A PROGRAM TO SHORTEN A SEQUENCE OF SETS OF INTEGER NUMBERS

1. The Program.

begin

comment There is given a sequence of integer numbers
$z[1], z[2], \ldots, z[x]$. The sequence contains $iz$ sets $Z_k \ (k = 1, 2, \ldots, iz)$. The set $Z_k$ consists of the following elements:
$z[azd[k]], z[azd[k]+1], \ldots, z[azg[k]]$.

The program produces the sequence $C$ that consists of the following elements:
$c[1], c[2], \ldots, c[ile]$,

and is the shortest possible of those having the property that every set $Z_k$ is a segment of it (the initial arrangement of the elements of $Z_k$ is not necessarily preserved in the corresponding segment of $C$). The program reads in the following values:
$iz, x,$
$azd[1], azg[1],$
$azd[2], azg[2],$
$\ldots$
$azd[iz], azg[iz],$
$z[1], z[2], \ldots, z[x].$

After the sequence $C$ is formed the program prints out the following:
$iz, ile,$
$azd1[1], azg[1],$
$azd1[2], azg[2],$
$\ldots$
$azd1[iz], azg[iz],$
$c[1], c[2], \ldots, c[ile],$

where $azd1[k]$ and $azg[k] \ (k = 1, 2, \ldots, iz)$ are the subscripts of the first and the last elements of that segment of the sequence $C$ which corresponds to $Z_k$;
integer \( k, k_1, k_2, i, i_1, j, j_1, g, g_1, wz, wz_1, wz_2, ni, iz, iz_1, x, maxw, maxi, ms_1, ngg, mkrok, ile, dy, d_1, d_2, krok, dgg, gg, iw, iw_2, iw_3, dgg_1, s_1, mi_1; \\
boolean \( b, b_1, b_2, b_3, b_4, b_5, mb_1, mb_2; \\
ininteger (I, iz); \\
ininteger (I, x); \\
begin \\
procedure iloczyn \( g, g_1; \) \\
value g, g_1; \\
integer g, g_1; \\
begin \\
if \( g_1 < g \) \\
then \\
begin \\
krok := d_1 := 1; \\
d_2 := 0 \\
end \( g_1 < g \) \\
else \\
begin \\
krok := -1; \\
d_1 := 0; \\
d_2 := 1 \\
end \( g_1 > g; \) \\
b := false; \\
dgg := d[g] - d_1; \\
gg := g; \\
iw := 0; \\
f: \\
gg := gg - krok; \\
dgg_1 := d[gg] - d_2; \\
b_2 := false; \\
iw_2 := 0; \\
j_1 := dgg + krok; \\
for k := dgg_1 step krok until dgg do \\
begin \\
ax := e[k]; \\
for k_1 := wz step 1 until wz_1 do \\
if x = z[k_1] \\
then \\
begin \\
iw_2 := iw_2 + 1; \\
if b
Algorithm 8

then
for $k2 := j1$ step $krok$ until $dg$ do
  if $x = c[k2]$
    then
      begin
        $iw2 := iw2 - 1$;
        go to e
      end $x = c[k2], k2, b$;
  go to e
end $x = z[k1], k1$;
$b2 := true$;
e:
end $k$;
iw := iw + iw2;
b3 := iw $\neq ni$;
b4 := gg $\neq g1$;
if $b$
  then
  begin
    if $\neg b2 \land b4$
      then go to e1;
  end $b3$
then
begin
if $b5$
  then $iw := 0$
else
  if $b1$
    then $iw := iw3$
end $b3$;
if $maxw < iw$
then
begin
  $maxw := iw$;
  $maxi := i$;
  $mi1 := i1$;
  $ms1 := s1$;
  $mkrok := krok$;
if $iw = iw3$
  then
    begin
      $mb2 := b1$;
      $mb1 := false$;
      $mgg := s1 - krok$
    end
end

