1 SUMMARY

This subroutine **generates the expanded structure for a sparse symmetric matrix** given the structure for the lower triangular part. Diagonal entries need not be present.

ATTRIBUTES — Version: 1.0.0. (12 July 2004) **Types:** Real (single, double). **Original date:** February 1987. **Origin:** I.S. Duff, Harwell.

2 HOW TO USE THE PACKAGE

2.1 Argument list

The single precision version

```
CALL MC34A(N,IRN,JCOLST,YESA,A,IW)
```

The double precision version

```
CALL MC34AD(N,IRN,JCOLST,YESA,A,IW)
```

`N` is an `INTEGER` variable that must be set by the user to the order of the matrix. This argument is not altered by the subroutine.

`IRN` is an `INTEGER` array that must be set by the user to hold the row indices of the lower triangular part of the symmetric matrix. The entries of a single column are contiguous. The entries of column `J` precede those of column `J+1` (`J=1`, ..., `N-1`), and there is no wasted space between columns. Row indices within a column may be in any order. On exit it will have the same meaning but will be changed to hold the row indices of entries in the expanded structure. Diagonal entries need not be present. The new row indices added in the upper triangular part will be in order for each column and will precede the row indices for the lower triangular part which will remain in the input order.

`JCOLST` is an `INTEGER` array of length `N+1` that must be set by the user so that `JCOLST(J)` is the position in arrays `IRN` and `A` of the first entry in column `J` (`J=1`, ..., `N`). `JCOLST(N+1)` must be set to one more than the total number of entries. On exit, `JCOLST(J)` will have the same meaning but will be changed to point to the position of the first entry of column `J` in the expanded structure. The new value of `JCOLST(N+1)` will be one greater than the number of entries in the expanded structure.

`YESA` is a `LOGICAL` variable that must be set to `.TRUE.` if the user desires to generate the expanded form for the values also. If `YESA` is `.FALSE.`, the array A will not be referenced. This argument is not altered by the subroutine.

`A` is a `REAL` (`DOUBLE PRECISION` in the D version) array that can be set by the user so that `A(K)` holds the value of the entry in position `K` of `IRN`, `K=1`, ... `JCOLST(N+1)-1`. On exit, if `YESA` is `.TRUE.`, the array will hold the values of the entries in the expanded structure corresponding to the output values of `IRN`. If `YESA` is `.FALSE.`, the array is not accessed by the subroutine.

`IW` is an `INTEGER` array of length `N` that will be used as workspace.

3 GENERAL INFORMATION

Workspace: Array `IW` is used as workspace (see §2.1).

Use of common: None.

Other routines called directly: None.

Input/output: None.

4 METHOD

The total number of entries in each column in the expanded form is first calculated by scanning the input matrix once. The entries of the input matrix are then placed at the ends of the columns in the expanded form. Since the same arrays are reused, this copying starts with the last column. A pointer is maintained pointing to the first filled position in each column. A final scan through the moved parts of the columns is used to place each entry a_{ij}, $i \geq j$ immediately in the correct position of its symmetric counterpart a_{ji}.

5 EXAMPLE OF USE

A simple example of the subroutine running on a 4×4 matrix is given by the following code.

```
      INTEGER IRN(7),IW(4),JCOLST(5),N,I,NZ
      DOUBLE PRECISION A(1)
      LOGICAL YESA

      YESA = .FALSE.
      READ (5,FMT=100) N
      READ (5,FMT=100) (JCOLST(I),I=1,N+1)
      NZ = JCOLST(N+1) - 1
      READ (5,FMT=100) (IRN(I),I=1,NZ)

  100 FORMAT (15I4)

      CALL MC34AD(N,IRN,JCOLST,YESA,A,IW)
      WRITE (6,FMT=101) N
      WRITE (6,FMT=102) (JCOLST(I),I=1,N+1)
      NZ = JCOLST(N+1) - 1
      WRITE (6,FMT=103) (IRN(I),I=1,NZ)

  101 FORMAT (' N = ',I3)
  102 FORMAT (' JCOLST .. ',/, (1X,15I4))
  103 FORMAT (' IRN ..... ',/, (1X,15I4))

      STOP

      END
```

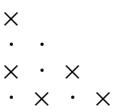
is run on the data...

```
 4
 1   3   4   5   6
 1   3   4   3   4
```

it produces as output

```
 N =   4
 JCOLST ..
    1   3   4   6   8
 IRN .....
    1   3   4   1   3   2   4
```

That is to say that the symmetric matrix originally held as :



would be held as: