Algorithms

J. H. WEGSTEIN, EDITOR

Contributions to this department must be in the form stated in the Algorithms Department policy statement (*Communications*, February, 1960) except that ALGOL 60 notation should be used (see *Communications*, May, 1960). Contributions should be sent in duplicate to J. H. Wegstein, Computation Laboratory, National Bureau of Standards, Washington 25, D. C. Algorithms should be in the Publication form of ALGOL 60 and written in a style patterned after the most recent algorithms appearing in this department.

Although each algorithm has been tested by its contributor, no warranty, express or implied, is made by the contributor, the editor, or the Association for Computing Machinery as to the accuracy and functioning of the algorithm and related algorithm material and no responsibility is assumed by the contributor, the editor, or the Association for Computing Machinery in connection therewith.

The reproduction of algorithms appearing in this department is explicitly permitted without any charge. When reproduction is for publication purposes, reference must be made to the algorithm author and to the *Communications* issue bearing the algorithm.

ALGORITHM 69
CHAIN TRACING

BRIAN H. MAYOH
Regnecentralen, Gl. Carlsbergvet. 2, Copenhagen.

```
procedure CHAIN tracing (iteration counter, number of
          identifiers, number of identifier links, final linkage
          matrix, couples);
    Boolean array final linkage matrix;
    integer array couples;
    integer iteration counter, number of identifiers, number of
      identifier links;
begin comment This procedure is given a list of pairs of inte-
    gers, the second being related to the first in some way. It finds
    those pairs of integers which are related to each other if the
    relation is transitive. It is supplied with,
    couples a matrix whose bound pairlist is [1:2, 1:number of
      identifier links] where couples [2, i] is related to couples
      [1, i] in some way.
    final linkage matrix a matrix whose bound pair list is
      [1:number of identifiers, 1:number of identifiers] and into
      which the procedure puts true if the second subscript
      expression is an integer which is related to the integer
      corresponding to the first subscript expression, if the
      relation is irreflexive then the diagonal entries of this
      matrix are false.
    iteration counter a place for the procedure to put the
      length of the longest chain it finds. CHAIN tracing can be
      applied to any system which can be represented by a Turing
      machine by letting the integers in couples correspond to
      the Turing machine states. Two integers j, k are related if
      there is an input symbol which causes state j to change to
      state k. If the Turing machine always stops whatever the
      sequence of input symbols, then its final linkage matrix
      will have false for all leading diagonal entries;
  integer i, j;
  Boolean array working linkage matrix [1:number of identi-
    fiers, 1:number of identifiers];
  Boolean procedure PROGRESS;
    begin PROGRESS := false;
      for i := 1 step 1 until number of identifiers
        do for j := 1 step 1 until number of identifiers
          do begin if Working linkage matrix [i, j] ≡ ¬ Final
              linkage matrix [i, j] then PROGRESS := true;
              Final linkage matrix [i, j] := Working linkage
              matrix [i, j]
            end of comparison
    end of PROGRESS;
BEGIN OF PROGRAM:
  for iteration counter := −1, 0, iteration counter + 1 while
    PROGRESS
    do for i := 1 step 1 until number of identifier links
      do for j := 1 step 1 until number of identifiers
        do begin if iteration number = −1
            thenFinal linkage Matrix [couples [1, i], j]
              := Working linkage Matrix [couples [1, i], j]
              := couples [2, i] = j
            else Working linkage Matrix [couples [1, i], j]
              := Working linkage Matrix [couples [1, i], j]
              ∨ Working linkage Matrix [couples [2, i], j];
          end of setting one linkage
end of CHAIN tracing;
```

CERTIFICATION OF ALGORITHM 40
CRITICAL PATH SCHEDULING (B. Leavenworth, *Comm. ACM*, Mar. 1961)

NEAL P. ALEXANDER
Union Carbide Olefins Company, South Charleston, West Virginia

The Critical Path Scheduling algorithm was coded in FORTRAN for the IBM 7070. The following changes were made:

(a) ti [k] := te [k] := 0;

should be

ti [k] := 0;
te [k] := 9999;

(b) **if** te [I[k]] = 0 ∨ te [I[k]] > min **then**

should be

if te [I[k]] > min **then**

This change permits a value of 0 to be calculated for te [I[k]] and remain as the minimum value.

In the statement

if ti [J[k]] = 0 ∨ ti [J[k]] < max **then**

the part of the statement "ti [J[k]] = 0" is redundant and can be omitted.