end iw = iw3
else
begin
    mb2 := b2;
    mb1 := b1;
    mgg := gg
end iw ≠ iw3
end maxw < iw;
if ¬ b4 ∨ krok < 0
then go to e4;
if ¬ b3
then go to koniec;
iw := iw2;
b := false
end b;
if iw2 ≠ 0
then
begin
    b := true;
    b1 := b2;
    s1 := gg + krok;
    b5 := s1 ≠ g;
    dg := dgg;
    iw3 := iw2;
    if ¬ (b4 ∧ b3)
then go to e2
end iw2 ≠ 0;
e1:  dgg := dgg1 − krok;
    if b4
then go to f;
e4: end iloczyn;
integer array d[1 : 4 × iz], c[1 : x + x], z[1 : x],
nr, azd, azg, azd1[1 : iz];
g := g1 := iz + iz;
d[g] := x;
ilc := iz1 := 0;
for i := 1 step 1 until iz do
begin
    ininteger (I, j);
ininteger (I, k);
    azd[i] := azd1[i] := j;
    azg[i] := k;
\[ d[i] := k - j \]
end \( i \);
for \( i := 1 \) step 1 until \( iz \) do
begin
\[ \text{maxw} := -1; \]
for \( j := 1 \) step 1 until \( iz \) do
begin
\[ x := d[j]; \]
if \( \text{maxw} < x \)
begin
\[ k := j; \]
\[ \text{maxw} := x \]
end \( \text{maxw} < x \)
end \( j \);
if \( \text{maxw} < 0 \)
then go to \( zu \);
\[ iz1 := iz1 + 1; \]
\[ nr[iz1] := k; \]
\[ d[k] := -1 \]
end \( i \);
\[ zu: \]
for \( i := 1 \) step 1 until \( x \) do
begin
\[ \text{ininteger} \ (1, z[i]); \]
go to \( new \);
end \( i \);
\[ ppi: \]
\[ \text{maxw} := 0; \]
for \( i1 := 1 \) step 1 until \( iz1 \) do
begin
\[ i := nr[i1]; \]
if \( i > 0 \)
then begin
\[ wz := azd[i]; \]
\[ wz1 := azg[i]; \]
\[ ni := wz1 - wz + 1; \]
if \( ni < \text{maxw} \)
then go to \( koniec \);
\[ iloczyn \ (g, g1); \]
\[ iloczyn \ (g1, g) \]
end \( i > 0 \)
end \( i1 \);
koniec:
   if maxw = 0
    then
    begin
       dg := d[g] - 1;
       dgg := d[gl];
       k2 := dgg - ilc - 1;
       for i := 1 step 1 until iz1 do
        begin
           k := - nr[i];
           if k > 0
           then
           begin
            if azd1[k] = 0
            then
            begin
             azd1[k] := azd[k] - k2;
             azg[k] := azg[k] - k2
            end
            azd1[k] = 0
           end
           k > 0
        end i;
       for i := dgg step 1 until dg do
        begin
         ilc := ilc + 1;
         c[ilc] := c[i]
        end i;
       gl := g;
    end i;
new:
    maxw := 0;
   for i := 1 step 1 until iz1 do
    begin
       k := nr[i];
       if k < 0
       then go to endi;
       k1 := azd[k];
       k2 := azg[k];
       if k2 - k1 < maxw
       then go to endnew;
       for j := i + 1 step 1 until iz1 do
        begin
         jl := nr[j];
         if jl < 0
         then go to endj;
        wz := azd[j1];
\( w1 = azg[j1]; \)
\[\text{if } w1 - wz < \text{maxw} \]
\[\text{then go to endi;} \]
\[wz = 0; \]
\[\text{for } d1 = k1 \text{ step } 1 \text{ until } k2 \text{ do} \]
\[\text{begin} \]
\[x = z[d1]; \]
\[\text{for } d2 = wz \text{ step } 1 \text{ until } wz1 \text{ do} \]
\[\text{if } x = z[d2] \]
\[\text{then} \]
\[\text{begin} \]
\[iw = iw + 1; \]
\[\text{go to endd1} \]
\[\text{end } x = z[d2], d2; \]
\[\text{endd1:} \]
\[\text{end } d1; \]
\[\text{if } iw > \text{maxw} \]
\[\text{then} \]
\[\text{begin} \]
\[\text{maxw} = iw; \]
\[\text{mi1} = i \]
\[\text{end } iw > \text{maxw}; \]
\[\text{endj:} \]
\[\text{end } j; \]
\[\text{endi:} \]
\[\text{end } i; \]
\[\text{endnew:} \]
\[\text{if } \text{maxw} = 0 \]
\[\text{then} \]
\[\text{begin} \]
\[\text{for } i = 1 \text{ step } 1 \text{ until } iz1 \text{ do} \]
\[\text{begin} \]
\[k = nr[i]; \]
\[\text{if } k > 0 \]
\[\text{then} \]
\[\text{begin} \]
\[wz = azd[k]; \]
\[wz1 = azg[k]; \]
\[azd1[k] = ile + 1; \]
\[azg[k] = ile + 1 + wz1 - wz; \]
\[\text{for } j = wz \text{ step } 1 \text{ until } wz1 \text{ do} \]
\[\text{begin} \]
\[ile = ile + 1; \]
\[c[ile] = z[j] \]
\[\text{end } j \]
\[\text{end } k > 0 \]

