UR1982-18

1/5

1982/18. A FORTRAN program for sorting areally distributed data to minimise plotting time.

R.G. Richardson

Abstract

This program sorts randomly distributed sample data into north-south strips and then orders the data within each strip to have increasing or decreasing north co-ordinates to minimise plotter pen movements.

THE PROGRAM

SORTST (Appendix 1)

The program reads the east co-ordinates of the sample points from a random access disk file and uses an indexed sort to order the data to have increasing east co-ordinates. The ordered data is divided into north-south strips of user-specified width. For each strip, the north co-ordinates of the sample points are read, sorted into increasing or decreasing order as appropriate, and the original sample data for that strip copied to the output file in the new order.

Control data is read from logical unit 5, the sample data is read from logical unit 4 (random access) and the sorted file is on logical unit 6.

Non-standard subroutines are:

SYSCOM (I, 'AS 0899*') - assign logical unit 8 to the core buffer.

FREEIN (ANUM, IVAL, IER, ILUN, ICC) - read IVAL real numbers in free format from logical unit ILUN and store them in ANUM. If there is an error, IER is TRUE and ICC is the actual number of values read.

POSITN (ILUN, IREC) - position logical unit ILUN ready to read or write record IREC.

IREC = 0, 1, 2, ..., N-1

Although a file may be random access, sequential access is possible by positioning the file using REWIND or POSITN and then accessing the file using normal READ or WRITE statements.

Control data read from logical unit 5 is:

XMIN - the western edge of the area containing the samples (km)
XNOM - the width of the north-south strips (km)

Data input from logical unit 4 is:

X, Y - format (10X, 2F9.1)
- the east and north co-ordinate of the sample point in metres.

Data output to logical unit 6 is:

An exact copy of the first 80 characters of the corresponding record on logical unit 4.

[26 May 1982]

