C     FIRST CARD TELLS HOW MANY CARDS FOLLOW                             0008
      READ (2,90) NRFIL                                                  0009
      WRITE (3,92)                                                       0010
C     LOOP READS CARD,WRITES RECORD ON DISK AND PRINTS RECORD            0011
      DO 1 I=1,NRFIL                                                     0012
      READ (2,90) N1,N2,(TEMP(J),J=1,37)                                 0013
      WRITE (1'I) N1,N2,(TEMP(J),J=1,37)                                 0014
    1 WRITE (3,91)N1,N2,(TEMP(J),J=1,37)                                 0015
C     SORT
      NWKEY=2                                                            0026
      NWREC=39                                                           0018
      CALL DSRT1 (NRFIL,NAREC,NWKEY,ARRAY,TEMP)                          0019
C     PRINT SORTED FILE                                                  0020
      WRITE (3,93)                                                       0021
      DO 2 I=1,NRFIL                                                     0022
      READ (1'I) N1,N2,(TEMP(J),J=1,37)                                  0023
    2 WRITE (3,91) N1,N2,(TEMP(J),J=1,37)                                0024
      CALL EXIT                                                          0033
   90 FORMAT (2I3,37A2)                                                  0034
   91 FORMAT (1H ,I3,1X,I3,1X,37A2)                                      0035
   92 FORMAT (48H1RECORDS PRINTED AS READ                          )     0036
   93 FORMAT (48H0RECORDS PRINTED AS SORTED                        )     0037
      END                                                                0039
```



```
VARIABLE ALLOCATIONS
 ARRAY=0A37  TEMP =0A2E  IPT  =0A2F  NRFIL=0A30  I    =0A31  N1   =0A32  N2   =0A33  J    =0A34  NWKEY=0A35  NWREC=0A36

STATEMENT ALLOCATIONS
 90   =0A3D  91   =0A42  92   =0A4B  93   =0A65  1    =0AD6  2    =0B2A

FEATURES SUPPORTED
 ONE WORD INTEGERS
 IOCS

CALLED SUBPROGRAMS
 DSRT1   FLD     FSTO    SRED    SWRT    SCOMP   SFIO    SIOIX   SIOI    SUBSC   CARDZ   PRNTZ   SUFIO   SUWEZ   SAT
 SDCOM   SDIX    SDI

INTEGER CONSTANTS
     2=0A38      3=0A39      1=0A3A     37=0A3B     39=0A3C

CORE REQUIREMENTS FOR MAINL
 COMMON      0  VARIABLES   2616  PROGRAM    278

END OF COMPILATION
```

```
// XEQ

                                                                        0040

RECORDS PRINTED AS HEAD
  2   2 SORT ROUTINES WOULD BE IF THE RESTRICTIONS ARE OBSERVED.
  2   1 FOR UPDATING OR REPORT PREPARATION. THEY CAN HOWEVE BE USED AS ANY   0042
  1   2 SO THAT ONE CAN FIND RECORDS IN A LARGER FILE IN PROPER SEQUENCE     0043
  1   1 THESE SORTS ARE INTENDED FOR SORTING KEYS AND TAGS  RECORD NUMBERS   0044
                                                                            0045
RECORDS PRINTED AS SORTED
  1   1 THESE SORTS ARE INTENDED FOR SORTING KEYS AND TAGS  RECORD NUMBERS   0045
  1   2 SO THAT ONE CAN FIND RECORDS IN A LARGER FILE IN PROPER SEQUENCE     0044
  2   1 FOR UPDATING OR REPORT PREPARATION. THEY CAN HOWEVE BE USED AS ANY   0043
  2   2 SORT ROUTINES WOULD BE IF THE RESTRICTIONS ARE OBSERVED.             0042
```

DSRT2, MAIN, compile, load & run

```
// JOB T
// FOR                                                                  0001
*
*       NAME DSRT2                                                      PAGE 01
*       ONE WORD INTEGERS                                               0002
*       LIST ALL                                                        0003
                                                                        0003A
```