Zastosowania Matematyki XI. 3.
end i;
go to druk
end maxw = 0
else
begin
maxi := nr[mi1];
wz := azd[maxi];
wz1 := azg[maxi];
d1 := k := d[g1];
g := g + 1;
for i := wz step 1 until wz1 do
begin
c[k] := z[i];
k := k + 1
end i;
d2 := k - 1;
d[g] := k;
go to endmax
end maxw ≠ 0
end maxw = 0
else
begin
d1 := if mkrok < 0 then 0 else 1;
d2 := abs(mkrok) - d1;
wz := azd[maxi];
wz1 := azg[maxi];
ni := wz1 - wz + 1;
k1 := d[mgg] - d2;
j1 := d[ms1] - d1;
b := maxw ≠ ni;
if mb2
then
begin
k2 := d[mgg + mkrok] - d1;
for wz2 := wz step 1 until wz1 do
begin
x := z[wz2];
for k := k1 step mkrok until k2 do
if x = e[k]
then
begin
c[k] := c[k2];
c[k2] := x;
end
\( k2 := k2 - \text{mkrok}; \)
\( \text{go to } e \)
\( \text{end } x = c[k], k; \)
\( e: \)
\( \text{end } \text{wz}2; \)
\( k1 := k2 + \text{mkrok} \)
\( \text{end } \text{mb}2; \)
\( \text{if } \text{mb}1 \)
\( \text{then} \)
\( \text{begin} \)
\( k2 := d[ms1 - \text{mkrok}] - d2; \)
\( \text{for } wz2 := wz \text{ step 1 until } wz1 \text{ do} \)
\( \begin{align*}
& \text{begin} \\
& x := z[wz2]; \\
& \text{for } k := k2 \text{ step mkrok until } j1 \text{ do} \\
& \quad \text{if } x = c[k] \\
& \quad \quad \text{then} \\
& \quad \quad \begin{align*}
& c[k] := c[k2]; \\
& c[k2] := x; \\
& k2 := k2 + \text{mkrok}; \\
& \text{go to } e1 \\
& \quad \text{end } x = c[k], k; \\
& \end{align*} \\
& \text{end } wz2; \\
& j1 := k2 - \text{mkrok}; \\
& \text{end } \text{mb}1 \\
& \text{else} \)
\( \text{if } b \)
\( \text{then} \)
\( \text{for } wz2 := wz \text{ step 1 until } wz1 \text{ do} \)
\( \begin{align*}
& \text{begin} \\
& x := z[wz2]; \\
& \text{for } k := k1 \text{ step mkrok until } j1 \text{ do} \\
& \quad \text{if } x = c[k] \\
& \quad \quad \text{then go to } e2; \\
& \quad j1 := j1 + \text{mkrok}; \\
& \quad c[j1] := x; \\
& \end{align*} \\
& \text{end } wz2, b, \neg \text{mb}1; \\
& mgg := mgg + \text{mkrok}; \\
& \text{if } \text{mkrok} < 0 \\
& \text{then} \\
& \text{begin} \\
& d1 := j1; \\
& d2 := k1; \\
& \end{align*} \)
if \ b
