- In general, we write $m \times n$ (read "m by n") to designate a matrix with m rows and n columns. The total no. of elements in such a matrix is mn. If m equals n, the matrix is square.
- When a matrix is represented as a 2D array defined as a[MAX_ROWS][MAX_COLS], we can locate quickly any element by writing a[i][j], where i is the row index and j is the column index.
- Problem: Huge wastage of space if matrix is sparse, i.e. most of the elements of the matrix are 0. (Difficult to determine exactly whether a matrix is sparse or not).

$$
\begin{array}{c c}
 & \begin{array}{cccccc} \text{col 0} & \text{col 1} & \text{col 2} & \text{col 3} & \text{col 4} & \text{col 5} \end{array} \\
\begin{array}{c} \text{row 0} \\ \text{row 1} \\ \text{row 2} \\ \text{row 3} \\ \text{row 4} \\ \text{row 5} \end{array} &
\begin{bmatrix}
15 & 0 & 0 & 22 & 0 & -15 \\
0 & 11 & 3 & 0 & 0 & 0 \\
0 & 0 & 0 & -6 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
91 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 28 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

Only 8 out of 36 elements are

- We can do much better by using a representation in which only the nonzero elements are stored.

ADT SparseMatrix is

objects: a set of triples, ⟨row, column, value⟩, where row and column are integers and form a unique combination, and value comes from the set item.

functions: for all a, b ∈ SparseMatrix, x ∈ item, i, j, maxCol, maxRow ∈ index

SparseMatrix Create(maxRow, maxCol) ::= return a SparseMatrix that can hold up to maxItems = maxRow × maxCol and whose maximum row size is maxRow and whose maximum column size is maxCol.

SparseMatrix Transpose(a) ::= return the matrix produced by interchanging the row and column value of every triple.