DEF: Elementary Row Operations (for matrices).

Type I Swap Row i and Row j

Type II Multiply Row i by c where $c \neq 0$.

Type III Add c times Row i to Row j where c is any scalar.

DEF: A matrix is in **row echelon form** if

- Each non-zero row is above all zero rows – that is – zero rows are "pushed" to the bottom.
- The leading entry of a row is *strictly* to the right of the leading entries of the rows above. (The leftmost non-zero entry of a row is called the "leading entry".)
- Each leading entry is "1". (*Note:* Many textbooks do not make this third requirement.)

If in addition...

- Only zeros appear above (& below) a leading entry of a row.

then a matrix is in **reduced row echelon form**.

Gauss-Jordan Elimination is an "algorithm" which given a matrix returns a row equivalent matrix in reduced row echelon form.

1. Determine the leftmost non-zero column. This is a **pivot column** and the topmost entry is a **pivot position**. If "0" is in this pivot position, swap (an unignored) row with the topmost row (use a Type I operation) so that there is a non-zero entry in the pivot position.
2. Add appropriate multiples of the topmost (unignored) row to the rows beneath it so that only "0" appears below the pivot (use several Type III operations).
3. Ignore the topmost (unignored) row. If any non-zero rows remain, go to step 1.

This part of Gauss-Jordan Elimination is called the **forward pass**. If in addition we scale each row so that the pivots are all 1's (using Type II operations), then our matrix would be in row echelon form. Now let's finish Gauss-Jordan Elimination.

1. If necessary, scale the rightmost unfinished pivot to 1 (use a Type II operation).
2. Add appropriate multiples of the current pivot's row to rows above it so that only 0 appears above the current pivot (using several Type III operations).
3. The current pivot is now "finished". If any unfinished pivots remain, go to step 4.

This part of Gauss-Jordan Elimination is called the **backward pass**. It should be fairly obvious that this algorithm will terminate in finitely many steps. Also, only elementary row operations have been used. So we end up with a row equivalent matrix. A tedious and wordy proof shows that the resulting matrix is in reduced row echelon form.