    then
    begin
        \textit{gi} := \textit{gi} - 1;
        \textit{d}[\textit{gi}] := \textit{ji}
    end \ b
    else
    if \ mbl
        then
        begin
            for \ k := \textit{gi} \ step 1 \ until \ msl \ do
                \textit{d}[\textit{k} - 1] := \textit{d}[\textit{k}];
                \textit{d}[\textit{msl}] := \textit{ji};
                \textit{gi} := \textit{gi} - 1
        end \ mbl, \ \neg \ b;
    if \ mbl
        then
        begin
            for \ k := \textit{gi} \ step 1 \ until \ \textit{mgg} \ do
                \textit{d}[\textit{k} - 1] := \textit{d}[\textit{k}];
                \textit{gi} := \textit{gi} - 1;
                \textit{d}[\textit{mgg}] := \textit{kl} + 1
        end \ mbl
    end \ mkrok < 0
    else
    begin
        \textit{d1} := \textit{kl};
        \textit{d2} := \textit{ji};
        if \ b
            then
            begin
                \textit{g} := \textit{g} + 1;
                \textit{d}[\textit{g}] := \textit{ji} + 1
            end \ b
            else
            if \ mbl
                then
                begin
                    for \ k := \textit{g} \ step - 1 \ until \ msl \ do
                        \textit{d}[\textit{k} + 1] := \textit{d}[\textit{k}];
                        \textit{d}[\textit{msl}] := \textit{ji} + 1;
                        \textit{g} := \textit{g} + 1
                end \ mbl, \ b;
if \( mb2 \) then begin
   \( k := g \) step \( -1 \) until \( mgg \) do
   \( d[k+1] := d[k] \);
   \( g := g + 1 \);
   \( d[mgg] := k1 \);
end \( mb2 \)
end \( mkrok \geq 0 \)
end \( maxw \neq 0 \);
endmax:
\( nr[miI] := -maxi \);
\( azd1[maxi] := 0 \);
\( azd[maxi] := d1 \);
\( azg[maxi] := d2 \);
go to \( ppi \);
druk:
\( outinteger(1, iz) \);
\( outinteger(1, ilc) \);
for \( i := 1 \) step \( 1 \) until \( iz \) do
begin
   \( outinteger(1, azdI[i]) \);
   \( outinteger(1, azg[i]) \)
end \( i \);
for \( i := 1 \) step \( 1 \) until \( ilc \) do
\( outinteger(1, e[i]) \)
end
end program