```
      SUBROUTINE DSRT2 (NREC,NWREC,NWKEY)                                    0004
C      QUICKSORT ALGORITHM 271 CACM VOLN NO11                                0005
C      NREC IS THE NUMBER OF RECORDS, NWREC IS THE NUMBER                    0006
C      OF WORDS PER RECORD, NWKEY IS THE NUMBER OF WORDS                     0007
C      PER KEY, A IS THE ARRAY OF RECORDS                                    0008
      INTEGER Q,P,UT(16),LT(16),AI(320),AJ(320),          AQ(320),AK(320),  0009
     1T(320)                                                                 0010
      M=1                                                                    0011
      I=1                                                                    0012
      J=NREC                                                                 0013
      READ (1'I) (AI(L),L=1,NWREC)                                           0014
C     N                                                                      0015
    1 IF(J-I-1)10,10,11                                                      0016
   11 P=(J+ I)/2                                                             0017
      READ (1'P) (T (L),L=1,NWREC)                                           0018
      WRITE (1'P) (AI(L),L=1,NWREC  )                                        0019
      Q=J                                                                    0020
      READ (1'Q) (AQ(L),L=1,NWREC)                                           0021
      I1=I+ 1                                                                0022
      DO 2 K=I1,Q                                                            0023
      READ (1'K) (AK(L),L=1,NWREC)                                           0024
C*****INSERT YOUR OWN CODE FOR NON NUMERIC KEYS                              0025
C*****OR KEYS WHICH ARE NOT THE INITIAL WORDS                                0026
C*****OF THE RECORD - FROM HERE                                              0027
      DO 130 N=1,NWKEY                                                       0028
      IF(AK(N)-T(N))2,130,13                                                 0029
  130 CONTINUE                                                               0030
      GO TO 2                                                                0031
C*****                  - TO HERE                                            0032
   13 IF (Q-K)14,15,15                                                       0033
C*****INSERT YOUR OWN CODE FOR NON NUMERIC KEYS                              0034
C*****OR KEYS WHICH ARE NOT THE INITIAL WORDS                                0035
C*****OF THE RECORD - FROM HERE                                              0036
   15 DO 150 N=1,NWKEY                                                       0037
      IF(AQ(N)-T(N))17,150,16                                                0038
  150 CONTINUE                                                               0039
      GO TO 16                                                               0040
C*****                  - TO HERE                                            0041
   17 WRITE (1'K) (AQ(L),L=1,NWREC)                                          0042
      WRITE (1'Q) (AK(L),L=1,NWREC)                                          0043
      DO 170 N=1,NWREC                                                       0044
  170 AK(N)=AQ(N)                                                            0045
      Q=Q-1                                                                  0046
      READ (1'Q) (AQ(L),L=1,NWREC)                                           0047
      GO TO 2                                                                0048
   16 Q=Q-1                                                                  0049
      READ (1'Q) (AQ(L),L=1,NWREC)                                           0050
      GO TO 13                                                               0051
   14 GO TO 3                                                                0052
C     L                                                                      0053
    2 CONTINUE                                                               0054
C     M                                                                      0055
    3 WRITE (1'I) (AQ(L),L=1,NWREC)                                          0056
```



```
      WRITE (1'Q) (T (L),L=1,NWREC)                                          0057
      DO 300 N=1,NWREC                                                       0058
      AI(N)=AQ(N)                                                            0059
  300 AQ(N) = T(N)                                                           0060
      IF((2*Q)-(I+ J))18,18,19                                               0061
   19 LT(M)=I                                                                0062
      UT(M)=Q-1                                                              0063
      I=Q+1                                                                  0064
      READ (1'I) (AI(L),L=1,NWREC)                                           0065
      GO TO 20                                                               0066
   18 LT(M)=Q+1                                                              0067
      UT(M)=J                                                                0068
      J=Q-1                                                                  0069
   20 M=M+1                                                                  0070
      GO TO 1                                                                0071
   10 IF(I-J)21,4,4                                                          0072
   21 READ (1'J) (AJ(L),L=1,NWREC)                                           0073
C*****INSERT YOUR OWN CODE FOR NON NUMERIC KEYS                              0074
C*****OR KEYS WHICH ARE NOT THE INITIAL WORDS                                0075
C*****OF THE RECORD - FROM HERE                                              0076
      DO 210 N=1,NWKEY                                                       0077
      IF(AI(N)-AJ(N))4,210,22                                                0078
  210 CONTINUE                                                               0079
      GO TO 4                                                                0080
C*****                  - TO HERE                                            0081
   22 WRITE (1'J) (AI(L),L=1,NWREC)                                          0082
      WRITE (1'I) (AJ(L),L=1,NWREC)                                          0083
C     P                                                                      0084
    4 M=M-1                                                                  0085
      IF(M)24,24,23                                                          0086
   23 I=LT(M)                                                                0087
      READ (1'I) (AI(L),L=1,NWREC)                                           0088
      J=UT(M)                                                                0089
      GO TO 1                                                                0090
   24 RETURN                                                                 0091
      END                                                                    0092
```

```
VARIABLE ALLOCATIONS
 N     =0002  P     =0003  UT    =0013  LT    =0023  AI    =0163  AJ    =02A3  AQ    =03E3  AK    =0523  T     =0663  M     =0664
 I     =0665  J     =0666  L     =0667  I1    =0668  K     =0669  N     =066A

