

## Algorithm AS 13

# Minimum Spanning Tree

By G. J. S. ROSS

*Rothamsted Experimental Station*

### LANGUAGE

Algol 60.

### DESCRIPTION AND PURPOSE

This procedure computes the minimum spanning tree of a distance matrix by the method of Prim, as described by Gower and Ross (1969) elsewhere in this journal. The procedure can be easily modified to handle a similarity matrix by reversing the relevant inequalities. The distance matrix is assumed stored in lower triangular form with diagonal terms omitted. The procedure can be modified to handle matrices stored in other ways.

### STRUCTURE

The procedure call is *Primtree* ( $n$ ,  $dlarge$ ,  $D$ ,  $B$ ,  $C$ , *ifault*).

#### *Formal parameters*

##### Input:

$n$	integer	The number of points = the order of the matrix.
$dlarge$	real	An arbitrary value larger than the maximum element of $D$ . A large power of 2 is usually suitable.
$D$	real array	The lower triangular distance matrix, with bounds [1 : $n \times (n-1)/2$ ].

##### Output:

$B$	integer array	On leaving the procedure $B[i]$ contains the index of a point to which $i$ is jointed. Bounds [2 : $n$ ].
$C$	real array	$C[i]$ is the distance between $i$ and $B[i]$ . Bounds [2 : $n$ ].
<i>ifault</i>	integer	Set to 1 if $n < 2$ , 0 otherwise.

### RESTRICTIONS

The minimum permissible value of  $n$  is 2. The maximum value will be determined by storage considerations.

### TIME

The time required depends on  $n^2$ .

### ACCURACY

The resulting tree may not be unique if the matrix contains equal elements.

## REFERENCES

- GOWER, J. C. and ROSS, G. J. S. (1969). Minimum spanning trees and single linkage cluster analysis. *Appl. Statist.*, 18, 54-64.
- PRIM, R. C. (1957). *Bell System Tech. J.*, 36, 1389-1401.

```

procedure Printtree(n, dlarge, D, B, C, ifault);

comment Algorithm AS 13, J.R.statist.Soc. C, (1969), Vol.18, No.1;

value n, dlarge; integer n, ifault; real dlarge;
array D, C; integer array B;
  begin

  comment D is a lower triangular distance matrix with diagonal elements
    omitted. dlarge is a constant larger than any element of D. The object
    of the procedure is to define for each number  $2 \leq i \leq n$  a partner
     $1 \leq R[i] \leq n$  with corresponding distance C[i] such that the tree is
    connected and of minimum length. The results of the procedure are thus
    packed as a single array. In some languages A, B and C may be conveniently
    packed as a single array. Unknown distances can be handled if they are
    coded as dlarge. The procedure can equally well be adapted for use with
    similarity matrices (minimum distance = maximum similarity, dlarge = 0,
    etc.);

  integer i, j, k, next; real min, dist;
  integer array A[2:n];

  comment A[i] := 1 if i is already assigned to the tree (initially
    consisting of no. 1 only), or 0 otherwise;

  if n > 1 then
    begin
    ifault := 0;
    for i := 2 step 1 until n do
      begin
      A[i] := B[i] := 0; C[i] := dlarge
      end ;
      j := 1;
      for i := 2 step 1 until n do
        begin
        min := dlarge;
        for k := 2 step 1 until n do
          if A[k] = 0 then
            begin
            dist := D[ if j > k then (j - 1) X (j - 2) ÷ 2 + k
                      else (k - 1) X (k - 2) ÷ 2 + j];
            if dist < C[k] then
              begin
              C[k] := dist; B[k] := j
              end ;
            if min > C[k] then
              begin
              min := C[k]; next := k
              end
            end ;
            j := next; A[j] := 1
          end
        end
      end
    else ifault := 1
  end Printtree

```