2. An application of the program. The problem of the economy of storage of a family of sets containing common elements was encountered during the preparation of the ODRA-ALGOL compiler (a hardware representation of Algol 60 for the ODRA 1204 computer). If the sets discussed here are understood as subtables of the Compactified Reducing Transition Tables (CRTT) of \([1]\), then the immediate application of the program made it possible to shorten the Final Table (see \([1]\)) by 335 locations. After the author of \([1]\) had redefined the CRTT the application of the program has resulted in a gain of 862 locations, i.e., 12 per cent of the compiler size.

3. The method employed. Let the identifiers with subscripts not enclosed in square brackets denote sets. All the sets are supplied with single subscripts only. If therefore a subscript consists of several symbols (i.e., letters and/or digits) then it denotes an identifier used in the program.
The identifiers with subscripts enclosed in square brackets denote elements of a set or of a sequence.

Let \( Z_1, Z_2, \ldots, Z_{iz} \) be the family of sets we are going to discuss, and let the shortened sequence of sets \( Z_k \) be denoted by the letter \( C \). To every set \( Z_k \) added to the sequence \( C \) there is assigned a pair of numbers \( azdI[k], azg[k], \)
in such a way that

\[
Z_k = \{c|azdI[k]|, c|azdI[k]+1|, \ldots, c|azg[k]|\}.
\]

The number of elements in a set \( Z_k \) is

\[
n[k] = azg[k] - azdI[k] + 1.
\]

Let us denote by the letter \( D \) a sequence, the elements of which

\[
d[g], d[g+1], \ldots, d[gI],
\]
such that

\[
d[g] < d[g+1] < \ldots < d[gI],
\]
are the subscripts of those elements of the sequence \( C \) which divide it into the following segments:

\[
S_g = \{c|d[g]|, c|d[g]+1|, \ldots, c|d[g+I]−I|\},
\]

\[
S_{g+1} = \{c|d[g+1]|, c|d[g+1]+1|, \ldots, c|d[g+2]−I|\},
\]

\[
S_{gI−1} = \{c|d[gI−1]|, c|d[gI−1]+1|, \ldots, c|d[gI]−I|\}.
\]

For every set \( Z_k \) added to the sequence \( C \) there are such numbers \( p \) and \( pI \) that

\[
g \leq p < pI \leq gI, \text{ and } azdI[k] = d[p], \text{ and } azg[k] = d[pI]−1.
\]

Hence, every set \( Z_k \) consists only of a certain number of whole segments and any transposition of elements within a segment is regarded here as not affecting the set \( Z_k \).

At the outset the program arranges the set \( Z_k \) in such an order that there is

\[
n[1] \leq n[2] \leq \ldots \leq n[iz].
\]

Example. A sequence of sets

\[
\begin{array}{c}
1 \ 2 \ 6 \ 7 \\
3 \ 4 \ 5 \ 6 \ 7 \\
3 \ 4 \ 5 \\
2 \ 3 \ 4 \ 6
\end{array}
\]
will be arranged in the following order:

\[
\begin{align*}
3 & \quad 4 & \quad 5 & \quad 6 & \quad 7 \\
1 & \quad 2 & \quad 6 & \quad 7 \\
2 & \quad 3 & \quad 4 & \quad 6 \\
3 & \quad 4 & \quad 5 & \quad 7
\end{align*}
\]

Such an arrangement of sets reduces in most cases the time required to process the data.

Next, the program finds a set Z_k for which there exists another set Z_l (l \neq k) such that Z_l \cap Z_k is the maximum one of all sets of the form Z_i \cap Z_k (i = 1, 2, ..., iz, i \neq k).

In the example given above there is k = 1. The set

\[
3 \quad 4 \quad 5 \quad 6 \quad 7
\]

has three elements in common with the set

\[
2 \quad 3 \quad 4 \quad 6.
\]

The sequence C consists now of one segment, viz., the set Z_k just found.