STATEMENT ALLOCATIONS
 1     =06AE  11    =0686  130   =0737  13    =0741  15    =0747  150   =0758  17    =0762  170   =0794  16    =07C3  14    =07E1
 2     =07E3  3     =07EB  300   =0826  19    =0845  18    =0874  20    =0888  10    =0893  21    =0899  210   =08C0  22    =08CA
 4     =08F8  23    =0902  24    =0929

FEATURES SUPPORTED
 ONE WORD INTEGERS

CALLED SUBPROGRAMS
 SUBSC    SUBIN    SDRED    SDWRT    SDCON    SDIX

INTEGER CONSTANTS
     1=066E       2=066F

CORE REQUIREMENTS FOR DSRT2
 COMMON       0  VARIABLES    1646  PROGRAM     702

END OF COMPILATION
```

```
// DUP
*STORE       WS  UA  DSRT2                                          0093
1A92 002D                                                          0094
```

```
// FOR                                                                    0001

                                                                          PAGE 01
*     NAME MAINL                                                          0002
*     ONE WORD INTEGERS                                                   0003
*     IOCS (CARD,1132PRINTER,DISK)                                        0004
*     LIST ALL                                                            0005
```

```
                                                                          PAGE 02
      INTEGER                TEMP(39)
      DEFINE FILE 1 (500,39,U,IPT)                                        0006
C     FIRST CARD TELLS HOW MANY CARDS FOLLOW                              0007
      READ (2,90) NRFIL                                                   0008
      WRITE (3,92)                                                        0009
C     LOOP READS CARD,WRITES RECORD ON DISK AND PRINTS RECORD             0010
      DO 1 I=1,NRFIL                                                      0011
      READ (2,90) N1,N2,(TEMP(J),J=1,37)                                  0012
      WRITE (1'I) N1,N2,(TEMP(J),J=1,37)                                  0013
    1 WRITE (3,91)N1,N2,(TEMP(J),J=1,37)                                  0014
C     SORT                                                                0015
      NWKEY=2                                                             0016
      NWREC=39                                                            0017
      CALL DSRT2 (NRFIL,NWREC,NWKEY)                                      0018
C     PRINT SORTED FILE                                                   0019
      WRITE (3,93)                                                        0020
      DO 2 I=1,NRFIL                                                      0021
      READ (1'I) N1,N2,(TEMP(J),J=1,37)                                   0022
    2 WRITE (3,91) N1,N2,(TEMP(J),J=1,37)                                 0023
      CALL EXIT                                                           0024
   90 FORMAT (2I3,37A2)                                                   0025
   91 FORMAT (1H ,I3,1X,I3,1X,37A2)                                       0026
   92 FORMAT (48H1RECORDS PRINTED AS READ                                 0027
   93 FORMAT (48HORECORDS PRINTED AS SORTED                               0028
      END                                                                 0029
                                                                          0030
```

```
VARIABLE ALLOCATIONS                                                    PAGE 03
 TEMP =002E  IPT  =002F  NRFIL=0030  I    =0031  N1   =0032  N2   =0033  J    =0034  NWKEY=0035  NWREC=0036

STATEMENT ALLOCATIONS
 90   =003D  91   =0042  92   =004B  93   =0065  1    =00D6  2    =0128

FEATURES SUPPORTED
 ONE WORD INTEGERS
 IOCS

CALLED SUBPROGRAMS
 DSRT2   FLD     FSTO    SRED    SWRT    SCOMP   SFIO    SIOIX   SIOI    SUBSC   CARDZ   PRNTZ   SDFIO   SDRED   SDWRT
 SDCOM   SDIX    SDI

INTEGER CONSTANTS
     2=0038      3=0039      1=003A     37=003B     39=003C

CORE REQUIREMENTS FOR MAINL
 COMMON      0  VARIABLES     56  PROGRAM     276

END OF COMPILATION
```

```
// XEQ

                                                                        0001

RECORDS PRINTED AS READ
  2    2 SORT ROUTINES WOULD BE IF THE RESTRICTIONS ARE OBSERVED.                0003
  2    1 FOR UPDATING OR REPORT PREPARATION. THEY CAN HOWEVE BE USED AS ANY      0004
  1    2 SO THAT ONE CAN FIND RECORDS IN A LARGER FILE IN PROPER SEQUENCE        0005
  1    1 THESE SORTS ARE INTENDED FOR SORTING KEYS AND TAGS  RECORD NUMBERS      0006

RECORDS PRINTED AS SORTED
  1    1 THESE SORTS ARE INTENDED FOR SORTING KEYS AND TAGS  RECORD NUMBERS      0006
  1    2 SO THAT ONE CAN FIND RECORDS IN A LARGER FILE IN PROPER SEQUENCE        0005
  2    1 FOR UPDATING OR REPORT PREPARATION. THEY CAN HOWEVE BE USED AS ANY      0004
  2    2 SORT ROUTINES WOULD BE IF THE RESTRICTIONS ARE OBSERVED.                0003
```