Algorithms supplement

Algorithm 106

AN ALGORITHM FOR UPDATING AN ORDERED LIST ON A VDU

A. Brown
formerly with British Rail Research Department,
currently with Pactel Computer Consultants

1. Introduction

Many visual display units (VDUs) have the capability to carry out certain text editing procedures locally on receipt of a single character command from the central computer, e.g. delete line, clear screen, etc. This paper describes an algorithm which makes use of some of these text editing facilities to reduce the amount of output required from the central computer to update an ordered list.

2. VDU characteristics

The following capabilities of the VDU are used in the application of the algorithm:

1. *Delete line*
Any line may be deleted by positioning the cursor at the start of the line and issuing a delete line command. All lines below the line deleted are moved up one line and a blank line is added at the bottom of the screen.

2. *Insert line*
A blank line may be inserted by positioning the cursor at the start of the appropriate line, and issuing an insert line command. The line addressed and all lower lines are moved down one line; the line at which the cursor is positioned becomes blank and the old bottom line of the screen is lost.

3. *Other capabilities*
Use is also made of cursor positioning functions.

4. *Reversal*
Clearly, using only the above local editing facilities, the order of two lines on the screen cannot be reversed. This process is achieved by deleting one of the lines and retransmitting it in full from the computer.

3. Nature of the information displayed

The algorithm is of use in applications in which the information on the screen changes only slowly, with most of the information on one frame being the same on the next. However, the position of any given line of information can move up or down between one frame and its successor.

4. Algorithm

Given two VDU frames: an existing frame and a required frame, each consisting of n lines of information, then we will construct two sets

$$\{D_i : i = 1 \text{ to } n\} \quad \{W_i : i = 1 \text{ to } n\}$$

for use in transforming the existing frame to the required frame

$$D_i = \begin{cases} 0 \text{ if line } i \text{ of the existing frame is not to be retained locally} \\ 1 \text{ otherwise} \end{cases}$$

$$W_i = \begin{cases} \text{if line } i \text{ of the required frame has not been retained locally} \\ \text{from the existing frame} \\ 1 \text{ otherwise} \end{cases}$$

5. Method

Each line, both of the existing frame and the required frame, is epitomised by a value E such that

$E_p = E_q$ if, and only if, line p is identical to line q

(The method of construction of these epitomes is not dealt with in this paper, but in practice a simple approximation was obtained by summing the characters in the line, with each set of three characters being treated as an integer).

Let θ_i be the epitome of line i on the existing frame, $i = 1$ to n
Let N_i be the epitome of line i on the required frame, $i = 1$ to n

Step 1
Set $D_i = W_i = 0$ for $i = 1$ to n

Step 2
Set $L_i = \begin{cases} j - i + n \text{ if } N_j = \theta_i, \text{ and } N_k \neq \theta_i \text{ for all } k < j \\ 0 \text{ otherwise} \end{cases}$
for
$i = 1$ to n
i.e. set $L_i = n +$ displacement between line i on the existing frame and its first occurrence on the required frame.
e.g. $L_i = n$ if line i is the same on both frames.

Step 3
If $L_i = 0$ then $\{D_i\}$ and $\{W_i\}$ are complete.
for $i = 1$ to n
Otherwise, find the most common occurring, non-zero value, K, in $\{L_i\}$. If two or more exist, then arbitrarily select the first.

Step 4
If every $L_i = K$ has been considered, then repeat Step 3.
Otherwise for each $L_i = K$
(*a*) set $D_i = 1$
set $W_{K+i-n} = 1$
(*b*) set $L_i = 0$
(*c*) For all $j < i$ such that $L_j \geqslant i - j + K$ set $L_j = 0$
For all $j > i$ such that $L_j \leqslant i - j + K$ set $L_j = 0$
Step 4 (*c*) is required to exclude reversals as described under VDU characteristics.