Let us denote by the symbol md[p, k] the number of elements z[i] which fulfill the following relation:

\[
z[i] \in S_p \cap Z_k.
\]

Let mc[k] be the number of elements common to the set Z_k and the sequence C, i.e.,

\[
mc[k] = \max (mc1[k], mc2[k], mc3[k]).
\]

Here

\[
mc1[k] = \begin{cases} 
n[k] & \text{(if there exist such } p \text{ and } pI \text{ that} } \\
Z_k \subseteq \bigcup_{i=p}^{p+I} S_i \text{ and } \bigcup_{i=p+1}^{p+I} S_i \subseteq Z_k, \\
0 & \text{(if otherwise),}
\end{cases}
\]

\[
mc2[k] = \sum_{i=p+1}^{gI} (d[i]-d[i-1]) + md[p-1, k],
\]

where p is such that

\[
S_{p-1} \not\subseteq Z_k \text{ and } S_j \subseteq Z_k \quad (j = p, p+1, ..., gI-1),
\]

and

\[
mc3[k] = \sum_{i=q}^{p-1} (d[i+1]-d[i]) + md[p, k],
\]

where p is such that
Further, the program seeks such a set $Z_k$ for which there is
\[
mc[k] = \max_{Z_i \in C} (mc[i]).
\]

The program starts to evaluate numbers $mc[k]$ for the sets with greatest numbers of elements. At the same time it determines the maximum value of $mc[k]$. If for a set $Z_i$ investigated as the next one the number of its elements, $n[i]$, is not greater than the altogether found maximum value of $mc[k]$, then, making use of the fact that the numbers of elements in the remaining sets are not greater than $n[l]$, the process of evaluating $mc[k]$ is terminated.

Let us assume that $mc[k] > 0$. The set $Z_k$ has been therefore determined. According to the value of $mc[k]$, there are three different ways of adding the set $Z_k$ to the sequence $C$.

1° $mc[k] = mc1[k]$.

The elements of the segment $S_p$ and those of the segment $S_{pl}$ are being arranged in such an order that the elements which also belong to $Z_k$ have their subscripts greater in the segment $S_p$ and smaller in the segment $S_{pl}$ than the remaining elements of these segments. Both segments are therefore divided into two parts. The subscript $r$ which divides the segment $S_p$ and the subscript $t$ dividing $S_{pl}$ are now added to the sequence $D$ in such a way that the resulting sequence is again an increasing one. The $azd1[k]$ and $azg[k]$ take on the values $r$ and $t−1$, respectively. The number of segments in the sequence $C$ has therefore increased by two. If $md[p, k] = 0$ and/or $md[pl, k] = 0$ then the respective segments $S_p$ and/or $S_{pl}$ are not divided, and the subscripts $r$ and/or $t$ have the values $d[p+1]$ and/or $d[pl−1]$.

Example. If the sequence $C$ has the form
\[
|1\ 3\ 7\ |\ 4\ 5\ 8\ 9\ |\ 2\ 6|,
\]
then after adding the following set $Z_k$
\[
2\ 3\ 4\ 5\ 8\ 9
\]
it becomes
\[
|1\ 7\ |\ 3\ |\ 4\ 5\ 8\ 9\ |\ 2\ |\ 6|.
\]

The number of segments has increased by two.
If the set $Z_k$ has the form
\[
4\ 5\ 8,
\]
then after adding it to $C$ the number of segments increases by one and the resulting sequence $C$ becomes
\[
|1\ 3\ 7\ |\ 9\ |\ 4\ 5\ 8\ |\ 2\ 6|.
\]

2° $mc[k] = mc2[k]$. 
The elements of the segment $S_{p-1}$ are being arranged in such an order that those which also belong to $Z_k$ have their subscripts greater than the remaining ones. The obtained in this way subscript $r$ which divides $S_{p-1}$ into two parts is now added to the sequence $D$ and we have $azdI[k] = r$. Moreover, there is formed another segment, $S_{q1}$, consisting of those elements of $Z_k$ which do not belong to $\bigcup_{i=p-1}^{gl-1} S_i$. The value of $azg[k]$ is equal to the greatest subscript of $S_{q1}$, i.e., $d[gI+1]−1$.

Example. If the sequence $C$ is as follows

$$|1\ 3\ 7\ 4\ 5\ 8\ 9\ |\ 2\ 6|,$$

then after adding to it the set $Z_k$

$$1\ 2\ 5\ 6,$$

the resulting sequence $C$ becomes

$$|1\ 3\ 7\ 4\ 8\ 9\ |\ 5\ 2\ 6\ |\ 1|.$$  

$3^o\ mc[k] = mc3[k].$

The elements of the segment $S_p$ are being arranged in such an order that those which also belong to $Z_k$ have their subscripts smaller than the remaining ones. The obtained in this way subscript $t$ which divides $S_p$ into two parts is now added to the sequence $D$ and we have $azg[k] = t−1$. Moreover, there is formed another segment, $S_{q-1}$, consisting of those elements of $Z_k$ which do not belong to $\bigcup_{i=q}^{p} S_i$. The $azdI[k]$ is equal to the smallest subscript of $S_{q-1}$, i.e., $d[g−1]$.

Example. The sequence $C$ has the form

$$|1\ 3\ 7\ 4\ 5\ 8\ 9\ |\ 2\ 6|.$$  

After the following set $Z_k$

$$1\ 2\ 3\ 4\ 7$$

is added to $C$, the latter becomes

$$|2\ |\ 1\ 3\ 7\ |\ 4\ |\ 5\ 8\ 9\ |\ 2\ 6|.$$  

If $mc[k] = 0$ and not all the sets have been added to the sequence $C$, then these sets are added to each other separately (in the same manner as the previous ones) and the resulting sequence is joined to the sequence $C$.

In the sequel there are given the successive steps of forming the sequence $C$ from the sets mentioned at the beginning of this section.

Step 1. The sequence $C$ consists of one segment, viz., the set

$$3\ 4\ 5\ 6\ 7.$$
Step 2. There is added the set
\[ 2 \ 3 \ 4 \ 6, \]
for which we have \( mc[k] = mc2[k] = 3. \) The sequence \( C \) now becomes
\[ | 5 \ 7 | \ 3 \ 4 \ 6 | \ 2 |. \]

Step 3. There is added the set
\[ 3 \ 4 \ 5, \]
for which \( mc[k] = mcI[k] = 3. \) The sequence \( C \) is now as follows:
\[ | 7 \ 5 | \ 3 \ 4 \ 6 | \ 2 |. \]

Step 4. There is added the set
\[ 1 \ 2 \ 6 \ 7, \]
for which \( mc[k] = mc2[k] = 2. \) The final form of the sequence \( C \) is therefore the following:
\[ | 7 \ 5 | \ 3 \ 4 \ 6 | \ 2 | \ 1 \ 7 |. \]

Returning to the initial order of the sets \( Z_k \) we obtain
\[
\begin{align*}
\end{align*}
\]

Example. The number of sets \( Z_k \) is 6. The overall number of elements of the sets \( Z_k = 34. \)

The limits of the sets \( Z_k \) are equal to
\[
\begin{align*}
1, & \ 5, \\
6, & \ 12, \\
13, & \ 20, \\
21, & \ 27, \\
28, & \ 33, \\
34, & \ 34.
\end{align*}
\]

The sequence of the sets \( Z_k \) is
\[ 1, \ 2, \ 3, \ 6, \ 7, \ 3, \ 4, \ 5, \ 6, \ 7, \ 8, \ 9, \ 1, \ 2, \ 3, \ 4, \ 5, \ 6, \ 7, \ 8, \ 9, \ 1, \ 2, \ 5, \ 6, \ 7, \ 9, \ 2. \]

The number of elements in the sequence \( C = 11. \) The limits of the sets \( Z_k \) in the sequence \( C \) are equal to
\[
\begin{align*}
3, & \ 7, \\
5, & \ 11, \\
3, & \ 10, \\
2, & \ 8, \\
1, & \ 6, \\
4, & \ 4.
\end{align*}
\]
The sequence $C$ has the following form:

$$5, 9, 1, 2, 6, 7, 3, 8, 4, 5, 9.$$ 

4. Running time. A comparison of the amounts of time required to process different sets of data in the case where the program is written in the ODRA 1204 machine code is given in table 1.

