ALGORITHM 27
ASSIGNMENT

ROLAND SILVER
MIT Lincoln Laboratory,* Lexington, Massachusetts

```
procedure Assignment(d, n, x) ; value n ; integer n ;
            array d ; integer array x ;
comment: Assignment determines that permutation x of the
            integers [1:n] for which the sum (i := 1(1)n) of
            the elements d[i, x[i]] of the n × n matrix d is a
            minimum. n ≧ 2. For more complete information
            see: An Algorithm for the Assignment Problem,
            Roland Silver, Comm. ACM, Nov. 1960, p. 605 ;
   begin
   switch Switch := NEXT, L1, NEXT1, MARK ;
   array a[1:n, 1:n] ;
   integer array c[1:n], cb[1:n], lambda[1:n], mu[1:n],
            r[1:n], y[1:n] ;
   integer cbl, cl, cl0, i, j, k, l, rl, rs, sw ;
   comment:
INITIALIZE ;
   for i := 1 step 1 until n do
      begin min := d[i, 1] ;
      for j := 2 step 1 until n do if d[i, j] < min then min
         := d[i, j] ;
      for j := 1 step 1 until n do a[i, j] := d[i, j] - min
      end i ;
   for j := 1 step 1 until n do
      begin min := a[1, j] ;
      for i := 2 step 1 until n do if a[i, j] < min then min
         := a[i, j] ;
      for i := 1 step 1 until n do a[i, j] := a[i, j] - min
      end j ;
   for i := 1 step 1 until n do x[i] := y[i] := 0 ;
   for i := 1 step 1 until n do
      begin
      for j := 1 step 1 until n do
         begin
         if a[i, j] ≠ 0 ∨ x[i] ≠ 0 ∨ y[j] ≠ 0 then go to J1 ;
         x[i] := j ; y[j] := i
J1:      end j ;
      end i ;
START:  comment: Start labeling ;
        rl := cl := 0 ; rs := 1 ;
        for i := 1 step 1 until n do
           begin mu[i] := lambda[i] := 0 ;
           if x[i] ≠ 0 then go to I1 ;
           rl := rl + 1 ; r[rl] := i ; mu[i] := -1
I1:        end i ;
LABEL:  comment: Label and scan ;
        i := r[rs] ; rs := rs + 1 ;
        for j := 1 step 1 until n do
           begin if a[i, j] ≠ 0 or lambda[j] ≠ 0 then go
             to J2 ;
           lambda[j] := i ; cl := cl + 1 ; c[cl] := j ;
           if y[j] = 0 then go to MARK ;
           rl := rl + 1 ; r[rl] := y[j] ; mu[y[j]] := i
J2:        end j ;
           if rs ≦ rl then go to LABEL ;
           comment:
RENORMALIZE ;
   sw := 1 ; cl0 := cl ; cbl := 0 ;
   for j := 1 step 1 until n do
      begin if lambda[j] ≠ 0 then go to J3 ;
      cbl := cbl + 1 ; cb[cbl] := j
J3:   end j ;
  min := a[r[i], cb[i]] ;
  for k := 1 step 1 until rl do
     begin
     for l := 1 step 1 until cbl do if a[r[k], cb[l]] ≦ min
        then min := a[r[k], cb[l]]
     end k ;
  for i := 1 step 1 until n do
     begin if mu[i] ≠ 0 then go to I2 ;
     for l := 1 step 1 until cl0 do a[i, c[l]] := a[i, c[l]] + min ;
     go to I3 ;
I2:  for l := 1 step 1 until cbl do
        begin a[i, cb[l]] := a[i, cb[l]] - min ;
       go to Switch[sw] ;
NEXT:  if a[i, cb[l] ≠ 0 ∨ lambda[cb[l]] ≠ 0 then go to L1 ;
       lambda[cb[l]] := i ;
       if y[cb[l]] = 0 then
          begin j := cb[l] ; sw := 2 ; go to L1 end ;
       cl := cl + 1 ; c[cl] := cb[l] ;
       rl := rl + 1 ; r[rl] := y[cb[l]] ;
L1:    end l ;
I3:        end i ;
        go to Switch[sw + 2] ;
NEXT1:  if cl0 = cl then go to LABEL ;
        for i := cl0 + 1 step 1 until cl do mu[y[c[i]]] := c[i] ;
        go to LABEL ;
MARK:  comment: mark new column and permute ;
       y[j] := i := lambda[j] ;
       if x[i] = 0 then begin x[i] := j ; go to
         START end ;
       k := j ; j := x[i] ; x[i] := k ;
       go to MARK
       end Assignment
```

* Operated with support from the U. S. Army, Navy and Air Force.

[NOTE: The reader should distinguish between the letter and the figure 1, both of which appear in the above algorithm.—Ed.]

CERTIFICATION OF ALGORITHM 27
ASSIGNMENT [Roland Silver, *Comm. ACM*, Nov. 1960]

ALBERT NEWHOUSE
University of Houston, Houston, Texas

The ASSIGNMENT algorithm was translated into MAD and successfully run on the IBM 709/7094 after the following corrections were made:

All references to array a and d refer to the same array, i.e. change all $a[i, j]$ to $d[i, j]$. Furthermore:

(a) 3rd line after *LABEL*: **comment**: Label and scan;
 should read
 begin if $d[i, j] \neq 0 \vee$ lambda $[j] \neq 0$ **then go**

(b) first line after *J3*: **end** j;
 should read
 $min := d[r[1], cb[1]]$;

(c) line *I2*:
 should read
 I2: **for** $l := 1$ **step** 1 **until** cbl **do**

Since there is no provision made for this algorithm to end the following additions were made:

(1) in the integer declaration add the variable: *flag*

(2) first line after *START*: **comment**: ···
 add the line
 $flag := n$;

(3) first line before *I1*: **end** i;
 change to read
 $rl := rl + 1$; $r[rl] := i$; $mu[i] := -1$; $flag := flag - 1$

(4) add a line after *I1*: **end** i;
 if $flag = n$ **then go to** *FINI*;

(5) change the last line of the algorithm to read:
 FINI: **end** Assignment

In order to obtain the minimum value of the $\sum_{i=1}^{n} a_{ix_i}$ (in the following called total) the following additions may be made:

Add a real variable *total* and

(A) new line after *INITIALIZE*;
 $total := 0$;

(B) new line after the first **end** i;
 $total := total + min$;

(C) new line after the first **end** j;
 $total := total + min$;

(D) after the line **end** k; after *J3*: **end** i;
 add the line
 $total := total + (rl+cbl-n) \times min$;

CERTIFICATION OF ALGORITHM 27

ASSIGNMENT [Roland Silvers, *Comm. ACM 3*, Nov. 1960].

Robert D. Witty
Burroughs Corp., Detroit, Mich.

Assignment was successfully run on the Burroughs B5000 using Burroughs extended Algol 60.

Input Array

60	0	0	76	0	0
0	40	18	0	60	24
60	16	2	4	0	40
0	27	18	3	55	75
0	40	62	16	11	53
28	4	10	84	0	16

Solution Vector: $X(6, 4, 3, 1, 5, 2)$

The following changes were made in the algorithm prior to its successful run:

```
FROM  MIN := a[r[i], cb[i]];
TO    MIN := a[r[1], cb[1]];
FROM  if X[i] = 0 then begin X[i] := j;
      go to START end;
TO    if X[i] = 0 then begin X[i] := j;
      for i := 1 step 1 until N do begin if X[i] = 0 then go
          to START;
        end;  go to EXIT;  end;
FROM  end ASSIGNMENT
TO    EXIT:  end;  ASSIGNMENT
```