6. Use of Sets $\{D_i\}$ and $\{W_i\}$

Now $\{D_i\}$ and $\{W_i\}$ can be used in the following way:

Deletion of lines not required
(*a*) Position cursor at top of screen.
(*b*) For each D_i; $i = 1$ to $n - 1$
If $D_i = 0$ delete line otherwise skip line.

Insertion of new lines
(*a*) Position cursor at top of screen
(*b*) For each W_i, $i = 1$ to n
If $W_i = 0$ then insert line, write required line i otherwise skip line.

In fact the deletions and insertions can be combined in any way, providing that the cumulative total of insertions at any stage does not exceed the corresponding cumulative total of deletions (as this would result in losing lines at the bottom of the screen). Thus an additional saving could be made by reducing the number of skip line instructions.

7. Examples ($n = 12$)

(*a*) $\{\theta_i\} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$
$\{N_i\} = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$
$\{D_i\} = \{0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$
$\{W_i\} = \{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0\}$
(*b*) $\{\theta_i\} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$
$\{N_i\} = \{4, 6, 3, 7, 1, 2, 5, 8, 10, 11, 9, 13\}$
$\{D_i\} = \{1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0\}$
$\{W_i\} = \{0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0\}$

(*c*) $\{\theta_i\} = \{4, 6, 3, 7, 1, 2, 5, 8, 10, 11, 9, 13\}$
$\{N_i\} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$
$\{D_i\} = \{1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0\}$
$\{W_i\} = \{0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0\}$

8. Computer implementation

This algorithm has been implemented in FORTRAN with $n = 12$. This forms part of a British Rail train control system using a GEC 905 minicomputer connected to VDT 4000 visual display units.

The number of characters transmitted from the computer to the VDU, using the algorithm, is given by the following formula:

Number of characters = 1 208 − 96*r*, where *r* is the number of lines of the existing frame retained (= number of 1's in $\{D_i\}$).

r	0	1	2	3	4	5	6
No of chars.	1208	1112	1016	920	824	728	632
r	7	8	9	10	11	12	
No of chars.	536	440	344	248	152	56	

Fig. 1

The number of characters required to transmit a full screen is 868. Hence if $0 \leqslant r \leqslant 3$ a full screen is transmitted, otherwise lines are deleted and inserted. (If $r = 12$ then the existing and required frames are identical, and no output is required).

9. Limitations

The algorithm is inefficient in cases where there are repeated lines since it always attempts to match a line with its first occurrence on the new screen. Even when all the lines are distinct, the algorithm may not produce the minimum number of changes (*cf* examples (*b*) and (*c*) above: Ex (*c*) is the inverse of Ex (*b*)).

10. Conclusion

From **Fig. 1** it can be seen that substantial savings can be achieved if the number of changes, between the existing frame and the required frame, is small. A more sophisticated algorithm could be devised to cope with the limitations indicated above.

For VDUs with insert character and delete character facilities analogous to the insert line and delete line above, a similar application could be made to characters within each line. The saving produced would be on a much smaller scale except in particularly suitable cases.

Acknowledgement
The author wishes to thank the British Railways Board for permission to publish this paper.

Appendix 1 Use of the computer program

The algorithm is programmed as a subroutine with 9 parameters:

1. IOND one-dimensional integer array of length $N = \{\theta_i\}$
2. NIND one-dimensional integer array of length $N = \{N_i\}$
3. LOINE one-dimensional integer array of length $N = \{D_i\}$
4. NLINE one-dimensional integer array of length $N = \{W_i\}$
5. NOA integer variable
6. NDISP one-dimensional integer array of length *N* used as workspace
7. NDIF one-dimensional integer array of length *N*2 used as workspace
8. N integer variable
9. N2 integer variable

Before calling the subroutine, the user must set up arrays IOND and NIND (see description of sets $\{\theta_i\}$ and $\{N_i\}$ in Section 5) and variables *N* (= number of lines per screen) and *N*2 (= twice *N*).