APPENDIX 1

Program SORTST

```
TITL  SORTST - SORTING STATIONS INTO GEOGRAPHICAL ORDER
 A PROGRAM TO SORT AREALLY DISTRIBUTED DATA TO OPTIMISE PLOTTING THE DATA
 THE DATA IS SORTED INTO ASCENDING X-COORDINATE VALUES.  N-S STRIPS ARE
 TAKEN AND THE STATIONS WITHIN EACH STRIP SORTED INTO ORDERED Y VALUES
 AND THEN OUTPUT IN THIS ORDERED FORM.
 INPUT DATA   LU 4    RANDOM ACCESS
 OUTPUT DATA  LU 6
 CONTROL DATA LU 5
 ASSUMES DATA IS ON A NOMINAL GRID AND READS THE STARTING GRID X VALUE
 AND THE NOMINAL X SPACING FROM LU 5
     IMPLICIT INTEGER*2 (I-N)
     INTEGER*2 INDEX(5000)
     REAL X(5000),ALINE(20),ANUM(2)
     LOGICAL UP,IER
     CALL SYSCOM(I,'AS 0899*')
 ASSIGN THE CORE BUFFER TO LU 8 FOR USE WITH FREEIN
     WRITE(5,200)
 200 FORMAT(' ENTER XMIN AND XNOM IN KM.'/' XMINNNNNN XNOMMMMMM')
     CALL FREEIN(ANUM,2,IER,5,ICC)
     XMIN=ANUM(1)
     XNOM=ANUM(2)
 HAVE STARTING VALUE AND NOMINAL GRID SPACING
     XMIN=XMIN*1000.0
     XNOM=XNOM*1000.0
 CONVERT TO METRES
     REWIND 4
 REWIND DATA FILE
     N=1
 COUNTER FOR NUMBER OF DATA VALUES
  10 READ(4,100,END=11) X(N)
 100 FORMAT(10X,F9.1)
 STORE AWAY
     INDEX(N)=N
 FILL SORT INDEX ARRAY
     N=N+1
 INCREMENT N
     IF (N .EQ. 5001) STOP 888
 TOO MANY POINTS
     GOTO 10
 BACK ROUND

 NOW SORT INTO ASCENDING X VALUES
  11 N=N-1
 COUNT WAS PRE-INCREMENTED
     CALL BUBSTR(INDEX,X,1,N)
 PERFORM THE SORT

 NOW DIVIDE INTO STRIPS XNOM WIDE AND SORT ON Y VALUES WITHIN THE STRIPS
     UP=.TRUE.
 USED TO OPTIMISE PEN MOVEMENT BY ORDERING OUTPUT IN INCREASING Y VALUE
 THEN IN DECREASING Y VALUE
     XMIN=XMIN-0.5*XNOM
 WANT TO CENTRE STRIPS ABOUT THE NOMINAL POSITION
     IBOT=1
  15 CONTINUE
     XUP=XMIN+XNOM
 EASTERN SIDE OF STRIP
     J=0
 COUNTER FOR NUMBER OF STATIONS IN STRIP
     DO 20 I=IBOT,N
     INDX=INDEX(I)
     IF (X(INDX) . GT.  XUP) GOTO 22
THIS IS THE FIRST STATION TO THE EAST OF THE STRIP
     J=J+1
     CALL POSITN(4,INDX-1)
FIND THE DATA RECORD USING A RANDOM SEEK
     READ(4,101) X(INDX)
101 FORMAT(19X,F9. 1)
REPLACE THE X VALUE WHICH IS NOW REDUNDANT BY THE Y VALUE
 20 CONTINUE
     I=N+1
ONLY GET TO HERE WHEN HAVE SCANNED ALL DATA ON THE FILE
 22 CONTINUE
     IF (J . EQ.  0) GOTO 24
NO POINTS IN THIS STRIP

IF POINTS IN THE STRIP ORDER THEM IN Y
     CALL BUBSTR(INDEX(IBOT),X,1,J)
SORT
     JBOT=IBOT-1
     INC=+1
PARAMETERS FOR OUTPUTTING IN ASCENDING Y
     IF (UP) GOTO 21
     JBOT=IBOT+J
     INC=-1
SET PARAMETERS FOR SCANNING DOWN IN Y FOR OUTPUT
 21 CONTINUE
     DO 23 K=1,J
     JBOT=JBOT+INC
     CALL POSITN(4,INDEX(JBOT)-1)
FIND THE RECORD IN THE ORIGINAL FILE TO COPY THE COMPLETE
RECORD TO THE OUTPUT FILE
     READ(4,102) ALINE
     WRITE(6,102) ALINE
102 FORMAT(20A4)
COPY HERE
 23 CONTINUE
     UP=. NOT.  UP
CHANGE THE OUTPUTTING DIRECTION
 24 IBOT=I
SET BASE FOR NEXT STRIP SEARCH
     IF (I . GT.  N) GOTO 25
BEEN THROUGH ALL STATIONS SO TIDY UP
     XMIN=XUP
MOVE WEST EDGE OF STRIP OVER
     GOTO 15
ROUND AGAIN

TIDY UP
 25 CONTINUE
     ENDFILE 6
     STOP
     END
     SUBROUTINE BUBSTR(IR,A,IBASE,N)
     REAL A(N)
     INTEGER*2 IR(N),IBASE,N,I,J,IRI,IRIP1,NM1,IP1
     LOGICAL NSWAP
     IF (N . LE.  1) RETURN
NOTHING TO SORT
     NM1=N-1
```

```
   DO 30 J=IBASE,NM1
   NSWAP=.TRUE.
   IRI=IR(1)
   DO 40 I=IBASE,NM1
   IP1=I+1
   IRIP1=IR(IP1)
   IF (A(IRI) .LE. A(IRIP1)) GOTO 40
   NSWAP=.FALSE.
   IR(I)=IRIP1
   IR(IP1)=IRI
   IRIP1=IRI
40 IRI=IRIP1
   IF (NSWAP) RETURN
30 CONTINUE
   RETURN
   END
```