As an example, let us consider the following matrix:

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}
\begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\left[\begin{array}{cccccc}
1 & & & & & 1 \\
1 & 1 & & & & \\
 & 1 & 1 & & & \\
 & & 1 & 1 & & \\
 & & & 1 & 1 & \\
 & & & & 1 & 1
\end{array}\right] \end{array} \tag{4.7}
$$

The first, third, and fifth rows are independent. Hence, we need at least three columns to cover the matrix. There is another independent set of three rows, namely the second, fourth, and sixth rows. Notice that in this case the lower bound is exact: we can cover the matrix with exactly three columns. In general, however, there will be independent sets of different sizes and the lower bound will not necessarily be exact. It is also easy to see that there is always at least an independent set of size 1. Further, for cyclic (irreducible) matrices with unit cost columns, we always get a lower bound of size 2 or more, as shown in the following theorem.

Theorem 4.8.1 *In the unate covering problem with unit costs for the columns, the lower bound for a cyclic matrix is at least 2 (even if there are no two independent rows).*

Proof. Suppose a constraint matrix of a unate covering problem contains a full column of ones. In this case, the matrix cannot be cyclic because the column with all ones dominates all the others. The matrix can therefore be reduced to one column, which is obviously essential. Thus such a matrix is not cyclic. It then follows that if the matrix is cyclic, there can be no column that covers all the rows. Hence, at least two columns are required to cover all the rows. □

An independent set of rows is called **maximal** if it intersects (that is, has a column in common with) every other row of the covering matrix. We shall use the abbreviation MIS for a Maximal Independent Set. This means that unless some of the decisions already made in building the set are reversed, the set cannot grow larger while retaining its independence (pairwise disjointness).

A simple algorithm for quickly finding an MIS is Procedure MIS_QUICK of Figure 4.7. Here $\| M \|$ denotes the number of rows left in M after deleting the rows intersecting the chosen row.

The key feature of this algorithm is Subprocedure CHOOSE_SHORTEST_ROW. In its simplest form it just chooses the "shortest" row, that is, the row with the fewest nonzero columns, and breaking ties in ascending lexicographical order. Better heuristic performance is usually obtained by a more sophisticated heuristic, in which the weight of row i is defined in terms of the column counts of its columns. That is, let