On return from the subroutine, the arrays LOINE and NLINE contain the retention indicators—see description of sets $\{D_i\}$ and $\{W_i\}$ in Section 4. NOA is set equal to the number of lines retained from the old screen to the new.

```
      SUBRØUTINE VDALGA(IØND,NIND,LØINE,NLINE,NØA,NDISP,NDIF,N,N2)
C 11/9/78
C SUBRØUTINE TØ REDUCE VDU ØUTPUT
C
C ØN ENTRY LØINE CØNTAINS THE EPITØMES ØF THE ØLD SCREEN
C          NLINE CØNTAINS THE EPITØMES ØF THE NEW SCREEN
C          N=THE NUMBER ØF LINES PER SCREEN;  N2=2*N
C ØN EXIT IØND CØNTAINS THE INDICATØRS ØF LINES ØF THE ØLD SCREEN
C            TØ BE RETAINED; 0=DØ NØT RETAIN, 1=RETAIN.
C         NIND CØNTAINS THE INDICATØRS ØF LINES ØF THE NEW
C            SCREEN WHICH NEED NØT BE RETRANSMITTED; 0=RETRANSMIT
C            1=DØ NØT RETRANSMIT.
C         NØA= THE TØTAL NUMBER ØF LINES RETAINED.
C
C NDISP AND NDIF ARE USED AS WØRKSPACE
C
      DIMENSIØN IØND(N),NIND(N),LØINE(N),NLINE(N),NDISP(N),NDIF(N2)
      NØA=0
C SET RETENTIØN ARRAYS TØ ZERØ
      DØ 100 I=1,N
      IØND(I)=0
100   NIND(I)=0
      DØ 400 I=1,N
200   DØ 300 J=1,N
C CØMPUTE DISPLACEMENT ØF LINES FRØM ØLD TØ NEW SCREEN
      IF(LØINE(I).EQ.NLINE(J))GØ TØ 320
300   CØNTINUE
C 0 MEANS LINE DØES NØT APPEAR ØN NEW SCREEN
      NDISP(I)=0
      GØ TØ 400
C
C NØN-ZERØ MEANS LINE APPEARS ØN NEW SCREEN
320   NDISP(I)=J-I+N
400   CØNTINUE
C
C CLEAR DISPLACEMENT-CØUNT-ARRAY NDIF AND CØUNT NPREH
450   DØ 500 I=1,N2
500   NDIF(I)=0
      NPREH=0

C
C FIND MØST CØMMØN DISPLACEMENT
      DØ 600 I=1,N
      IDISP=NDISP(I)
      IF(IDISP.EQ.0)GØ TØ 600
      ICH=NDIF(IDISP)+1
C UPDATE CØRRESP CØUNT
      NDIF(IDISP)=ICH
      IF(ICH.LE.NPREH)GØ TØ 600
C CHECK IF NEW MØST CØMMØN
      NPREH=ICH
      IMDIF=IDISP
600   CØNTINUE
C
C STØP WHEN NØ MØRE LINES ARE RETAINED
      IF(NPREH.EQ.0)RETURN
      NØA=NØA+NPREH
C
C MARK LINES RETAINED AND DELETE LINESWHICH CANNØT
C BE RETAINED BECAUSE ØF ØRDER
      DØ 900 I=1,N
      IF(NDISP(I).NE.IMDIF)GØ TØ 900
      IDISP=I+IMDIF
C MARK I-TH LINE ØF ØLD SCREEN FØR RETENTIØN
      IØND(I)=1
C MARK CØRRESPØNDING LINE ØF NEW SCREEN AS RETAINED
      IDISPN=IDISP-N
      NIND(IDISPN)=1
C REMØVE REVERSALS AND ALL LINES MAPPED ØNTØ THIS LINE
      DØ 700 J=1,I
      IF(NDISP(J).GE.IDISP-J)NDISP(J)=0
700   CØNTINUE
      DØ 800 J=I,N
      IF(NDISP(J).LE.IDISP-J)NDISP(J)=0
800   CØNTINUE
900   CØNTINUE
      GØ TØ 450
      END
```