The program in ODRA-ALGOL has required 40 sec. to process the first set of data.

<table>
<thead>
<tr>
<th>Number of sets</th>
<th>Overall number of elements</th>
<th>Number of elements in the sequence $C$</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>173</td>
<td>119</td>
<td>4</td>
</tr>
<tr>
<td>32</td>
<td>102</td>
<td>79</td>
<td>4</td>
</tr>
<tr>
<td>74</td>
<td>1133</td>
<td>798</td>
<td>137</td>
</tr>
<tr>
<td>87</td>
<td>1380</td>
<td>989</td>
<td>190</td>
</tr>
<tr>
<td>249</td>
<td>1311</td>
<td>861</td>
<td>961</td>
</tr>
<tr>
<td>327</td>
<td>2131</td>
<td>1291</td>
<td>2619</td>
</tr>
</tbody>
</table>

Reference


DEPT. OF NUMERICAL METHODS
UNIVERSITY OF WROCŁAW

Received on 3. 11. 1969

KRYSZTYNA JERZYKIEWICZ (Wrocław)

ALGORYTM 8

PROGRAM SKRACANIA CIĄGU ZBIORÓW

STRESZCZENIE

Dany jest ciąg liczb całkowitych

$$z[1], z[2], \ldots, z[x].$$

Ciąg ten zawiera $i$ zbiorów $Z_k$ ($k = 1, 2, \ldots, i$). Do zbioru $Z_k$ należą elementy

$$z[azd[k]], z[azd[k]+1], \ldots, z[asg[k]].$$
Algorithm 8

Program tworzy możliwie najkrótszy ciąg C składający się z elementów
c[1], c[2], ..., c[ile]

w ten sposób, aby każdy zbiór Z_k był odcinkiem tego ciągu (przy czym kolejność elementów zbioru Z_k w takim ciągu nie jest istotna).

Program czyta następujące wartości:
iz, x, azd[1], azg[1], azd[2], azg[2], ..., azd[iz], azg[iz], z[1], z[2], ..., z[x].

Po utworzeniu ciągu C program drukuje następujące wartości:
iz, ile, azd I[1], azg I[1], azd I[2], azg I[2], ..., azd I[iz], azg I[iz], c[1], c[2], ..., c[ile],

gdzie azd I[k], azg I[k] (k = 1, 2, ..., iz) są odpowiednio wskaźnikami pierwszego i ostatniego elementu tego odcinka ciągu C, który stanowi zbiór Z_k.

-------------------

KRYSZYNA EJIKIEWICZ (Brojlaw)

ПРОГРАММА СЖАТИЯ ПОСЛЕДОВАТЕЛЬНОСТИ МНОЖЕСТВ

РЕЗЮМЕ

Пусть дана последовательность целых чисел

z[1], z[2], ..., z[x].

Она содержит iz множеств Z_k (k = 1, 2, ..., iz). Множество Z_k состоит из элементов

z[azd[k]], z[azd[k]+1], ..., z[azg[k]].

Программа создает возможно кратную последовательность C, составленную из элементов
c[1], c[2], ..., c[ile]

tak, чтобы любое множество Z_k было отрезком этой последовательности (причем порядок элементов множества Z_k в этой последовательности не существуетен). Программа читает следующие данные

iz, x, azd[1], azg[1], azd[2], azg[2], ..., azd[iz], azg[iz], z[1], z[2], ..., z[x]

и печатает следующие значения

iz, ile, azd I[1], azg I[1], azd I[2], azg I[2], ..., azd I[iz], azg I[iz], c[1], c[2], ..., c[ile],

где azd I[k], azg I[k] (k = 1, 2, ..., iz) являются соответственно индексами первого и последнего элемента того отрезка последовательности C, который дает множество Z_k.